Order No : ....

UNIVERSITY MOHAMMED SEDDIK BENYAHIA
JIJEL
FACULTY OF EXACT SCIENCES AND COMPUTER SCIENCES

MASTER'S REPORT

Presented for obtaining the :

MASTER'S DEGREE

In **COMPUTER SCIENCE**

Option: ARTIFICIAL INTELLIGENCE

By :

BOUDERBALA Mohammed Amine
LALLOUCHE Roumaissa

Subject

Neurovectors for Drone Flight Planning

Publicly presented, in front of jury composed of :

M.	BOUAZIZ Hamida	President
M.	KOUICEM Amel	Examiner
M.	ZENNIR Mohamed Nadjib	Supervisor
M.	ALIOUA Ahmed	Co-supervisor

Academic year 2021 - 2022

I dedicate this modest work

To my dear Parents

To my dear siblings

To all my dears and friends especially **Khalil**, **Hamza**, **Haitem**

To all the members of Esperanza Club

To my partner in this project **Roumaissa**

To all people who have been there for me

Amine

I dedicate this modest work

To my dear Parents, and my dear siblings

To my partner in this project **Amine**

To all my dears, my friends and my classmates

A special dedication to my grandfather, may he rest in peace, for inspiring me, teaching me to never give up and everything is about willingness.

To my father, may he rest in peace, I hope I made you proud.

AND

To the most important person in my life, who have always been there for me, who loved me unconditionally, who inspires me everyday with her strength, her patience and her faith.
Mother, thank you for your advice, for your love and for always understanding.
thank you for everything.

Roumaissa

Acknowledgment

We would like to sincerely thank our supervisor Mr Mohamed Nadjib ZENNIR and co-supervisor Ahmed ALIOUA for their support and guidance.

And also we would like to thank the jury for agreeing to judge this work.

Abstract

The usage of a completely bio-inspired neuronal model in the navigation of drones is a challenging process. In this thesis, we propose several neuron models for different drone flight scenarios that allow a drone to navigate its way and execute certain tasks using only biologically plausible approaches. We proposed three scenarios, the first one focused on motion camouflage which is most frequently used when an attacker mimics the optic flow of the background. The second scenario handled the tracking of both ground and flying targets, and in the last one, we proposed a cooperative system for drones to provide cellular wireless covering to ground users.

Resumé

L'utilisation d'un modèle neuronal entièrement bio-inspiré dans la navigation des drones est un processus difficile. Dans cette thèse, nous proposons plusieurs modèles de neurones pour différents scénarios de vol de drone qui permettent à un drone de naviguer sur son chemin et d'exécuter certaines tâches en utilisant uniquement des approches biologiquement plausibles. Nous avons proposé trois scénarios, le premier axé sur le camouflage de mouvement qui est le plus fréquemment utilisé lorsqu'un attaquant imite le flux optique de l'arrière-plan. Le deuxième scénario gérait le suivi des cibles au sol et volantes, et dans le dernier, nous avons proposé un système coopératif pour que les drones fournissent une couverture sans fil cellulaire aux utilisateurs au sol.

ملخص

يعد استخدام نموذج عصبي مستوحى بالكامل من العناصر الحيوية في التنقل بين الطائرات بدون طيار عمليةيعد استخدام نموذج عصبي مستوحى بالكامل من الحيوية في التنقل بين الطائرات بدون طيار عملية صعبة. في هذه الأطروحة ، نقترح عدة نماذج عصبية لسيناريوهات طيران مختلفة للطائرات بدون طيار والتي تسمح للطائرة بدون طيار بالتنقل في طريقها وتنفيذ مهام معينة باستخدام الأساليب المعقولة بيولوجيًا فقط. اقترحنا ثلاثة سيناريوهات ، أولها يركز على تمويه الحركة والذي يتم استخدامه بشكل متكرر عندما يحاكي المهاجم التدفق البصري للخلفية. تناول السيناريو الثاني تتبع كل من الأهداف الأرضية والطائرة ، وفي السيناريو الأخير ، اقترحنا نظامًا تعاونيًا للطائرات بدون طيار لتوفير غطاء لاسلكي خلوي لمستخدمي الأرض.

key-words: *Spiking neural network, Neurovector, Unmanned aerial vehicle, Aerial base station, Cellular wireless coverage.*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ABS	Areal base station
BS	Base station
LTD	Long-Term Depression
LTP	Long-Term Potentiation
LoS	Line of sight
MINLP	Mixed Integer Nonlinear Programming
NLoS	Line of sight
QoS	Quality of Service
STDP	Spike-timing dependent plasticity
UAV	Unmanned Aerial Vehicle

INTRODUCTION

Context and Motivation

Drones, commonly referred to as Unmanned aerial vehicles (UAVs), were initially employed by the military forces as weapons for battlefield surveillance and target designation. However, after their level of safety was enhanced they fell into the public domain. These robotic UAVs fly independently and with varying degrees of autonomy. The level of autonomy for a drone can vary from remotely piloted, where a person controls its motions, to advanced autonomy, where it uses a network of sensors and detectors to determine its movement.

They can be an ideal contender for the most difficult tasks that require mobility such as area covering, target tracking, and helping to enhance scientific research in some of the most hostile climates. These tasks can be split into two parts, solo tasks that only require a single drone, and cooperative tasks that require multiple drones to resolve a task. These multi-agent resolving approaches can be difficult as they sometimes require communication and cooperation between the drones.

Highly accurate drone navigation is very important in the air. The latest drones have dual global navigation satellite systems such as GPS, and they can also fly in a non-satellite mode where they use different sensors such as a camera In this case, they rely largely on computer vision to process these images and recognize objects. However, to process all the data in real-time, drones can be equipped with a processing unit. These units come in different flavors (architectures) and can do different types of computations, and one of them is bio-inspired neuromorphic computing which mimics the behavior found in living brains.

This thesis handles the following research question: Is it possible for a drone to use a bio-inspired model using neuromorphic computations to navigate autonomously? and, how well does it perform?

Contributions

The main contributions of this thesis are summarized in the following:

- The construction of a graphical representation framework that allows the modeling of neurovectors and their operations, which are used in the neural modeling of different scenarios to provide simplicity and better understanding.

- Describing a camouflage hunting model, including its mathematical description based on three-dimensional vector equations and how these equations can be converted into a neural model.
- Proposing a neural model for target tracking, where the target is either flying of moving on the ground.
- Proposing a neural model that allows a fleet of UAVs to distribute and provide cellular wireless coverage to a set of ground users.

Organization of the Thesis

This thesis is organized into two parts, the first part covers the state of the arts and in the second different scenarios are proposed where the UAVs use neurovectors to achieve different tasks.

The first part consists of two chapters. In the first one, we introduce spiking neural networks (SNNs), how the concept of time is incorporated into the workings of SNNs, and how the data can be encoded and transferred between neurons. The first chapter was an entry point for the second one in which we introduce the concept of neurovectors, and how these structures allow a population of neurons to represent a vector. We illustrate different operations with graphical schemas in order to model and depict various scenarios for a better understanding.

The second part consists of three chapters that describe three different flying scenarios. We propose, test, and evaluate a neuron model in each scenario. In the first chapter, we handle a nature-inspired behavior that allows the UAV to perform a camouflaged pursuit, the scenario in the second chapter consists of tracking a target who can be flying or moving on the ground, and in the last chapter we proposed an electrostatic based solution to deploy a fleet of UAVs to provide wireless coverage for a set of users.

Part I

State Of The Art

CHAPTER 1

SPIKING NEURAL NETWORKS

1.1 Introduction

Artificial intelligence (AI) is a part of computer science where biological inspiration has a critical impact. Artificial intelligence research started after WWII when many specialists have begun working independently on intelligent machines [1]. Since then artificial intelligence got branched into a wide-open list of computer science subjects such as logical AI, search, pattern recognition, ontology, heuristics, and genetic programming. It's been applied in a lot of domains and some of the applications are e-commerce, education, game playing, speech recognition, computer vision, and expert systems. Two very different approaches of rule-based systems and neural networks have generated several applications capable of making complex decisions.

Despite the high complexity of the brain cells and their connected networks, neuroscientists were able to extract a simple description of how the brain operates consisting of units that work in parallel, each performing a calculation on its inputs and sending the result to the next layer of neurons. Simulating this formalism yields a powerful computational tool called an artificial neural network (ANN) often just called "neural network" (NN). Now the artificial neural network is seen as a simplified model of natural neural processing [2]. Over time and with a better understanding of brain structure and function, artificial neural networks have developed and come closer and closer to real biological models [3]. Since the first perceptron, artificial neural networks went through several generations based on their computational units. First-generation, also referred to as perceptrons or threshold gates are based on McCulloch-Pitts neurons as computational units [4]. Second-generation incorporates the concept of "activation function", such as the Sigmoid function $a(y) = \frac{1}{1+e^{-y}}$, and then the third-generation, networks of spiking neurons are the subject of this chapter.

1.2 Spiking Neural Networks

A computational model inspired by the form and the function of biological neural networks is known as an artificial neural network (ANN). Inside each of our brains lies a biological neural network that is used for information and signal processing, decision making, and a variety of other functions.

Artificial neurons, despite their similarities, do not behave like organic neurons. As a result, biological and artificial NNs vary fundamentally as Figure 1.1 shows.

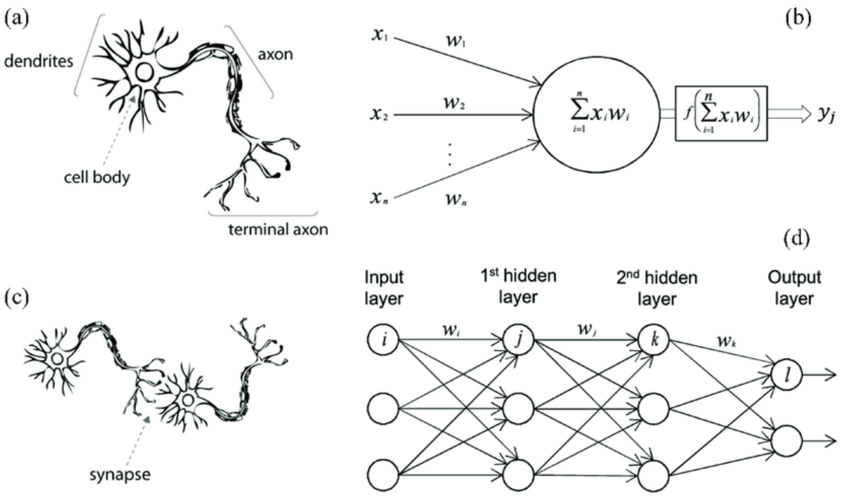

Figure 1.1: A biological neuron in comparison to an artificial neural network: (a) human neuron; (b) artificial neuron; (c) biological synapse; and (d) ANN synapses [5].

The first scientific model of a Spiking Neural Network was created in 1952 by Alan Hodgkin and Andrew Huxley [6]. The model describes how action potentials are initiated and propagated in biological neurons that do not transmit impulses directly. The exchange of chemicals known as neurotransmitters in the synaptic gap is required for such communication [7].

1.3 Neuron Models

Spiking neuron models are mathematical descriptions of the properties of certain cells in the nervous system. Spiking neurons generate sharp electrical potentials, or spikes, called action potentials or spikes. The following section provides an overview of different spiking neuron models.

1.3.1 Integrate and Fire

The integrate-and-fire model has a long and illustrious history. Dating back to Lapicque (1907, 1926) who was able to describe the model before any specific knowledge of excitable membrane biophysics was available. Several people have studied more recent versions of this model (Stein, 1967; 1968; Knight, 1972; Softky and Koch, 1993) [8]. The integrate-and-fire neuron is one of the most basic models of a neuron's electrical properties, and it's also one of the most popular in neuroscience. The model works by dividing the neuron's voltage changes into two parts:

1. Below the threshold, the membrane is considered to behave passively (i.e., without voltage-dependent ion channels) and acts as a leaky capacitor whose voltage decays

(or "leaks") to a resting level EL (short for "E_{Leak}") in the absence of injected current [9].

2. The model assumes that when the voltage hits the action potential threshold (due to injected currents charging up the membrane), the voltage spikes to a level Vspike and then resets to a hyperpolarized level Vreset. The ion channel kinetics responsible for this spiking are not explicitly modeled. Instead, it is assumed (reasonably assumed...) that once the cell hits its threshold, it will immediately produce an action potential and reset itself [9].

The membrane potential is considered to integrate the input current in its most basic form: $C\frac{dV}{dt} = I(t)$. A nerve impulse is fired when V reaches a criterion level V_{th}, V is reset to 0, and the process restarts. The leaky or forgetful integrate-and-fire model assumes that the neuron is a single passive compartment, which is a much more appealing variant. In reaction to the injected current, the membrane potential charges and discharges exponentially $C\frac{dV}{dt} + glV - I(t) = 0$. Where V is the membrane potential and gl is the leak conductance. Once again a nerve impulse is fired when V reaches a criterion level V_{th}, V resets to 0, and the process restarts. “The leaky integrate-and-fire model has been applied to model the firing behavior of many cells, such as neurons in the Limulus eye (Knight, 1972) and cortical cells (Wehmeier et al., 1989; Softky and Koch, 1993; Somers et al., 1995)” [8].

1.3.2 Hodgkin-Huxley Model

The most prominent spiking neuron model is the Hodgkin-Huxley model. It consists of a system of four ordinary differential equations that may be easily combined using a variety of methods. The fundamental concept is based on an electrical description of the neuron that solely considers voltage-gated potassium (K) and sodium (Na) ion channels. Figure 1.2 shows a schematic representation

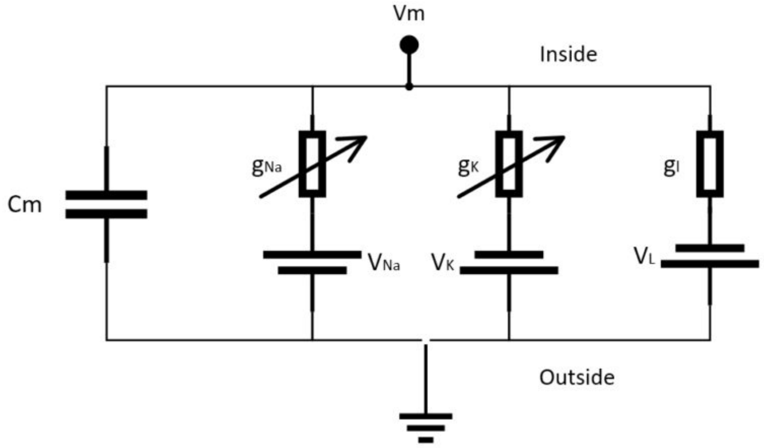

Figure 1.2: Electrical equivalent circuit proposed by Hodgkin and Huxley for a short segment of squid giant axon. The variable resistances represent voltage-dependent conductances (Hodgkin and Huxley 1952d).

The elements are:

- Cm: a capacitance per unit area representing the membrane lipid-bilayer (adopted value: 1 µF/cm^2)
- gNa: voltage-controlled conductance per unit area associated with the Sodium (Na) ion-channel (adopted value: 120 µS/cm^2)
- gK: voltage-controlled conductance per unit area associated with the Potassium (K) ion-channel (adopted value: 36 µS/cm^2)
- gl: conductance per unit area associated with the leak channels (adopted value: 0.3 36 µS/cm^2)
- VNa: voltage source representing the electrochemical gradient for Sodium ions (adopted value: 115 mV)
- VK: voltage source representing the electrochemical gradient for Potassium ions (adopted value: -12 mV)
- Vl: voltage source that determines the leakage current density together with gl (adopted value: 10.613 mV)

The external stimulus current is not indicated in the scheme, but we assume it exists as a current density (I) that encodes the input information. The experimental values are the same as those proposed by the authors in [6], and they all correspond to a zero-voltage equilibrium potential. The following ODE system is used to define the system:

$$\begin{cases} \frac{dV_m}{dt} = \frac{I}{C_m} - \frac{\bar{g}_K n^4}{C_m}(V_m - V_k) - \frac{\bar{g}_{Na} m^3 h}{C_m}(V_m - V_{Na}) - \frac{\bar{g}_l}{C_m}(V_m - V_l) \\ \frac{dn}{dt} = \alpha_n(V_m)(1-n) - \beta_n(V_m)n \\ \frac{dm}{dt} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m \\ \frac{dh}{dt} = \alpha_h(V_m)(1-h) - \beta_h(V_m)h \end{cases} \tag{1.1}$$

The first equation determines the derivative of Vm when the external stimulus (I) is taken into account, as well as the contributions of K, Na, and leakage current densities. The variables n, m, and h are related to the likelihood of each channel opening and are solely reliant on the channel's nature. The K channel, for example, is voltage-gated and comprises four sub-units that must all be open for current to flow, hence its probability is n to the power of 4. Sodium has a slightly more complicated behavior and requires two independent parameters (m and h). The ion-channel kinetic model is described by computing the derivatives of n, and m, and has functions of the same variables and two voltage-dependent functions in the last three equations. The number of closed channels that are opening is the first phase, whereas the number of open channels that are closing is the second term. Hodgkin and Huxley suggest the following functions:

$$\begin{cases} \alpha_n(V_m) = \frac{0.01(10-V_m)}{e^{(1.0-0.1V_m)}-1} \\ \beta_n(V_m) = 0.125e^{-\frac{V_m}{80}} \end{cases} \tag{1.2}$$

$$\begin{cases}\alpha_m(V_m) = \frac{0.1(25-V_m)}{e^{(2.5-0.1V_m)}-1} \\ \beta_m(V_m) = 4e^{-\frac{V_m}{18}}\end{cases} \tag{1.3}$$

$$\begin{cases}\alpha_h(V_m) = 0.07e^{(-\frac{V_m}{20})} \\ \beta_h(V_m) = \frac{1}{e^{(3-0.1V_m)}+1}\end{cases} \tag{1.4}$$

1.3.3 Izhikevich Model

This mathematical model is the most recent in a long line of models that have been used to examine individual neurons with spiking/bursting behavior. These models include Hodgkin-Huxley and FitzHugh-Nagumo, among others. The Izhikevich model is particularly intriguing since it is a small model that can imitate the behavior of a large number of neurons by changing a few parameters. A system of two differential equations makes up the Izhikevich model. The following are the two variables in the system: v, The membrane potential of a neuron is measured in millivolts (mV), and the generic recovery variable is denoted by u. In addition, there are five undefined parameters in the equation [10]

- I is the external input to the neuron, such as those from synaptic inputs.
- a is the recovery rate of u.
- b is the 'sensitivity of recovery to sub threshold fluctuations of membrane potential'.
- c and d are the after spike resets of v and u respectively, i.e. the values to which they get set back after a spike occurs.

The Izhikevich equation is as follows:

$$\frac{dv}{dt} = 0.04 * v^2 + 5 * v + 140 - u + I \tag{1.5}$$

$$\frac{du}{dt} = a * (b * v - u) \tag{1.6}$$

Together with the following reset conditions: if $v \geq 30mV, then \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases}$

1.4 Rate Model

A rate model is a mathematical tractable model for neural networks which is formulated on the level of firing rates, given the gain function $F(h)$ of rate models, it can always be chosen so that the population activity $A_0 = F(h_0)$ in the stationary state of asynchronous firing is accurately characterized for constant input $h_0 = RI_0$, where R and I_0 are two inputs that represents resistance and current respectively. However, the dynamic equations that explain the rate model's approach to the stationary state are ad hoc to some extent [11]. This means that in rate models, the analysis of transients, as well as the examination of stability in recurrent networks will yield different conclusions than in spiking neuron models.

Let's take a look at a network made up of K populations. A homogeneous population of neurons exists in each population [11]. The population k's input from other populations n and recurrent coupling within the population is described as

$$I_k(t) = \sum_n C_{kn} w_{kn} \int_0^\infty \alpha(s) A_n(t-s) ds. \tag{1.7}$$

Here, $A_n(t)$ denotes population n activity, and C_{kn} is the number of presynaptic neurons in population n that are connected to a typical neuron in population k, α and w_{kn} denote the time course and strength of synaptic connections, respectively. With the differential equation 1.8, we explain the dynamics of the input potential h_k of population k, and for each population, we utilize the quasi-stationary rate model $A_n(t) = F_n(h_n)$, where F_n is the gain function of the neurons in population n. The outcome is:

$$\tau_m \frac{dh_k(t)}{dt} = -h_k + R \sum_n C_{kn} w_{kn} \int_0^\infty \alpha(s) F_n(h_n(t-s)) ds. \tag{1.8}$$

The effective mean firing rate becomes a smooth sigmoid-like (Figure 1.3) function of injected current where is κ the firing threshold and η is the gain This simplifies to a Heaviside firing rate function in the high-gain limit $\eta \to \infty$.

$$F(I) = \frac{F_0}{1 + e^{-\eta(I-\kappa)}} \tag{1.9}$$

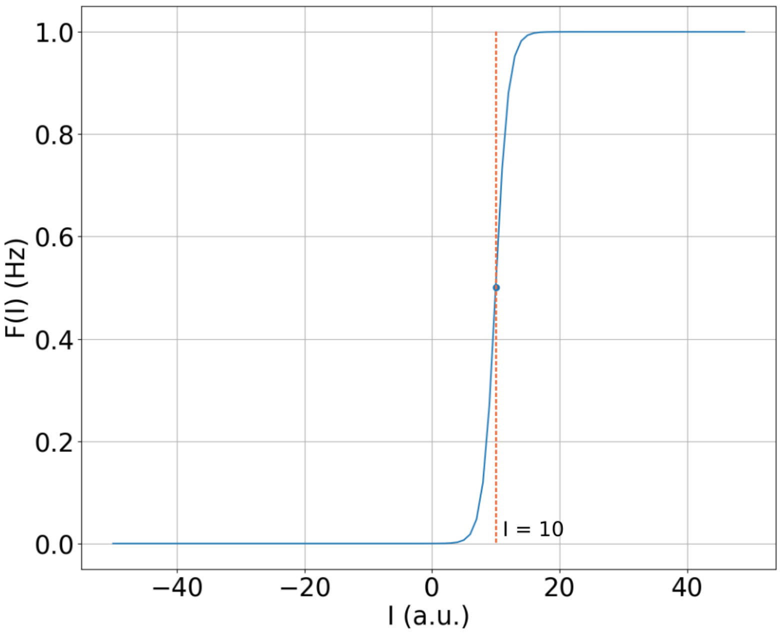

Figure 1.3: A graphical representation of $F(I)$ where $F_0 = 1$, $\kappa = 10$ and $\eta = 1$.

Where is κ the firing threshold and η is the gain This simplifies to a Heaviside firing rate function in the high-gain limit $\eta \to \infty$.

$$F(I) = F_0 H(I - \kappa) = \{F_0 \ if \ I > \kappa, 0 \ if \ I < \kappa\} \tag{1.10}$$

1.5 Learning Models (LTP-LTD, STDP)

Here we introduce the concept of the learning model. SNNs use gradient backpropagation (not our topic) and Hebb's rule. In the following, we will introduce the concept of Hebbian learning and its most well-known variations: LTP, LTD, and STDP.

1.5.1 Hebbian Learning Rule

Hebbian theory is a neuroscientific theory that claims that a presynaptic cell's frequent and sustained stimulation of a postsynaptic cell leads to an increase in synaptic effectiveness. It's an attempt to explain synaptic plasticity or brain neuron adaptation throughout the learning process. Donald Hebb presented it in his 1949 book The Organization of Behavior [12]. Hebb's rule, Hebb's postulate, and cell assembly theory are all names for the rule. Hebb's principle can be thought of as a way to decide how to change the weights between model neurons in artificial neurons and artificial neural networks. When two neurons fire together, their weight increases; when they fire independently, it decreases. Strong positive weights are assigned to nodes that tend to be either simultaneously positive or simultaneously negative, whereas strong negative weights are assigned to nodes that tend to be the reverse. We can describe the Hebbian rule as in equation 1.11

$$w_{ij} = x_i x_j \tag{1.11}$$

where w_{ij} is the weight of the connection from neuron j to neuron i and x_i the input for neuron i.

1.5.2 LTP

Long-term potentiation (LTP) is the most well-known and largely accepted theory of how memory is built in the nervous system. The lack of more plausible explanations, as well as the fact that such use-dependent alterations in the synaptic response suit theoretical expectations so well, are presumably the reasons for their appeal. The first evidence of LTP was found in the hippocampus [13], a region known to be involved in learning processes [14]. It was created in an in vivo experiment by inserting electrodes into a rabbit's perforant route and dentate (Figure 1.4). Anesthetized (Doyle et al., 1996) or freely moving animals can be used for recording [15].

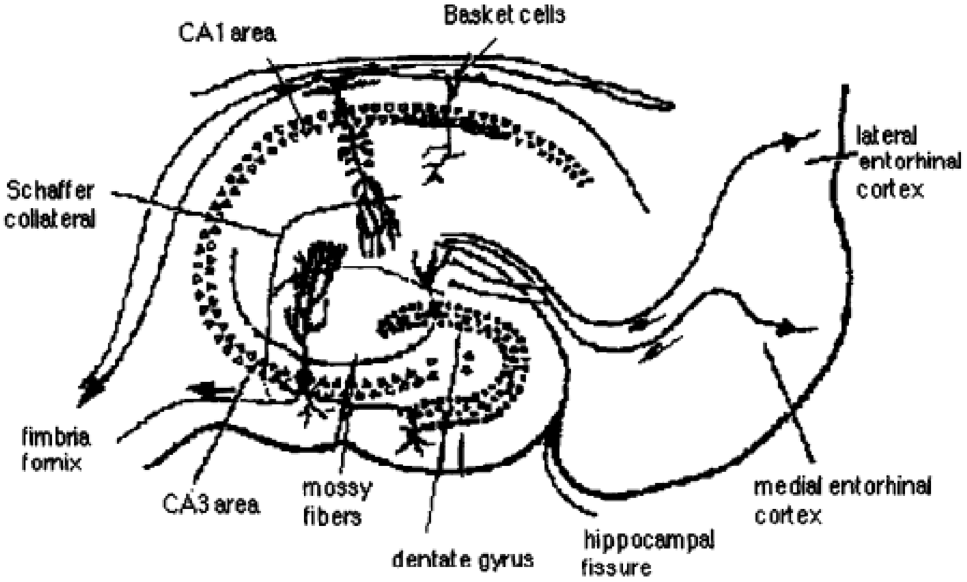

Figure 1.4: Drawing showing a slice preparation of the rat hippocampus and the entorhinal cortex. The section is a coronal cut, showing the trisynaptic pathway from the entorhinal cortex to the dentate gyrus (perforant pathway), from the granule cells of the dentate gyrus to the CA3 area (mossy-fiber pathway), and from the CA3 pyramidal neurons to the pyramidal neurons of the CA1 area (Schaffer collateral/commisural pathway). The CA1 neurons send axons to basal brain nuclei via the fimbria-fornix pathway. LTP can be elicited in all 3 synaptic connections. A common protocol is to stimulate the Schaffer collateral-commissural pathway and to record in the CA1 area. (Adapted from O'Keefe and Nadel, 1978).

LTP is a persistent strengthening of synapses based on recent activity patterns in neuroscience. These are synaptic activity patterns that result in an increase in signal transmission between two neurons that lasts for a long time [16]. Long-term depression is the polar opposite of LTP, resulting in a long-term decline in synaptic strength. Synaptic plasticity, or the ability of chemical synapses to vary their strength, is caused by numerous processes. Because synaptic strength is assumed to be used to encode memories [17], LTP is usually regarded as one of the most important cellular mechanisms underlying learning and memory [16] [17].

1.5.3 LTD

Long-term depression (LTD) is an activity-dependent decline in the effectiveness of neural synapses that lasts hours or longer after a long patterned stimulation in neurophysiology. LTD affects numerous parts of the CNS, with different processes depending on the brain region and stage of development. [18]

The hippocampus and cerebellum have been the most studied for LTD, although there are other brain locations where LTD processes are known. [18] LTD has also been discovered in many types of neurons that release multiple neurotransmitters; nonetheless, L-glutamate is the most prevalent neurotransmitter involved in LTD. [18] During LTD, L-glutamate binds to N-methyl-D-aspartate (NMDARs), $\alpha - amino - 3 - hydroxy - 5 - methylisoxazole - 4 - propionicacid$ (AMPARs), kainate receptors (KARs), and metabotropic glutamate receptors (mGluRs). Strong synaptic stimulation (as in the cerebellar Purkinje cells) or continuous mild synaptic activation (as in the hippocampus) can cause it. Long-term potentiation (LTP) is the polar opposite of LTD. it is the long-term strengthening of synaptic connections. LTD and LTP are both processes that influence neuronal synaptic plasticity. LTD is assumed to be caused mostly by a reduction in postsynaptic receptor density, while presynaptic neurotransmitter release may also be involved. Cerebellar LTD has been suggested to play a role in motor learning. Other plasticity processes, however, are likely to play a role as well. The cleaning of old memory traces may be aided by Hippocampal LTD. [19] [20] NMDA receptors, metabotropic glutamate receptors (mGluR), and endocannabinoids can all play a role in hippocampal/cortical LTD. [21] The phosphorylation of AMPA glutamate receptors and their removal from the surface of the parallel fiber-Purkinje cell (PF-PC) synapse is the outcome of the underlying-LTD molecular mechanism in the cerebellum. [22]

1.5.4 STDP

In vitro and in vivo, STDP has been found at hundreds of synapses in a range of brain locations and experimental circumstances. As a result, STDP is a prevalent type of plasticity that drives a variety of developmental and adult learning processes. However, because many synapses are not affected by somatic spikes, STDP is not a universal plasticity rule, and spike timing is only one aspect of a multifactor learning rule that includes firing rate, spike timing, dendritic depolarization, and synaptic cooperativity. Except for being impacted by depolarization from backpropagation of somatic spikes, STDP is not biochemically separate from classical CDP or local, associative plasticity. As a result, no chemical or genetic alterations that selectively block STDP have been discovered. [24]

STDP is a mechanism for introducing LTP and LTD into the brain [23]. STDP alters synaptic strength as a function of the presynaptic and postsynaptic action potentials' timing (Figure 1.5). Synaptic strength is increased when a presynaptic spike occurs a few milliseconds before a postsynaptic spike. The higher the shift in synaptic weight, the shorter the delay between the two spikes. Synaptic strength is weakened when a postsynaptic spike occurs a few milliseconds before a presynaptic spike. The higher the shift in synaptic intensity, the shorter the time delay between the two spikes.

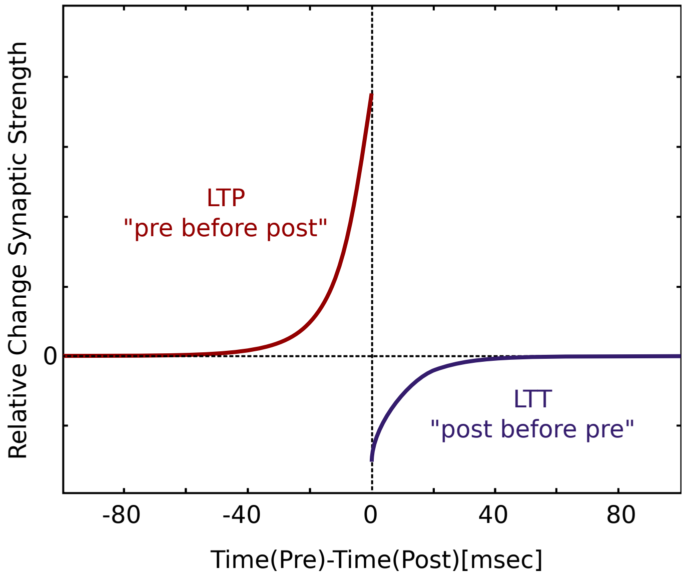

Figure 1.5: Spike-timing-dependent plasticity (STDP). If the presynaptic spike occurs before the postsynaptic spike ("pre before post"), the synapse is strengthened (red, LTP, long-term potentiation). If the postsynaptic spike occurs before the presynaptic spike, the synapses are weakened (blue, LTD, long-term depression). Typically, two action potentials need to occur within at most a few tens of milliseconds for STDP to be recruited.. (Adapted from Flavio Fröhlich, in Network Neuroscience, 2016)

1.6 Spike Coding

To explain the information transfer between neurons, different theories of information representation in the brain referred to as neural codes, have been presented. The ability of brain-inspired spiking neural networks (SNNs) to execute various tasks is dependent on neural coding.

The spiking model differs significantly from past artificial neuron generations, the key information transmitted by spikes is the time relation between them, as a consequence, delivering usable data over a spiking net necessitates conversion from other types to temporal data [25]. Neural coding schemes are used to convert pieces of information into spikes that are delivered to the excitatory neurons. Rate coding, Time To First Spike (TTFS) coding, phase coding, and burst coding are four distinct forms of neural coding methods that were examined and compared [26], in this section, we are going to only cover rate coding and binary coding. Rate coding, sometimes called frequency coding is the most widely used coding scheme in neural network models, the information about the signal is contained in the spike firing rate, its first appearance was after the experiments by ED Adrian and Y Zotterman in 1926 different weights were hanged from a muscle. The number of spikes recorded from sensory neurons innervating the muscle rose as the stimulus weight increased [27], Measurement of firing rates became a typical method of defining the properties of all types of neurons in the following decades, partially due to the relative ease with which rates could be measured experimentally. This method, however, ignores all of the information that could be contained in the precise timing of the spikes and interspike intervals, as well as the intrinsic properties of each action potential. The term "firing rate" has several definitions, referring to various averaging processes such as an average over time or an average over multiple repeats of an experiment.

A temporal average is the first and most commonly used definition of a firing rate. This is essentially the spike count in an interval of duration T divided by T [28]; (see Figure 1.6). The experimenter determines the length T of the time window, which is dependent on the type of neuron recorded from and the stimulus. In practice, several spikes should occur within the time window to obtain reasonable averages. T = 100 ms or T = 500 ms are common values.

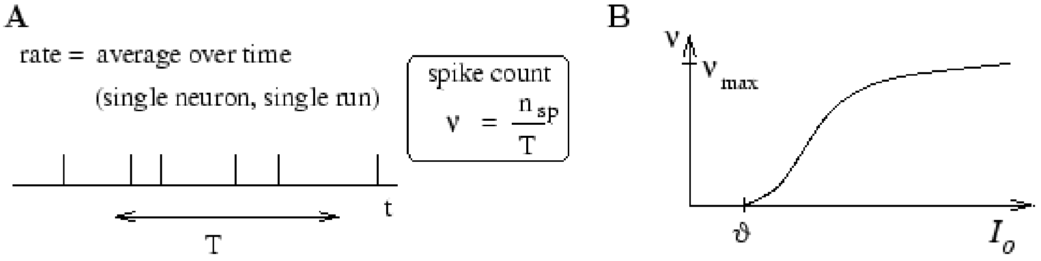

Figure 1.6: **A**. Definition of the mean firing rate via a temporal average. **B**. Gain function, schematic. The output rate ν is given as a function of the total input I0.

This rate definition has been successfully used in numerous preparations, particularly in sensory or motor system investigations. The stretch receptor in a muscle spindle is a classic example (Adrian, 1926). With the force applied to the muscle, the number of spikes generated by the receptor neuron rises. The touch receptor in the leech is another classic example (Kandel and Schwartz, 1991). During a 500-ms stimulation interval, the greater the contact stimulus, the more spikes occur. [29]

The second definition of rate exists, which applies to both stationary and time-dependent stimuli. While stimulating a neuron with some input sequence, the experimenter records from it. The same stimulation sequence is repeated several times, and the neuronal response is recorded in a Peri-Stimulus-Time Histogram (PSTH), as shown in Figure 1.7. The time t is measured relative to the start of the stimulation sequence, and Δt is usually one or a few milliseconds long. The number of spike occurrences $n_K(t; t+\Delta t)$ summed over all repetitions of the experiment divided by the number K of repetitions is a measure of the neuron's typical activity between time t and time $t + \Delta t$. A further division by the Δt interval length yields.

$$p(t) = \frac{1}{\Delta t} \frac{n_K(t; t + \Delta t)}{K} \tag{1.12}$$

Binary coding is a type of encoding in which a neuron is either active or inactive during a period of time, firing one or more spikes throughout that time. This encoding was aided by the discovery that physiological neurons tend to activate when they receive input (a sensory stimulus like light or external electrical inputs).

Because neurons are represented as binary units that can only accept two on/off values, they can benefit from this binary abstraction. It can also be used to analyze spike

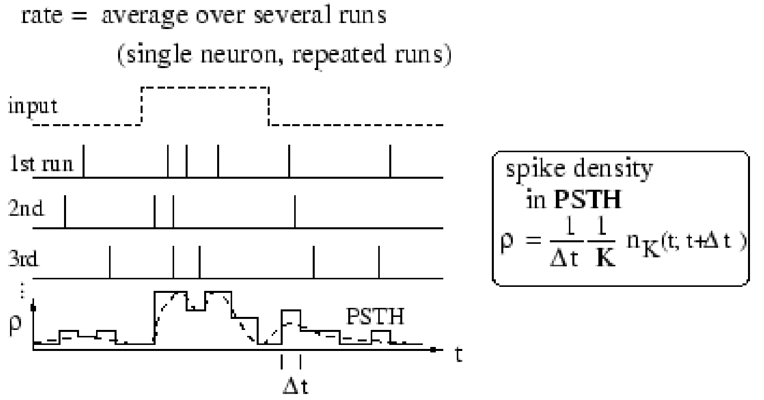
rate = average over several runs
(single neuron, repeated runs)
input
1st run
2nd
3rd
ρ
PSTH
t
Δt
spike density
in PSTH
ρ = 1/Δt 1/K n_K(t; t+Δt)

Figure 1.7: Definition of the spike density in the Peri-Stimulus-Time Histogram (PSTH) as an average over several runs of the experiment..

trains from current spiking neural networks, where spike train classification uses a binary interpretation of the output spike trains.

Binary coding also referred to as temporal coding, assumes that the precise timing of spikes carries information [28] [4], to model this idea, binary numbers can be used to mark the spikes: 1 for a spike, 0 for no spike. Binary coding allows the sequences that have the same firing rate to represent different pieces of information

1.7 Conclusion

In this chapter, we introduced spiking neural networks and different neuron models, as well as rate models and how information can be transferred between neurons.
In the next chapter we are going to represent vectors with a population of spiking neurons called neurovectors and use them to implement drone flight planning.

CHAPTER 2

NEUROVECTORS

2.1 Introduction

In this chapter, we are going to introduce the notion of neurovectors, their representations as a "sinusoidal vector" for the 2D vectors and their generalization in the 3D space to become "sinusoidal matrices" representing 3D vectors. These structures make it possible to represent a vector using a population of neurons, and to implement several mathematical operations that allow us to perform multiple calculations. We will also construct a graphical representation of the neurovectors and their different operations to model and visualize different scenarios for a better understanding.

2.2 Neural Representation of Vectors in the Plane

Touretzky et al. (1993) proposed a neural formalisation based on the representation of vectors as "phasors". A phasor is a $\cos(\theta)$ function in the domain $\theta \in [0, 2\pi]$ which encodes for the magnitude and angle of the vector in such a way that this function is out of phase by the value of the angle and its amplitude is the vector's norm. Hence, A sinusoidal function $f_{v_0}(\theta) = L_0\, cos(\theta - \theta_0)$ is used to represent a planar vector $v_0 = (L_0, \theta_0)$. The function $f(\theta)$ is represented by a population of neurons where each neuron identifies a discrete value of the initial set (θ in this case), and its output (discharge frequency) provides the image of that value. The authors also proposed a new method for vector summation, where the operation of addition of two vectors is equivalent to the addition of their sinusoidal vector.

If two vectors $v_1 = (L_1, \theta_1)$ and $v_2 = (L_2, \theta_2)$ are encoded with the functions

$$f_{v_1} = L_1\, cos(\theta - \theta_1) \tag{2.1}$$

$$f_{v_2} = L_2\, cos(\theta - \theta_2) \tag{2.2}$$

The sum of the vectors v_1 and v_2 is equivalent to the sum of the respective functions. The resulting function will be the sinusoidal f_v of the same frequency as the first two and coding for the resulting vector :

$$f_v = f_{v_1} + f_{v_2} = L_v\, cos(\theta - \theta_v) \tag{2.3}$$

A vector v represented by its sinusoidal f_v can have its polar coordinates θ and L extracted by applying the formulas

$$\theta_v = max\theta f_v(\theta) \tag{2.4}$$

$$L_v = f_v(\theta_v) \tag{2.5}$$

The amplitude of $f_v(\theta)$ is equal to the modulus (length) of v, the equivalence between the sum of two vectors and the sum of their representation in sinusoidal functions [30]

2.3 Generalization of Sine Wave Vectors to Three Dimensions

To understand the concept of sine wave vectors in higher dimensions we have to study the trigonometric function representative of a 2D vector in a polar axis in order to extract the geometric specifications. (Figure 2.1).

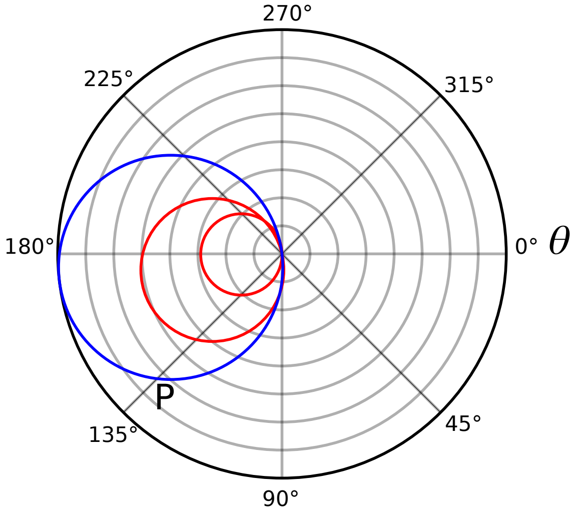

Figure 2.1: A polar representation of a summation of two sinusoidal vectors. Red circles represent operand vectors, and the blue circle is the result vector.

A sine wave function of a vector represents a circle C when drawn on a polar axis. Circle C passes through the origin and carries the vector $\vec{v}$ that it represents. This means the vector traces the diameter of the circle passing by its center where $\vec{v} = 2\vec{OC_c}$, with C_c representing the center of the circle. The point $p = (\rho, \theta)$ (top of the vector) is the origin antipodal of the circle, because the segment $[O, p]$ that passes by these points passes through the center of the circle. If the function that defines a circle $p(\theta) = p_0\, cos(\theta - \theta_0)$ is the sine wave vector of a 2D vector, $v(p_0, \theta_0)$, that means the function that defines a sphere $p(\theta, \sigma)$ is the sine wave matrix (SM) of a 3D vector.

$$\begin{cases} x = pcos(\sigma)cos(\theta) \\ y = pcos(\sigma)sin(\theta) \\ z = psin(\sigma) \end{cases} \tag{2.6}$$

The sine wave function characterizing the 3D vector is the equation of a hypersphere in a spherical space and its obtained in a classic form of a sphere in a cartesian space

$$(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 = r^2 \tag{2.7}$$

Where r is the radius of the hypersphere and (c_x, c_y, c_z) is the coordinates of its center. As the hypersphere carries the vector v, which means $(p_0, \theta_0, \sigma_0)$ is the antipode of the origin, the length of v will be the double the radius r, which means that $r = \frac{p_0}{2}$, also the center of the sphere is the middle of the vector v (Figure 2.2).

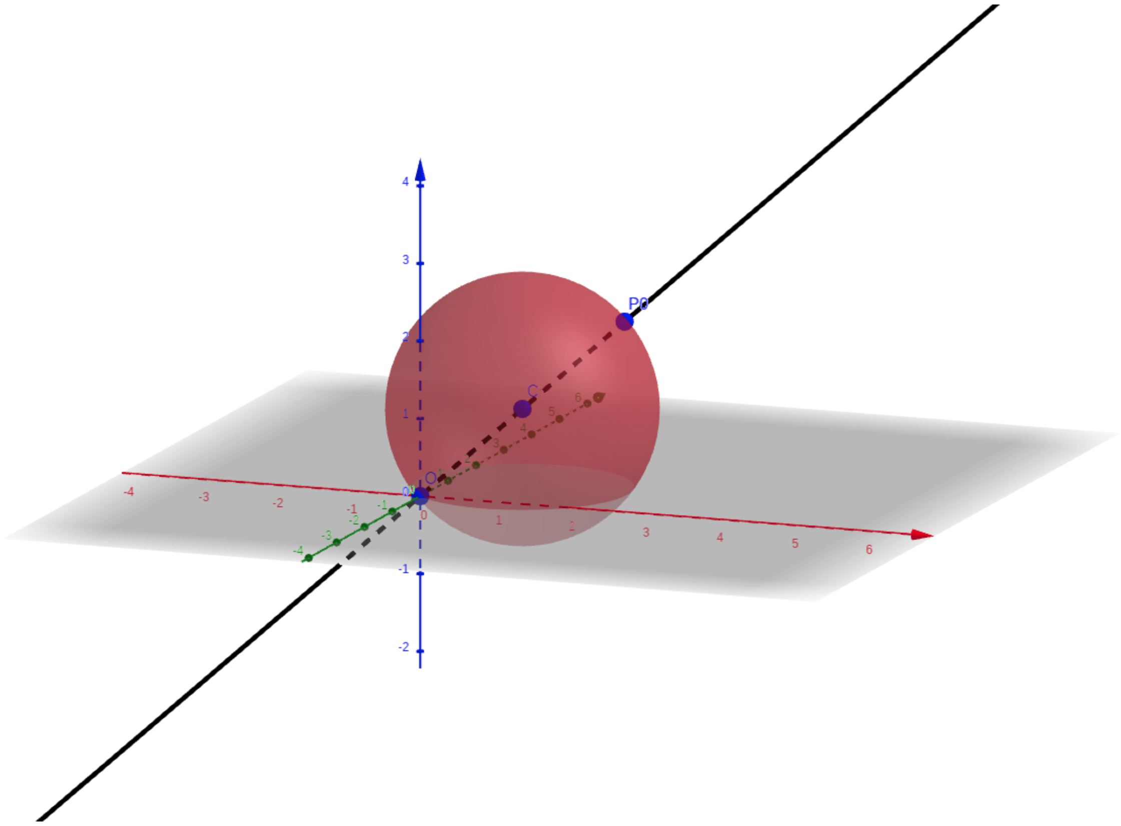

Figure 2.2: Spherical representation of a neurovector

After application of few trigonometric identities, we get [30]

$$M(\theta, \sigma) = p_0(cos\ \sigma\ cos\ \sigma_0\ cos(\theta - \theta_0) + sin\ \sigma\ sin\ \sigma_0) \tag{2.8}$$

Which is the sine wave matrix characterizing a 3D vector and allows the same vectorial operations as is in 2D. A 2D phasor of a vector dimensional represents the dot product of the vector with every vector of the angle θ_i.

$$V_1.V_2 = |V_1|.|V_2|.cos(V_1 V_2) \tag{2.9}$$

Let say we have $V_1(p_1, \theta_1, \sigma_1).V_2(p_2, \theta_2, \sigma_2)$

$$V_1.V_2 = |V_1|.|V_2|.cos(V_1 V_2) = p_1 p_2 (cos\ \sigma_1\ cos\ \sigma_2\ cos(\theta_1 - \theta_2) + sin\ \sigma_1\ sin\ \sigma_2) \tag{2.10}$$

So the equation $M(\theta, \sigma)$ defines the dot product of a known vector $V(p_0, \theta_0, \sigma_0)$ and an unknown vector $V(1, \theta, \sigma)$.

In (1995) Wittmann & Schwegler proposed to discretize the sine wave vector to a table of values with a predefined size, the resolution and quality of the representation rely on the size of that table. It becomes more faithful as the size of the table is large. The authors have proposed to represent every cell of the table with a unique neuron. The value of that cell matches the output frequency of the neuron and every cell has a unique direction. Therefore, a table representing a sine wave is it self represented by a population of neurons where each neuron outputs a value and has a direction θ_i of the original vector (see Figure 2.3).

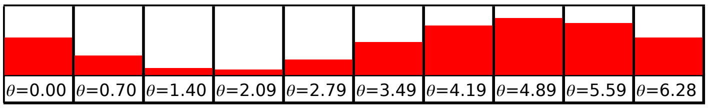

Figure 2.3: A graphical illustration of the frequency output of every neuron in a 2D neurovector

As for 3D neurovectors, we will be using a discretization of the function $M(\theta, \sigma)$ (see Figure 2.4).

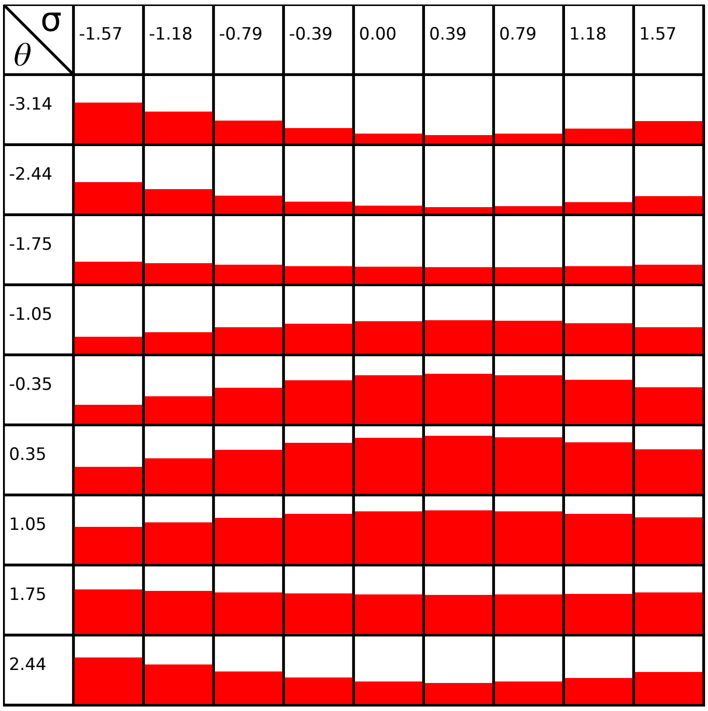

Figure 2.4: A graphical illustration of the sinusoidal matrix where each cell represents the frequency output of every neuron in a 3D neurovector

2.4 Neurovectors Graphical Representation

Since there are numerous possible operations in the context of neuronal vector arithmetic, it is better that we represent them visually for a better readability and understandability. In this section we will be going through our contribution to representing different operations, but before diving any deeper, we start by the representation of a simple neurovector as illustrated in Figure 2.5

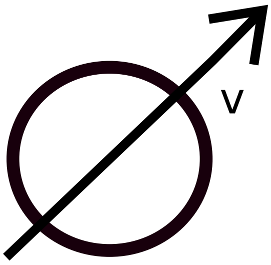

Figure 2.5: Graphical representation of a simple population of neurons (neurovector)

The arrow must always be directed to the top right passing through the center of the circle.

2.4.1 Unit Vector

A unit vector is a vector that has a magnitude 1, and it can be obtained by multiplying the sinusoidal matrix (SM) by $\frac{1}{v_{max}}$ (see Figure 2.6), and v_{max} represents the maximum spiking value of the population which also represents the norm of the vector.

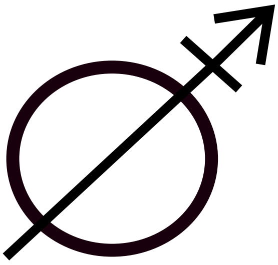

Figure 2.6: Graphical representation of a unit vector

The same constraints as that of the neuron population are required for the schematization of the normalized vector.

2.4.2 Multiplication

Scalar multiplication is one of the fundamental operations which can be achieved by multiplying the spiking value of each cell in the sinusoidal matrix with a factor α (see Figure 2.7).

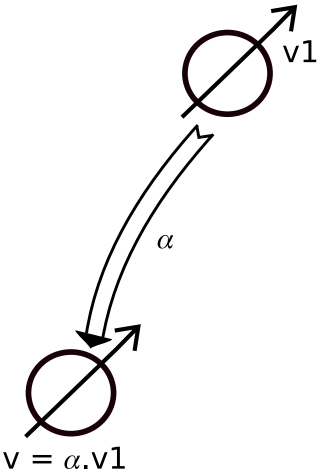

Figure 2.7: Graphical representation of a multiplication

The scalar alpha must always be noted at the right side of the arrow. The arrow symbolizing the transfer of neuronal activity must have a non-zero width. When not noted, the alpha value is 1.

2.4.3 Summation

Summation is also one of the fundamental and trivial operations which can be done by making an element-wise addition of the two sinusoidal matrices that represent the two neurovectors (see Figure 2.8).

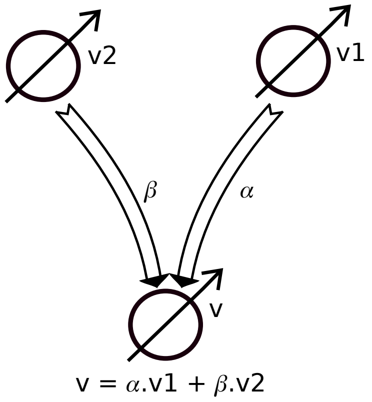

Figure 2.8: Graphical representation of an addition

If not noted, the values of alpha and beta are 1 by default.

2.4.4 Rotation

Rotation is a very important operation. Any rotation is a motion that preserves at least one point, it can be done by shifting the (SM) by θ. and it is illustrated as in the Figure 2.9

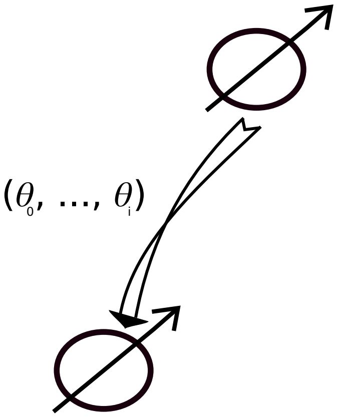

Figure 2.9: Graphical representation of a rotation

Unlike the scalar weight, the angle (or angles) of rotation of the vector is (are) noted on the left of the link. When not noted, the value of the angle is π.

2.4.5 Projection

It is possible to project a population of neurons (neurovector) into a single neuron and it will have a spiking value of the maximum spiking value in that population which represents the norm of the vector and is illustrated as in Figure 2.10

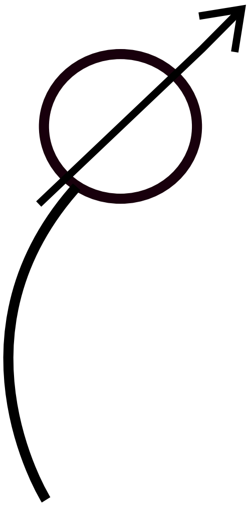

Figure 2.10: Graphical representation of a projection of a population of neurons into a single neuron

2.4.6 Inhibition

inhibitory neurons are neurons that release neurotransmitters to make the post-synaptic neuron less-likely to generate an action potential.

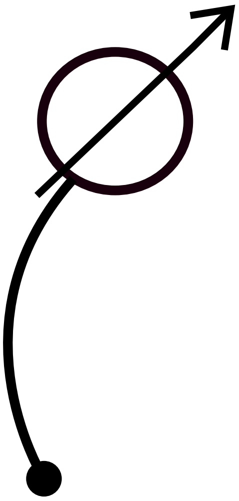

Figure 2.11: Graphical representation of an inhibitory action

The difference between inhibition and projection graphical representation is the circle at the end of the line.

2.5 Biological Plausibility

Recently, it has been discovered that neurovectors are at work within the brain complex of insects. Lyu et al [58] demonstrated that a genus of flies known as drosophila have neurons that are tuned to the allocentric traveling direction in their brain.

The central complex of drosophila includes three parts, the ellipsoid body, the protocerebral bridge and the fan-shaped body. One of a few dozen sets of columnar neurons in the central complex is the EPG cells, when the fly is walking or flying, the full population of EPG expresses a bump of calcium activity in the ellipsoid body, and copies of this bump in the left and right bridge. However, $h\delta B$ cells, which are another type of columnar neurons appear to show activity in coordination with the activity in the EPG cells.

There exists sets of neurons that provide four motion-related inputs to the central complex, these inputs L_1, L_2, L_3, L_4 represent the projections of the traveling vector of the fly onto axes oriented ±45 and ±135 (Figure 2.12.a). The egocentric traveling direction of the fly can be computed by adding the four vectors defined by these projection lengths and angles. As to the allocentric traveling direction, it is calculated by referencing the four projections to the allocentric heading H of the fly before taking the vector sum (Figure 2.12.b). It was shown that these allocentric motion vectors can be represented using sinusoids where the amplitudes and phases match the lengths L1-4 and angles (45 and 135) respectively. Then, the vector sum would be calculated by simply summing the vectors' corresponding sinusoids (Figure 2.12.c).

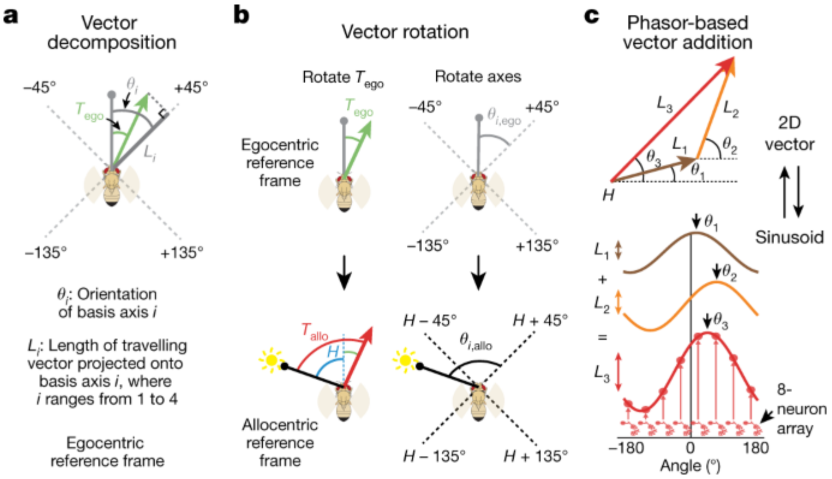

Figure 2.12: **The allocentric travelling direction can be computed by vector rotation and summation, which can be implemented by phasors**. **a**, The travelling direction vector (green) for a fly translating at an egocentric travelling angle, T_{ego}, referenced to its head direction (grey line with a circle), is projected onto four axes oriented ±45° and ±135° relative to the head, yielding four scalars L1–4. The +45° projection is shown. The head direction of the fly represents 0° in this egocentric reference frame. Angles are positive clockwise. **b**, The allocentric travelling direction, T_{allo}, of the fly can be computed either by rotating the egocentric travelling angle (T_{ego}) such that it becomes referenced to the external world (that is, the sun) (left) or, as in the fly circuit, by first referencing the ±45° and ±135ř projection axes to the external world (right) and then taking the vector sum of the four projection vectors. Egocentric vectors are referenced to the external world by adding H, the allocentric heading angle of the fly, to them. **c**, 2D vectors can be represented by sinusoids, and adding sinusoids then implements addition of vectors. Adapted from [58]

There are another two types of columnar neurons which are PFN_d and PFN_v. Separate arrays of PFN_d and PFN_v in the left and right bridge receive extensive monosynaptic and disynaptic input from EPG cells, while also project axons to the fan shaped body where they synapse onto $h\delta B$ cells. The activity profiles across the bridge of the four PFN populations are well fit by sinusoidal functions, and their bumps are poised to represent the four allocentric projection vectors. Although their phases are not offset by ±45 and ±135, their projection anatomy allows their bumps to acquire ±45 and ±135 offsets from the bridge to the fan-shape. And these sinusoids appear to be summed at the level of the $h\delta B$ axons that represent the allocentric traveling direction.

2.6 Conclusion

In this chapter, we explained the concept of neurovectors in both 2D and 3D spaces, and also showed the different operations that can be performed on them and their corresponding graphical representation.
In the next chapters, we are going to construct models to solve different drone flight planning scenarios using neurovectors as well as running simulations and interpreting the results to evaluate the performance of these models.

Part II

Drone Flight Planning

CHAPTER 3 MOTION CAMOUFLAGE

3.1 Introduction

Motion camouflage conceals the optical flow of a moving object, The most common type of motion camouflage and the one most people are familiar with, entails an attacker imitating the optic flow of the background as viewed by its victim. In contrast to the traditional pursuit, this allows the attacker to approach the target while appearing to remain motionless from the target's perspective (where the attacker moves straight towards the target at all times and often appears to the target to move sideways). The attacker chooses a flight path that keeps it on a straight line between the target and a landmark. As a result, the target does not notice the attacker moving away from the landmark point. The looming, the change in size as the attacker approaches, is the sole apparent sign that the attacker is moving. Swaying to simulate plant movements in the wind or ocean currents is one example of how motion can be employed for camouflage. Motion camouflage by reducing optic flow was first observed in hoverflies in 1995 and has since been demonstrated in another insect order, dragonflies, as well as two vertebrate taxa, falcons, and echolocating bats. Because bats hunting at night cannot use the method for camouflage, it has been given the moniker constant absolute target direction to describe its mechanism. This is a good homing strategy, and it's been argued that anti-aircraft missiles could benefit from it as well. Motion can help camouflage, as shown in the leafy sea dragon and several stick insects. These animals add to their passive camouflage by swaying like plants, which allows predators to miss them.

3.2 Motion Signal Minimization (MSM)

Motion Signal Minimization (MSM) can be broken down into two subcategories. The first involves reducing the motion signal, and the second involves reducing the motion itself. Predators use the latter, perhaps the most evident motion camouflage strategy when approaching stationary prey by moving slowly. The hunter sprints towards the prey once it is close enough for the last interception. The first MSM technique is based on the idea of reducing the amount of signal that reaches the prey's visual data processing system. The cuttlefish, with its dynamic stripes, is a well-known example. This animal has the ability to change the patterns on its skin's surface. The cuttlefish decreases the prey's perception of its true speed by generating horizontal stripes that move in the same direction as its axis of movement, which gives it a distinct advantage when hunting. Disruptive camouflage involves manipulating contours and shapes in order to generate a lag effect in the prey's or attacker's perception of movement. Animals use this sort of camouflage to blend in with their surroundings. The tiger's coat, the cheetah's spots, and the cicada's camouflage are all motifs that allow the wearer to confuse the spectator as to the individual's size, direction, and speed of movement. Disruptive camouflage was utilized by the fleets of the belligerents in the First World War to hinder the efficacy of enemy submarines at the turn of the century.

3.3 Disruptive Camouflage

Disruptive coloration (also known as disruptive camouflage or disruptive patterning) is a type of camouflage that uses a highly contrasting pattern to break up the outlines of an animal, soldier, or military vehicle. It's frequently used in conjunction with other crypsis techniques, such as background color matching and countershading; unique cases include concurrent disruptive coloring and the disruptive eye mask present in several fish, amphibians, and reptiles. It appears counterintuitive as a technique of not being noticed, because disruption of edges requires great contrast, making the patches of color themselves noticeable. In 1909, the artist Abbott Thayer anticipated the necessity of high-contrast patterns for successful disruption, and zoologist Hugh Cott predicted it explicitly in 1940. Experimental research has since begun to back up these expectations. When all of the components of a disruptive pattern match the background, it works best. When an animal or a military vehicle has a range of backgrounds, disruptive coloration is a more effective tactic than background matching. Animals that indicate their presence with warning colors (aposematism), on the other hand, adopt patterns that enhance rather than disrupt their outlines. Skunks, salamanders, and monarch butterflies, for example, all have high-contrast patterns that show their contours.

3.4 Mimicking Optic Flow of Background

An attacker can imitate the background's optic flow by choosing a flight path that keeps it on a straight line between the target and a real landmark point or an infinitely distant point (giving different pursuit algorithms). As a result, it does not move away from the target's landmark point, though it does loom larger as it approaches. This is not the same as moving straight towards the target (classical pursuit), which causes visible sideways motion and a visible difference in optic flow from the background. Whether the background is plain or textured, the strategy works [31]. While studying hoverfly mating behavior in 1995, M. V. Srinivasan and M. Davey discovered and modeled this motion camouflage strategy as an algorithm. The male hoverfly appeared to be approaching potential mates using the tracking technique [31]. Males of the Australian emperor dragonfly, Hemianax papuensis, were seen choosing their flight paths to appear stationary to their rivals in 6 of 15 encounters, demonstrating motion camouflage in high-speed territorial battles between dragonflies. They used both real-point and infinity-point strategies to achieve their goals [32] [33].

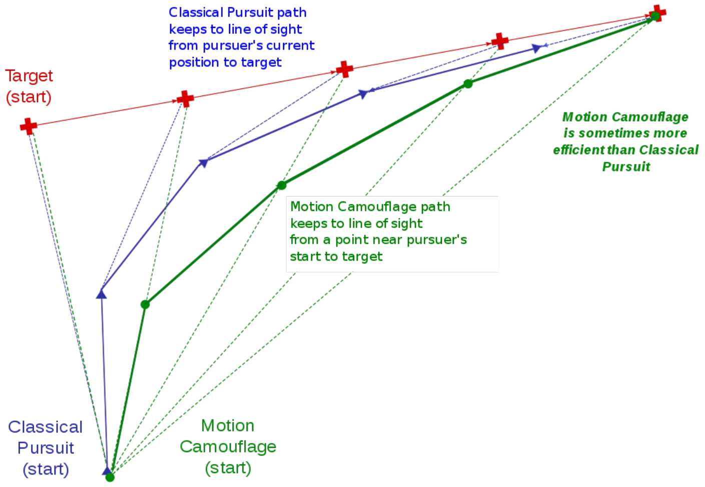

Figure 3.1: Optical flow minimisation using in relation to a fixed-point.

3.5 Fixed-Point Motion Camouflage

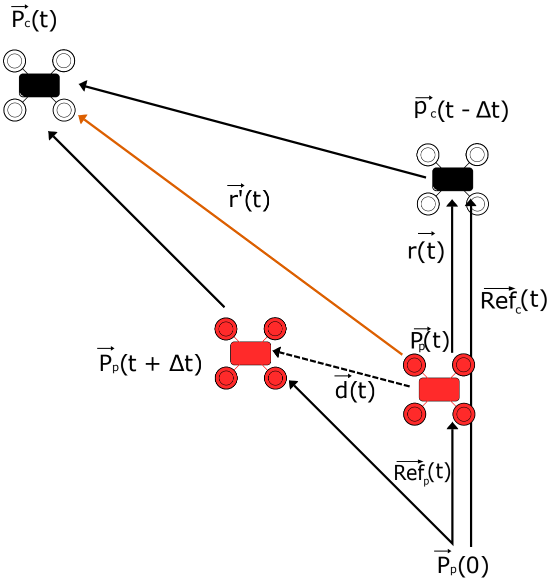

Figure 3.2: Fixed point motion camouflage.

In terms of limiting optical flow, the first technique is the most effective. Indeed, it works regardless of the distance between the predator and the background, unlike the infinite point technique, which requires an endless plane behind the predator as a requirement. The predator's first absolute position will serve as the fixed point for the remainder of the chase. The meaning of the vectors discussed previously is illustrated in Figure 3.2. Two reference vectors $ref_p(t)$ and $ref_c(t)$ are used for this method, which designates respectively the position of the predator and the prey in relation to the fixed point of reference $p_p(0)$. The purpose is to use the predator's information to construct a vector formulation of the elementary displacement vector $d(t)$. The displacement vector $d(t)$ can be obtained from the Chasles with the constraint that this vector must always locate the predator on the line passing through the fixed point and the prey (see Figure 3.2). As a result, it is required to maintain this equality 3.1

$$ref_p(t) + d(t) = \lambda ref_c(t) \tag{3.1}$$

where λ is a parameter that will be described later. The predator can obtain the information $ref_c(t)$ by adding the vector sum of its reference position $ref_p(t)$, which is the inverse vector of the aiming vector towards the fixed point, and the aiming vector $r'(t)$. It is possible to reformulate the previous equation by using vector substitutions deduced from Figure 3.2 by

$$d(t) = \lambda(ref_p(t) + r'(t)) - ref_p(t) \tag{3.2}$$

$$d(t) = \lambda r'(t) + (\lambda - 1)ref_p(t) \tag{3.3}$$

The parameter λ is the expression of parallelism between two vectors in space. When two vectors x and y are parallel, they form a linear analytic relationship known as $x = \lambda y$. Given that the predator must always remain on the line between the fixed point and the prey, the new reference position $ref_p(t)$ must remain parallel to $ref_c(t)$ for each $d(t)$ resulting in the formulation of equation (3.3). This movement must also allow for a reduction in the distance between the predator and the target, allowing for an interception. The position of the predator coincides with the fixed point at the start of the chase, giving it a value of zero. In order for the predator to intercept the prey, $ref_p(t)$ must be equal to $ref_c(t)$, implying that the predator and prey positions must coincide. To achieve this equality, the value must change from 0 to 1 according to a linear relationship of type $\lambda(t) = \alpha t$ where α is the velocity of the approach. In our example, we chose an inverse exponential variation, which increases towards the start of the chase and slows down towards the end, as determined by this differential equation: [30]

$$\frac{d\lambda(t)}{dt} = \alpha(1 - \lambda(t)) \tag{3.4}$$

Neuronal Modeling

The neuron model that enables the fixed-point motion camouflage demands the predator to provide two vectors. The first one is $r'(t)$ which is the reference vector from the current position to the position of the target, and it is extracted by the predator's vision system, the other one, is $ref_p(t)$ which is another reference vector to the fixed point. Each vector is represented with a neurovector and $d(t)$ is calculated by summing and multiplying these neurovectors according to previous equations 3.3. Figure 3.3 visualizes the whole process.

3.6 Infinite-Point Motion Camouflage

Flying predators use the infinity point approach to take advantage of the unlimited distance plane represented by the sky. It has been established that the bat uses this

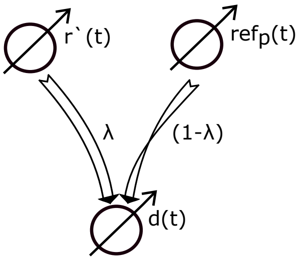

Figure 3.3: A fixed point camouflage modelization using neurovectors.

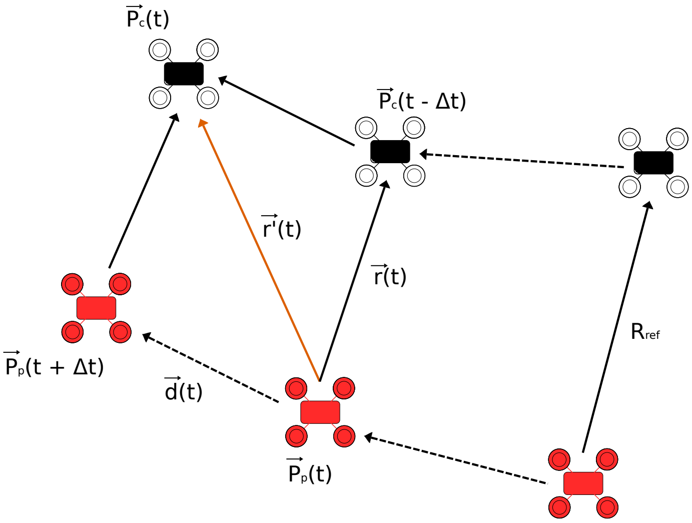

Figure 3.4: Infinite point motion camouflage.

method to hunt flying insects and to save hunting time by following an optimally concealed route [34]. A pursuit with respect to an infinite point is equivalent to following a trajectory in which the vector connecting the prey and the predator is always parallel to the reference vector. Traditionally, this reference vector is built using the predator and prey initial positions (at the start of the chase). This technique is diagrammed in Figure 3.4. The infinite point method uses the same vector notations as to the fixed point method. The purpose is to find an equation for the displacement vector $d(t)$ for each instant t that, when added to the predator's current position, ensures that two essential criteria are met:

- The aiming vector of the prey-predator remains parallel to a reference vector R_{ref}.
- The distance between prey and predator reduces with time to allow interception.

The parallelism criteria between the targeting vector and the reference vector $R_{ref} = p_c(0) - p_p(0)$ imposes this equality, at all times:

$$r(t) = \lambda R_{ref} \tag{3.5}$$

$$d(t) = r'(t) - \lambda R_{ref} \tag{3.6}$$

The fluctuation of the parameter λ identifies the predator's velocity of approach, which is inversely related to the hunting duration, just as it does in the fixed point method. In this scenario, instead of varying from 0 to 1, the value of λ now varies from 1 to 0. This variation is governed by the differential equation:

$$\frac{d\lambda(t)}{t} = -\alpha\lambda(t) \tag{3.7}$$

Neuronal modeling

The neuron model that enables the infinite-point motion camouflage demands the predator to provide two vectors. The first one is $r'(t)$ which is the reference vector from the current position to the position of the target, and it is extracted by the predator's vision system, the other one, R_{ref} is also a reference vector from the current position of the predator to the position of the target but it is not going to change over time as it is only going to be extracted as the chase starts. Each vector is represented with a neurovector and $d(t)$ is calculated by summing and multiplying these neurovectors according to previous equations 3.7. Figure 3.5 visualizes the whole process.

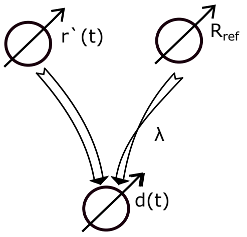

Figure 3.5: An infinite point camouflage modelization using neurovectors.

3.7 Simulations and Results

To evaluate the performance of the proposed models, we have developed a neurovector-based simulator that can emulate neurovectors calculations using the Python programming language, and all the graphics were implemented using the OpenGL graphics library. Using this simulator, we are going to study the fixed and infinite point camouflage and the other scenarios with different prey's movement trajectories and track the predator's behavior and error, as well as the efficiency of neurovectors with a population of 4^2 neurons.

3.7.1 Fixed-Point Motion Camouflage

Straight Movement

In the first phase the prey moves in a straight line with respect to the fixed point at a constant speed (see Figure 3.6), the predator starts at high speed, and then it begins to decelerate as it gets closer to the prey to first keep its camouflage intact and then catch up to it. The deviation of the predator's angle from the line between the fixed point and the prey starts from almost 0° until it reaches about 6° at the end of the simulation, and this is because as the predator gets closer to the prey, it becomes challenging to stay on the line between the fixed-point and the prey.

Helical Movement

In the next scenario the prey moves in a rotational way with respect to the fixed point (rotates around the fixed point without moving away or approaching) at a constant speed (see Figure 3.7), the predator behavior does not change a lot compared to the last simulation. It starts at a slower speed compared to the straight movement, and then it decelerates as it gets closer to the prey in order to keep its camouflage intact and then catch up to it. The deviation of the predator's angle from the line between the fixed point and the prey starts from almost 0° until it reaches about 2° at the end of the simulation, with an average of 0.76°. This movement has better results compared to the last one, and this is due to the fact that the distance between every fixed-point/prey line is so much lower and constant, this makes it easier for the predator to stay on the line.

Random Movement

For this last scenario the prey moves in a random trajectory at a constant speed (see Figure 3.8), the predator's velocity profile is noisy, the explanation for the sudden accelerations of the predators comes from the fact that it is probable that the prey moves away from the fixed point when it starts randomly moving, the deviation of the predator's angle from the line between fixed point and prey, reaches 10° with an average of 1°.

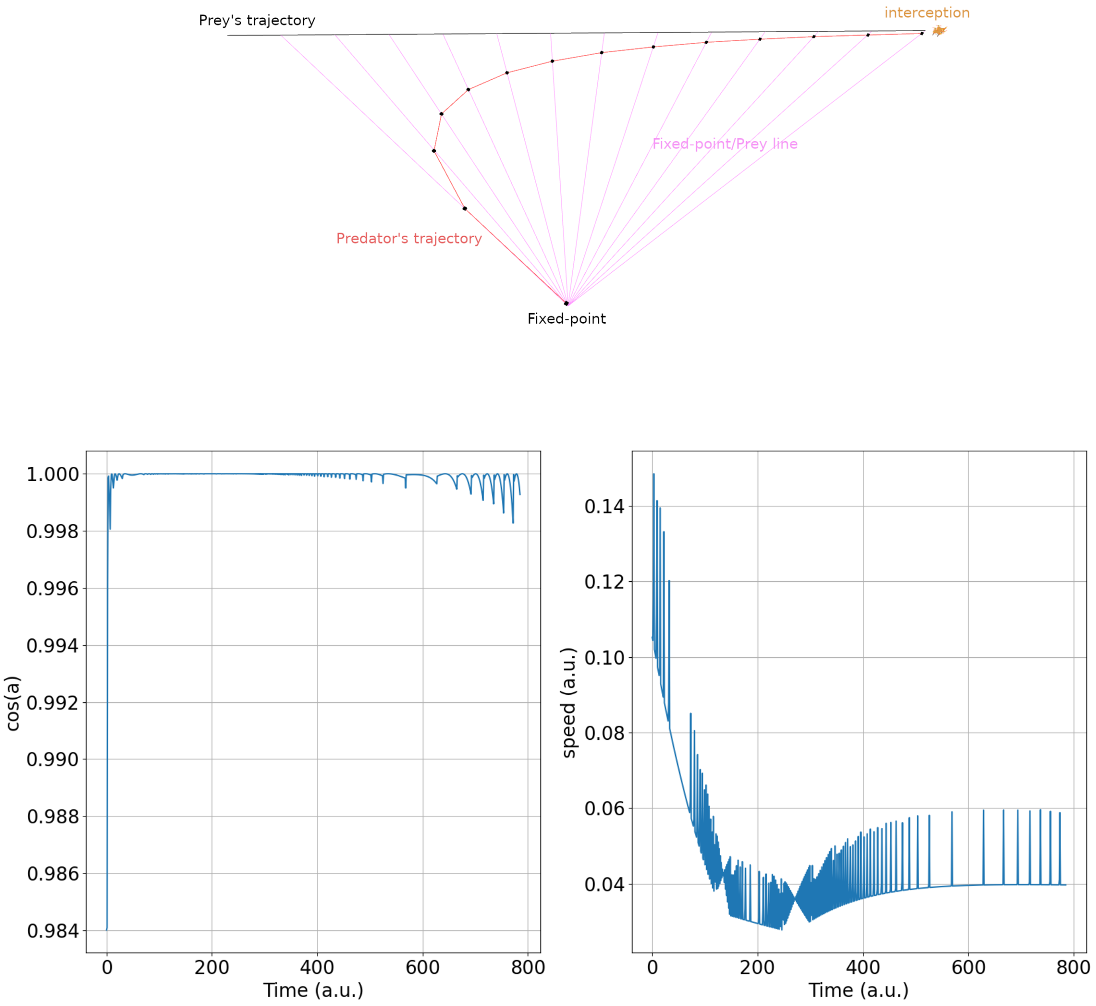

Figure 3.6: Straight movement camouflage relative to a fixed point. On the right, the velocity profiles of the predator. On the left, the cosine of the angle deviation of the predator from the line of camouflage.

3.7.2 Infinite-Point Motion Camouflage

Straight Movement

In the first phase the prey moves in a straight line with respect to the fixed point at a constant speed (see Figure 3.9), the predator's behavior is similar to its behavior from the fixed point simulation. It starts at high speed, and then it begins to decelerate as it gets closer to the prey to first keep its camouflage intact and then catch up to it, the only difference is the angle results became more precise as the deviation of the predator's angle starts from almost 0° until it reaches about 2° at the end of the simulation which is x3 times better than the fixed point.

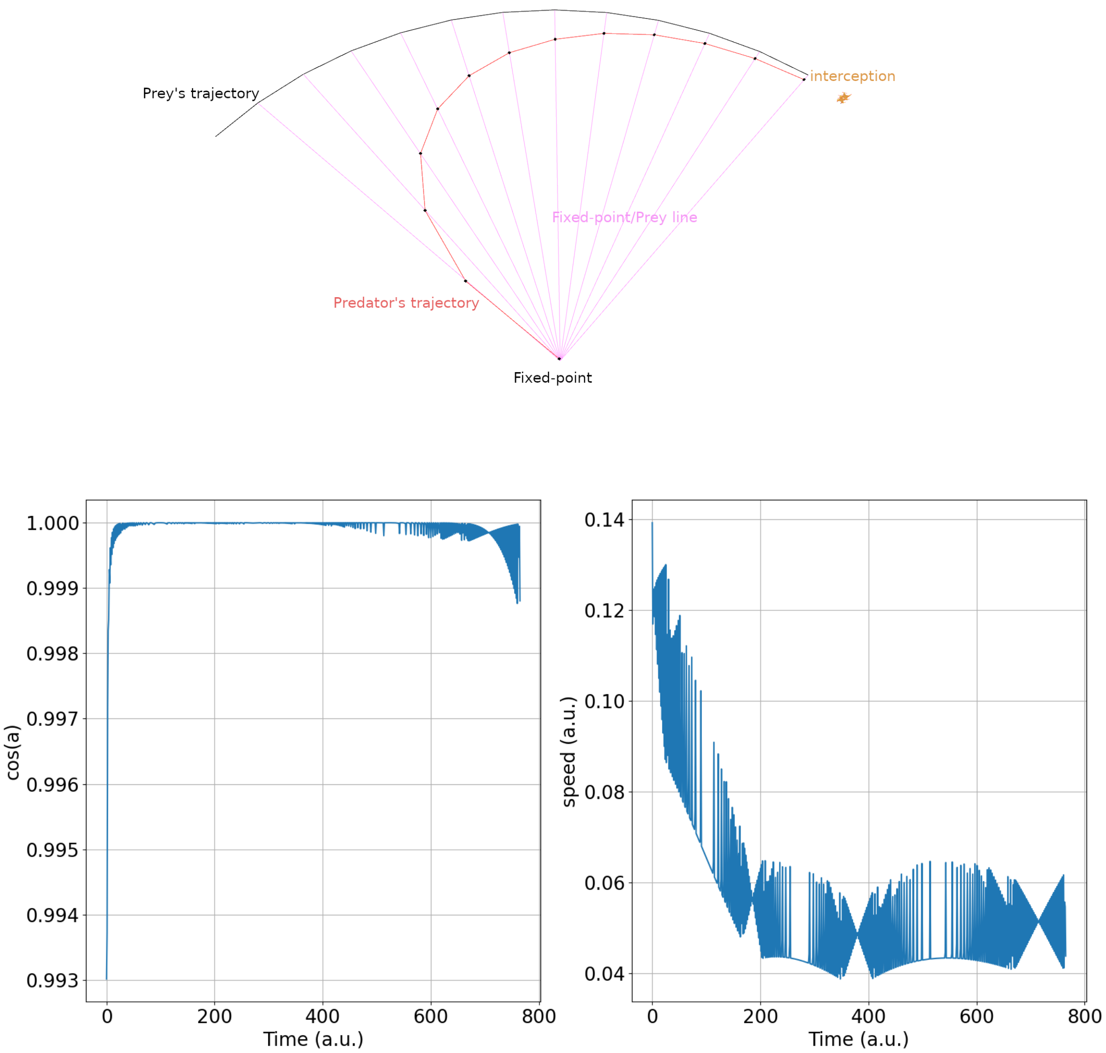

Figure 3.7: Helical movement camouflage relative to a fixed point.

Helical Movement

In the next phase, the prey moves in a rotational way at a constant speed (see Figure 3.10), the simulation results got better angle-wise compared to the fixed point. It starts at high speed, and then it decelerates as it gets closer to the prey, the deviation of the predator's angle from the line between fixed point and prey starts from almost 0° until it reaches about 3.62° at the end of the simulation.

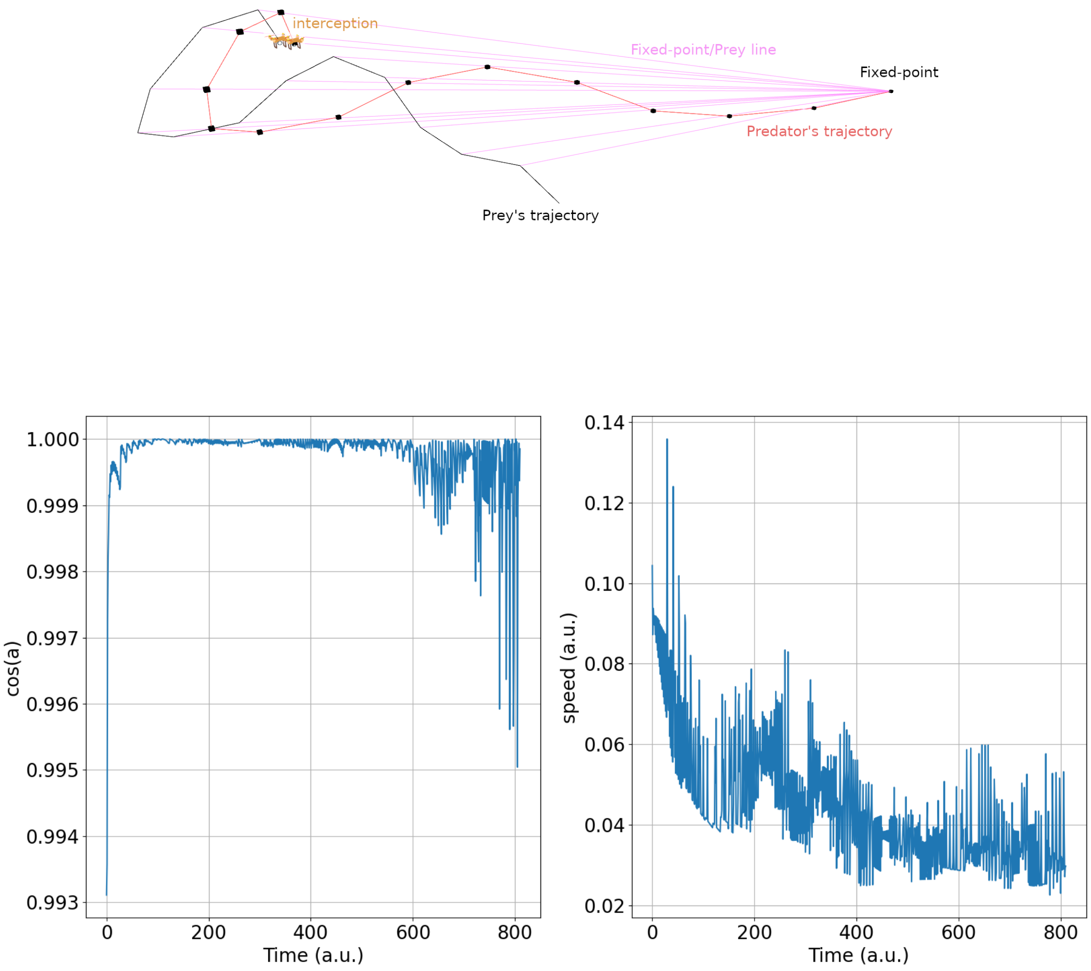

Figure 3.8: Random movement camouflage relative to a fixed point.

Random Movement

For this last phase, the prey moves in a random trajectory at a constant speed (see Figure 3.11), The predator's velocity profile is noisy, as well as the deviation of the predator's angle reaches 5°, compared to random movement from the fixed point simulation this is x2 times more precise.

3.8 Conclusion

Two neural models for stealth hunting with motion camouflage are presented in this chapter. The first model enables the predator to conceal itself by acting as a stationary point in the immediate surroundings. In the second illustration, the predator represents an infinite point in the background of the animal. It has been demonstrated through experiments that vector equations controlling the dynamics of flight and hunting can be

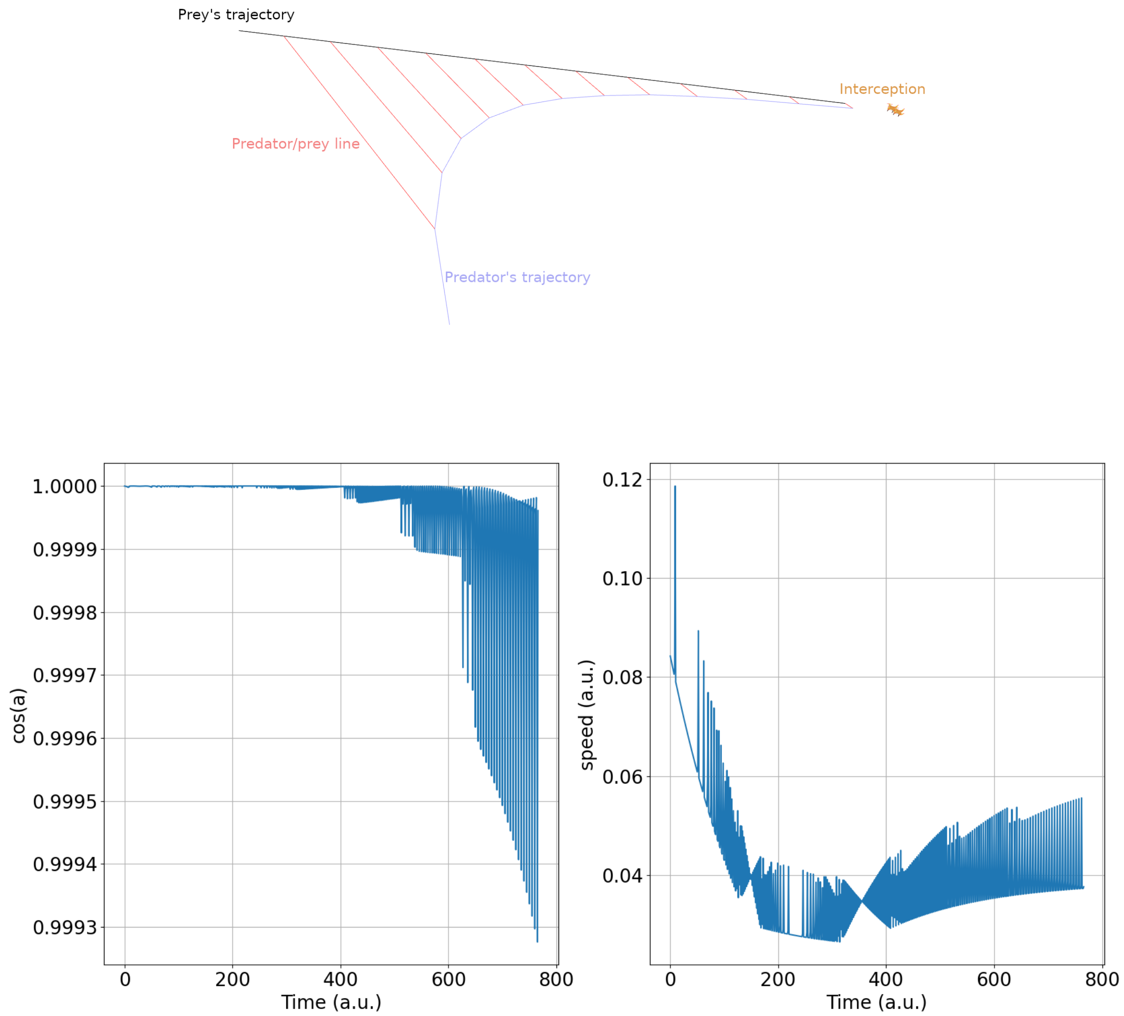

Figure 3.9: Straight movement camouflage relative to an infinite point. On the right, the velocity profiles of the predator. On the left, the cosine of the angle deviation of the predator from the line of camouflage.

applied to populations of neurons that form sinusoidal matrices. As long as there are at least 16 neurons per population of neurons, flight pathways are extremely precise, effective at camouflaging, and durable. The results show that the camouflage model for an infinite point has a higher precision.

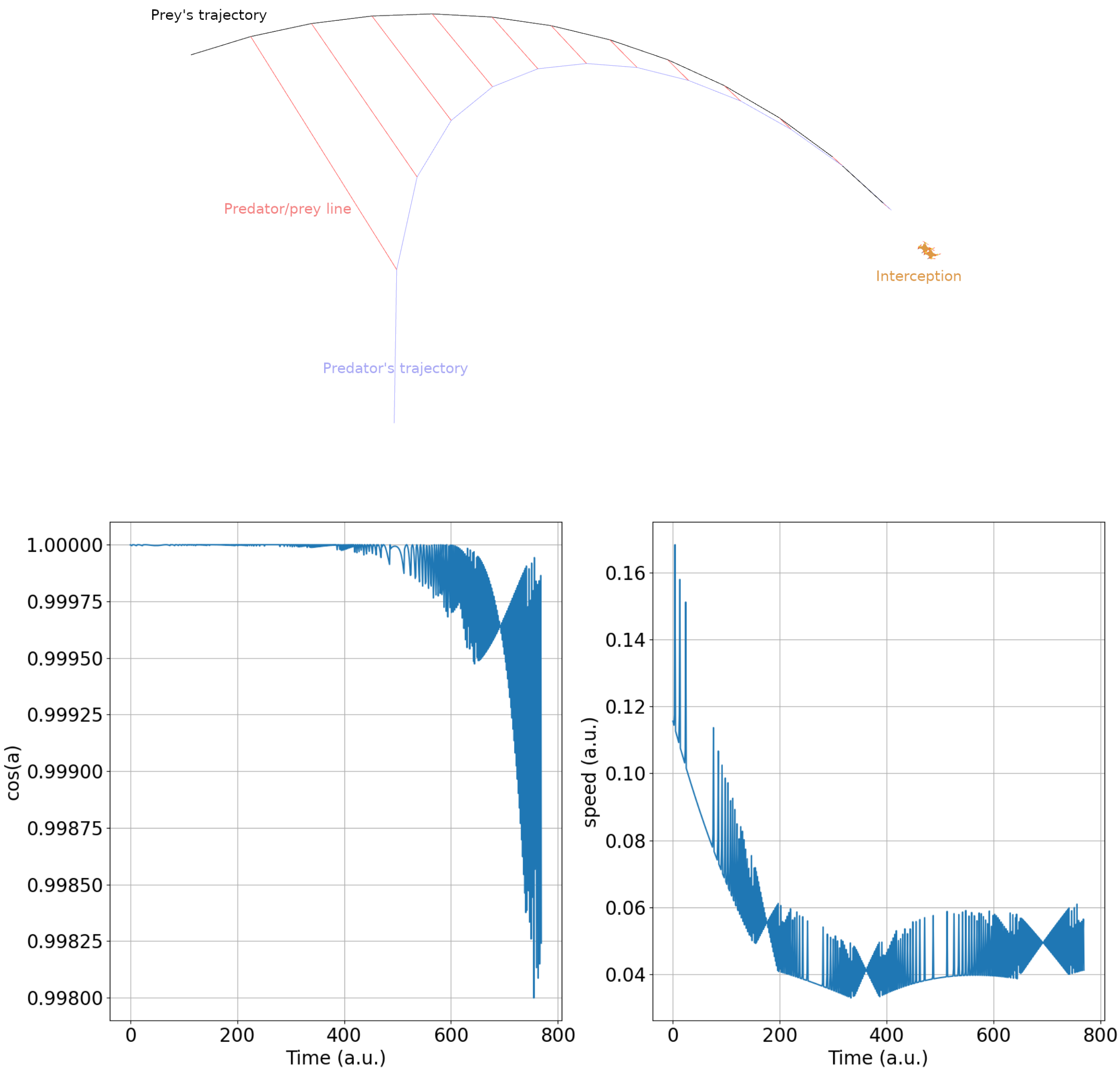

Figure 3.10: Helical movement camouflage relative to an infinite point.

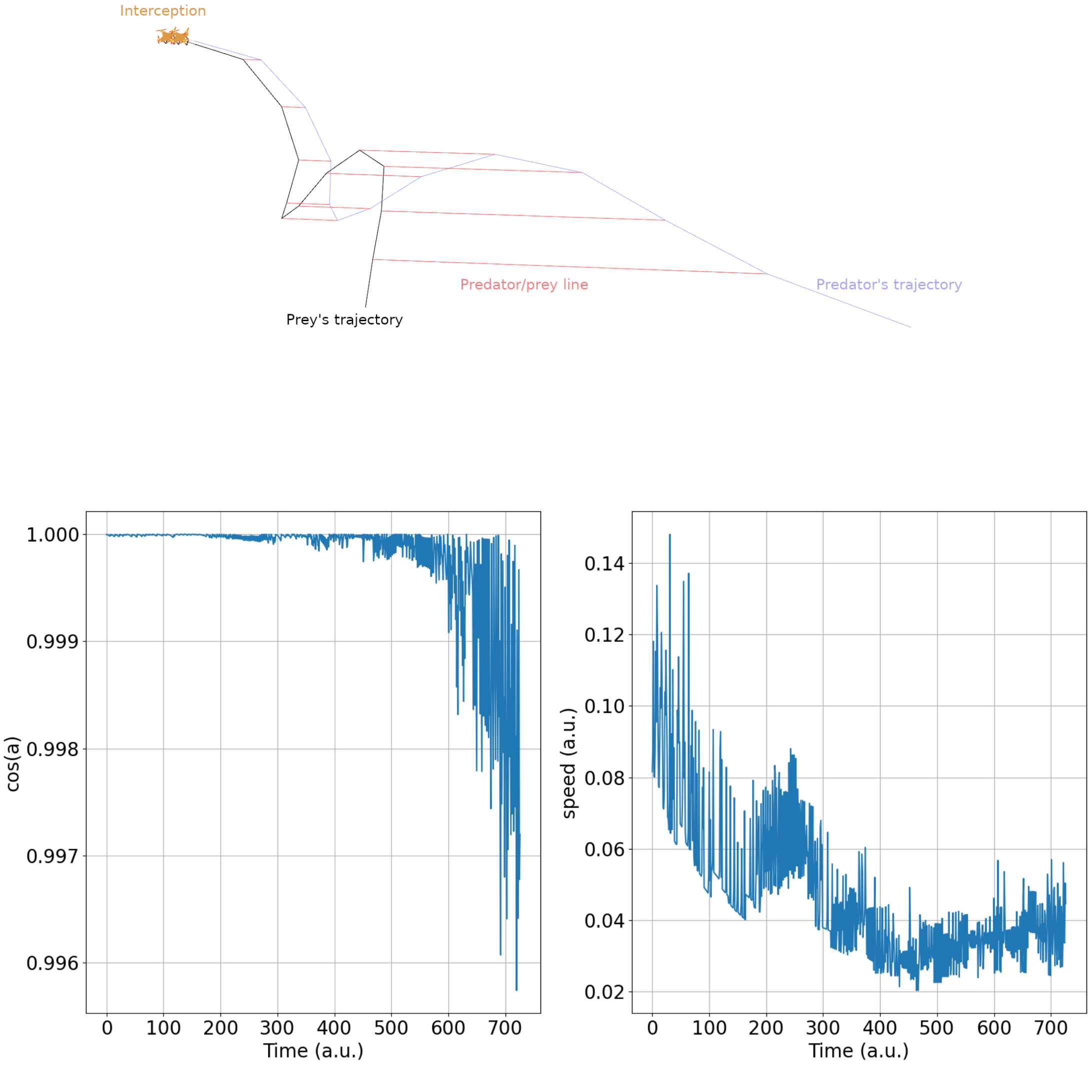

Figure 3.11: Random movement camouflage relative to an infinite point.

CHAPTER 4

TARGET TRACKING

4.1 Introduction

One of the many applications of UAVs is dynamic target tracking, where one UAV or a swarm of UAVs are given the mission of following and tracking a given target that moves freely in the environment. In this chapter, we will address the case of a single UAV tracking a single target. However, the target could be considered to be a person or even a car and move on the ground as shown in Figure4.1.a, or it could have more freedom and move in the 3D space. For example Figure 4.1.b, where a UAV is tracking another UAV.

Aerial tracking of both ground and flying targets can be a game changer in various complex missions, especially in military fields [35], or in civilian fields such as surveillance, search, and rescue [36], where the UAV is given the task of hovering over the target at the desired distance or to maintain the target within a certain distance.

We consider that the UAV can determine its own altitude and the position of the target through advanced monitoring systems. Using the provided visual information, the UAV is actually mimicking flying creatures' behaviour such as birds and insects [37], that depends only on their visualization while flying. In this chapter, we are going to extract two separate sets of kinematic equations to control the UAV's movement and allow it to track the target, and then propose and test the neuronal models for both scenarios.

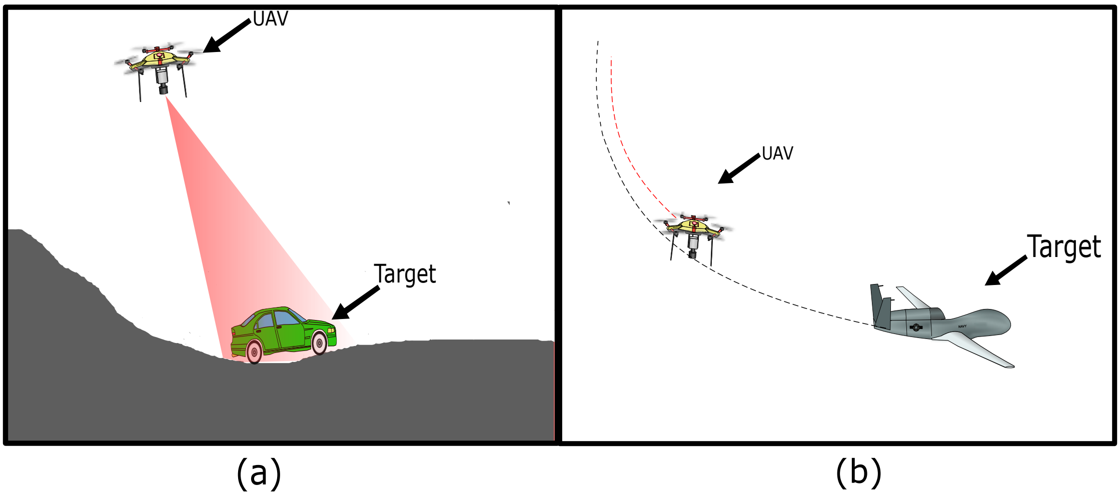

Figure 4.1: UAV for target tracking. (a) Ground target, (b) Flying target.

4.2 Vector Model

In order to exploit the neurovectors in the earlier mentioned scenarios, first, we need to provide the two vector models that schematize the entire structure, so that every single vector will be represented by a neurovector. In this section, the entire process is described.

4.2.1 Ground Target

In this first scenario, we consider the target to be moving on the ground, this last has the freedom to move in any direction, with any velocity and acceleration. From one side, knowing that the UAV has the privilege of moving in 3D space, it will always be superior to the ground target considering that the road is not straight all the way, so all the ups and downs of the road will not be in his favour. But on the other side, the UAV also needs to consider this and adapt its position to avoid any possible collision.

In order for the UAV to determine its next position, it only needs the position of the ground target which is already provided. Taking into account the allowed height limit h_{min} from the ground, and the distance l_d to keep from the target, the vector equations that allow the UAV to make the next step will be derived.

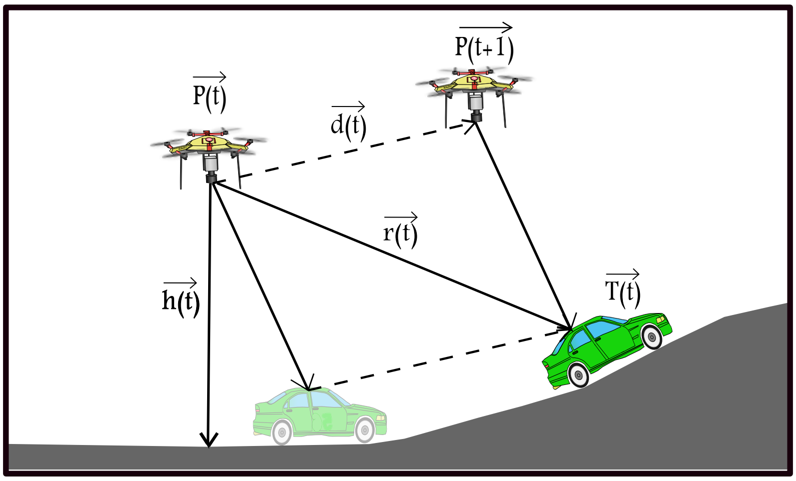

Figure 4.2: vector model for ground target tracking .

First, let's define the set of three-dimensional vectors from Figure 4.2 and the variables needed in the process: $\vec{P(t)}$ designates the absolute position of the UAV at time t, and $\vec{T(t)}$ designates the absolute position of the target at time t. $\vec{h(t)}$ represents the altitude vector that is the height at which the UAV flies at time t. The Target-UAV reference vector $\vec{r(t)}$ can be extracted from the difference :

$$\vec{r(t)} = \vec{T(t)} - \vec{P(t)} \tag{4.1}$$

For the UAV to follow the target and take the shortest path for energy saving, it should move according to $\vec{r(t)}$, taking into account the distance l_d that should be kept from the target, we derive the vector $\vec{dr}$ as follows :

$$\vec{dr} = \lambda \vec{r(t)} \tag{4.2}$$

where $\lambda = 1 - \frac{l_d}{\left\|\vec{r(t)}\right\|}$, and $\left\|\vec{r(t)}\right\|$ is the length of the vector $\vec{r(t)}$.

However, the UAV needs to keep its altitude always above a height limit h_{min} from the ground. For this purpose, we introduce two vectors $\vec{h_m}$ and $\vec{dh}$. $\vec{h_m}$ is parallel to $\vec{h(t)}$, with opposite direction and with a magnitude equal to h_{min}, and dh is defined as follows :

$$\vec{dh} = \vec{r(t)} + \vec{h_m} \tag{4.3}$$

Hence, the movement vector $\vec{d(t)}$ is given as follows :

$$\vec{d(t)} = (1 - \alpha)\vec{dr} + \alpha \vec{dh} \tag{4.4}$$

where $\alpha = \frac{1}{\left\|\vec{h(t)}\right\| - h_{min}}$, so when the altitude of the UAV drop, λ get bigger, and increasing the altitude is prioritized over conserving the distance from the target.

4.2.2 Flying Target

In this scenario, the target has the freedom to move in the 3D space, in any direction in the sky, with any velocity. And the UAV is only limited by the distance l_d that should be kept from the target. Hence, the UAV's movement vector $\vec{d(t)}$ will always be the vector that represents the shortest path to the target, so the UAV would avoid any unnecessary movement that affects its energy availability. To find $\vec{d(t)}$ we only need the vector $\vec{T(t)}$ of the target's position at time t and is derived as follows :

$$\vec{d(t)} = \lambda \vec{r(t)} \tag{4.5}$$

where $\lambda = 1 - \frac{l_d}{\left\|\vec{r(t)}\right\|}$, $\vec{r(t)} = \vec{T(t)} - \vec{P(t)}$ and $\left\|\vec{r(t)}\right\|$ is the length of the vector $\vec{r(t)}$.

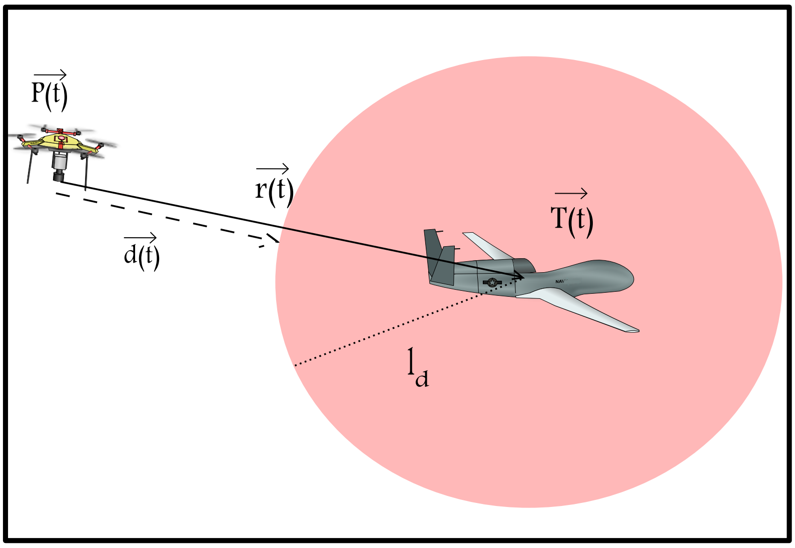

Figure 4.3: vector model for flying target tracking.

4.3 Neuronal Modelling

The vector equations 4.1 and 4.2 extracted previously allow the UAV to perform an immediate shift to the next position that allows it to track effectively the target; these equations involve three-dimensional vectors that are straightforwardly calculated by the UAV and can be represented using neuron populations via neurovectors. Next, we're going to present the neuron model for both ground target and flying target tracking.

4.3.1 Ground Target

The neuron model that enables the tracking of the ground target demands the UAV to provide two vectors. The first one is $\vec{r(t)}$ which is the position of the target compared to its own position, and it is extracted by the UAV's vision system, the other one is $\vec{h_m}$ which is a unit vector $\vec{h}$ directed from the ground up and multiplied by h_{min}. Each vector is represented with a neurovector, and $\vec{d(t)}$ is calculated by summing and multiplying these neurovectors according to equations 4.2, 4.3 and 4.4. Figure 4.4 summarizes the whole process.

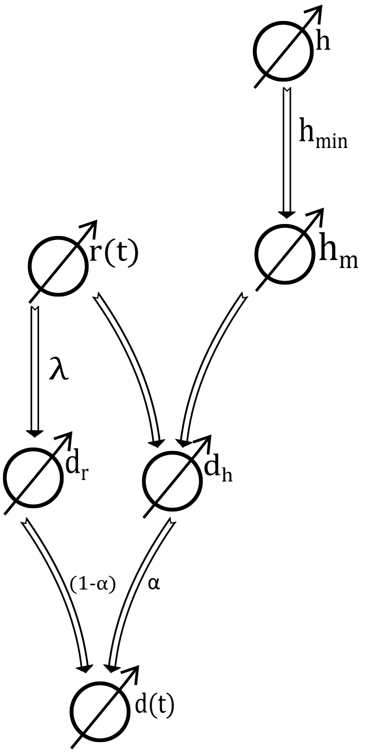

Figure 4.4: Neuronal model the for ground target tracking.

4.3.2 Flying Target

For this kind of tracking the only vector that the UAV needs to provide is $r(\vec{t})$ the reference vector that represents the target's position, this last is represented with a neurovector that is multiplied by a scalar according to equation 4.5 as shown in Figure4.5.

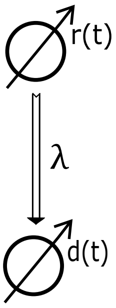

Figure 4.5: Neuronal model the for flying target tracking.

4.4 Simulations and Results

To test the performance of the proposed neuronal models, we set up a few simulation scenarios for the tracking of the two types of targets. For the following simulations we used neurovectors with a population size of (10 × 10).

4.4.1 Ground Target

For this scenario, the ground target starts off on the flat ground moving with a fixed speed, than getting to the up-road, it starts to slow down to stabilize on the top, and it picks up the pace as it goes down to go back to its initial velocity. In this simulation, we set the UAV's height limit $h_{min} = 6(a.u.)$, and the distance to keep from the target $l_d = 10(a.u.)$.

Figure 4.6 shows the trajectory of both the target and the UAV, and we can see that at the beginning the UAV increased its altitude to respect the height limit, then he adapted to the target movement and speed while keeping the distance l_d.

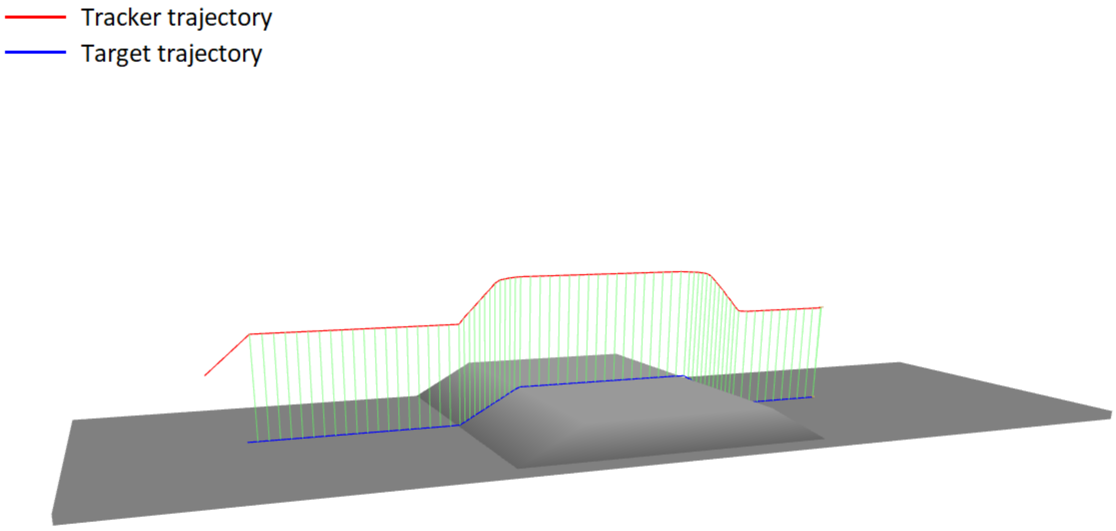

Figure 4.6: Simulation snapshot showing the trajectory of both the UAV and the ground target.

Figure 4.7(a) shows the speed of both the target and the UAV. At first, as the UAV takes off, its speed is relatively high to reduce the distance from the target to reach l_d, then the UAV picks the target's pace and moves according to it, getting to the up road, as the target starts decreasing its speed, the UAV does too to keep the distance while trying to stay above the height limit. In Figure 4.7(b) we can see that the distance is a little bit above l_d, and that gap is a compromise that the UAV makes to stay above the height limit at all times.

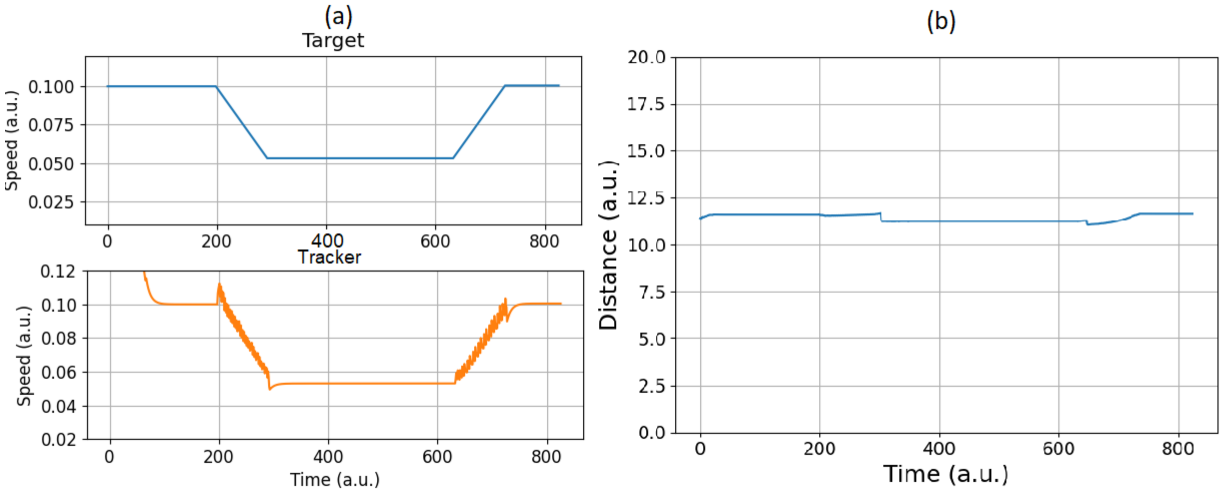

Figure 4.7: Numerical result for ground target tracking: (a) Speed of the UAV and the ground target, (b) Distance between the UAV and the target.

4.4.2 Flying Target

In this scenario, the target is flying in the 3D space with a fixed velocity and moving according to a random movement, where an initial random direction vector is picked and after a certain amount of time, the vector is rotated with a given angle around one of the three axes. The distance l_d in this simulation is equal to $5(a.u.)$.

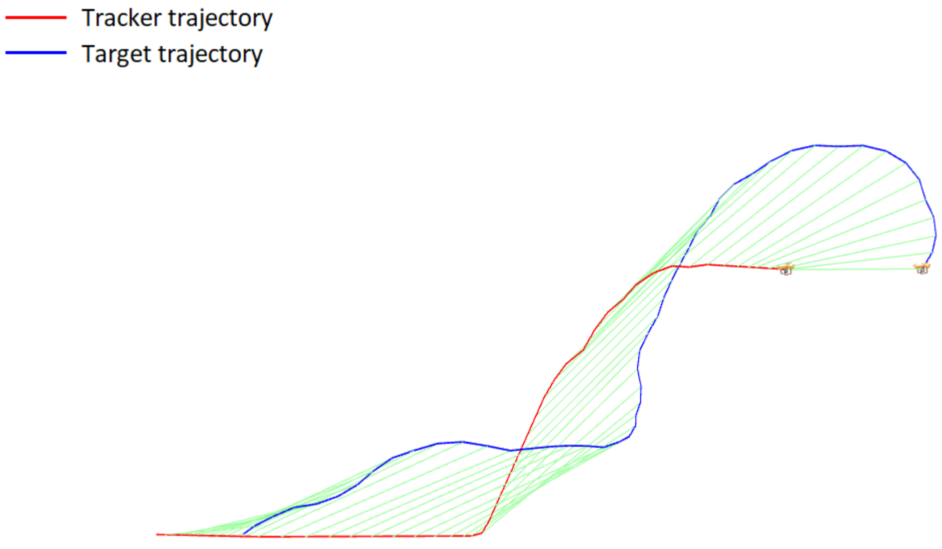

Figure 4.8: Simulation snapshot showing the trajectory of both the UAV and flying target.

Figure 4.8 shows the trajectory of both the UAV and the target, even if the target is moving openly within the space the UAV only makes the minimum necessary movement to keep the distance l_d in between, and that is what shows Figure 4.9 wherein (a) we can see that the speed of the UAV is always lower than the speed of the target and that because when the target performs a helicoidal movement where he moves around the UAV, this last doesn't need to move as long as the distance is respected, and it's what is shown in Figure 4.9 (b). Despite the random movement of the target the distance is almost equal to $5(a.u.)$ at all times.

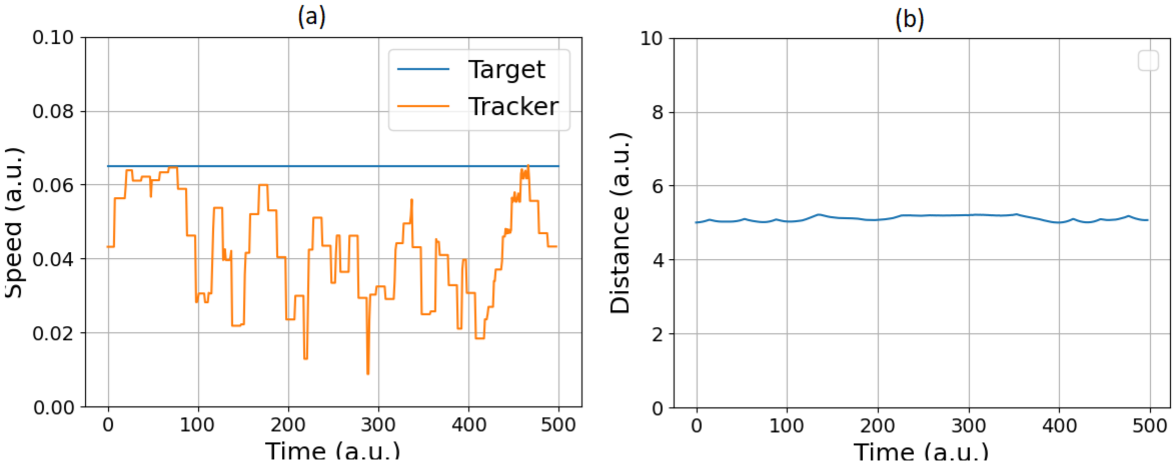

Figure 4.9: Numerical result for flying target tracking: (a) Speed of the UAV and the flying target, (b) Distance between the UAV and the target.

4.5 Conclusion

In this chapter, we presented two neuronal models for target tracking, the first one handled the case where the target is moving on the ground so the UAV had to keep a constant distance from the target to avoid of any possible collision with the ground (ups and downs of the road), and in the second one, the target was moving in the air, so he had the freedom to move in the 3D space.

From the simulation results, we could see that the UAV in both tracking scenarios was successful in tracking the target by keeping a constant distance from the flying target, and for maintaining a relatively close distance and avoiding any collision with the ground target. Hence, the neurovectors representing the vector models were precise and effective.

CHAPTER 5

AUTONOMOUS 3D DEPLOYMENT OF AERIAL BASE STATIONS FOR WIRELESS COVERAGE IN CELLULAR NETWORKS

5.1 Introduction

In recent years, the demands for higher data rates and better quality of service have been continuously growing [38], this goes back to the increase in the number of users, and the expansion of cellular communication use into different fields. As a result, the network capacity should be enhanced in order to satisfy all the needs. Instead of deploying a higher number of terrestrial base stations (TBSs) which could be a waste of resources to invest in an infrastructure that will only be required for a short amount of time in peak hours of the day [39]. A promising solution has been open to discussion over the past few years, which is the deployment of unmanned aerial Vehicles (UAVs) acting as aerial base stations (ABSs).

Because of their high mobility, and adjustable altitude, ABSs can avoid obstacles and increase their probability of establishing line-of-sight (LoS) communication links with the ground users which improves the network quality [40]. Hence, in urban areas, ABSs are superior to TBSs because they can adapt to the ground users' requirements and their movement. Another use case is when the TBS is damaged or not completely functional due to a disaster, e.g. earthquake [41], or when the network is overloaded. And due to their small size and fast movement, they can be deployed to access unreachable places in a very short period of time. Also, they can be very practical to provide uncovered areas, e.g. suburban areas with network connectivity.

The deployment of ABSs can be challenging. taking into account the power consumption, coverage optimization and interference management [42], the ABSs should find the optimal positions for them to provide the demanded quality of service for the mobile ground users. In this chapter, we present a neurovectors electrostatic-based solution for the 3D deployment of a swarm of ABSs to provide wireless coverage, the proposed neuron model will be tested and discussed.

5.2 Related works

There have been many interesting works that addressed the placement problem of ABSs, most of them studied the optimal position of a single ABS for either static or mobile ground users, for a group of non-vehicle users, the authors in [43] consider the different urban environment conditions and the maximum allowable path loss to derive the optimal altitude of the UAV that maximize the coverage on the ground. In [44], the placement problem was decoupled, so the placement in the vertical and horizontal dimensions were separately studied. First, the optimal altitude that results in the maximum coverage region was found. Then, the optimal 2-D position was determined by modelling the problem as a circle placement problem and a smallest enclosing circle problem. Finally, the altitude is recalculated to maximize the number of covered users and minimize the transmit power. And in [45], the UAV was used as a relay between the base station (BS) and the users, the structure of the UAV-user propagation was exploited to find the UAV's optimal position. In order to minimize the transmission power or maximize the capacity, the UAV searches for the opportunity of having a LOS propagation toward the user, while maintaining a good connection with the BS

Some other works consider the situation of busy traffic and assumed the users to be in a crowd and move linearly mimicking the movement of vehicles on the road, for a fixed altitude, the authors in [46] studied the automatic self-organization of an ABS, where ABS update its position according to the users' requirement and mobility, it was proven that a single ABS can replace 10 TBSs deployed along that street. In [47], the goal was to enhance the duration of communication coverage in future mobile networks based on NOMA (non-orthogonal multiple access), the optimal transmission power was studied in terms of the users' clustering for NOMA purposes and the positioning of the ABS over time as the users move. The work in [48] focused on optimizing the placement update interval so the ABS can adapt its position to maximize the number of covered users, an iterative approach that takes into account two metrics which are, the total ABS flight time and user coverage probability were proposed. In [49], the authors used a Q-learning algorithm to find the optimal updated 3D position of the ABS after the users' movement and the QoS drop under the specified threshold; this method was shown to be successful in compensating for QoS loss due to user mobility.

Some other works investigated the 3-D placement of multiple UAVs, and even fewer considered the users to be in movement. for a static set of ground users, the authors in [42] assumed that the UAVs had the same transmit power and altitude, which means all the UAVs have an equal coverage radius, the problem was formulated and modelled as a circle packing problem, where several circles corresponding to the UAVs coverage regions are packed in the coverage area in a way that there's no overlapping between the circles, and the UAVs use a minimum transmit power. In [50], the optimal altitude that leads to maximum coverage and minimum required transmit power was derived. Then, the deployment of two ABSs in both interference and interference-free situations was studied. The results have shown the existence of an optimal separation distance that provides maximum coverage for a given area. In [41], the 3D placement of the UAVs was investigated while considering two scenarios. In the first one, the maximum hover time and the users' spatial distribution were taken into account to find the optimal cell portioning of the area using the optimal transport theory, and associate each cell with a UAV. In the second scenario, given the load requirement of each user, the minimum average hover time needed was determined.

Unlike the above contributions, the authors in [51]studied the deployment of multiple UAVs while considering the users' mobility, their main objective was to find the minimum number of needed UAVs, and their optimal 3D position to provide efficient connectivity with a minimum cost, to solve the positioning problem, an algorithm based on electrostatic forces was proposed. The authors assumed the users to be particles with static negative charges, and the ABSs to be particles with dynamic positive charges, the attractive forces formed between the ABSs and the users and the repulsive forces formed between the ABSs make these lasts move until electrostatic equilibrium is achieved. However, the authors assumed that the ABSs use separate frequency bands and do not interfere with each other.

The ABSs deployment with consideration of interference was shown in [52], where the ABSs are used to support existing terrestrial base stations (TBSs), and they both communicate with the ground users using the same bandwidth to maximize the frequency reuse, so the interference among all the base stations (ABSs and TBSs) is considered. The authors' objective was to maximize the ABSs' deployment efficiency by maximizing the number of satisfied users to the energy consumed by the ABSs to get to their initial positions. First, the sets of users associated with each ABS are determined by clustering the unsatisfied users, and each ABS is placed at the gravity centre of its corresponding cluster. Then, depending on Coulomb's law the optimal position of each ABS is the sum vector of all the forces exerted by the users, other ABSs and the ABS base camp.

In [53], the authors proposed a genetic-algorithm-based approach to determine the number and locations of the ABSs, An SDN architecture was considered, so, the management and placement decisions of the ABSs were made by a centralized node (CN). In order to reduce the complexity of the deployment task, the altitude of each ABS was assumed to be fixed, hence, the position to be selected is only the 2D coordinates of each ABS location. In all the previous works, the deployment problem of ABSs for wireless coverage considers either the deployment of a single ABS and static users or a single ABS and moving crowd. Some works handled the problem of the deployment of multiple ABSs, but the users were stable. However, the few works that did deploy multiple ABSs to cover dynamic users have fixed their altitude which limits the vertical movement of the ABSs.

5.3 System Model

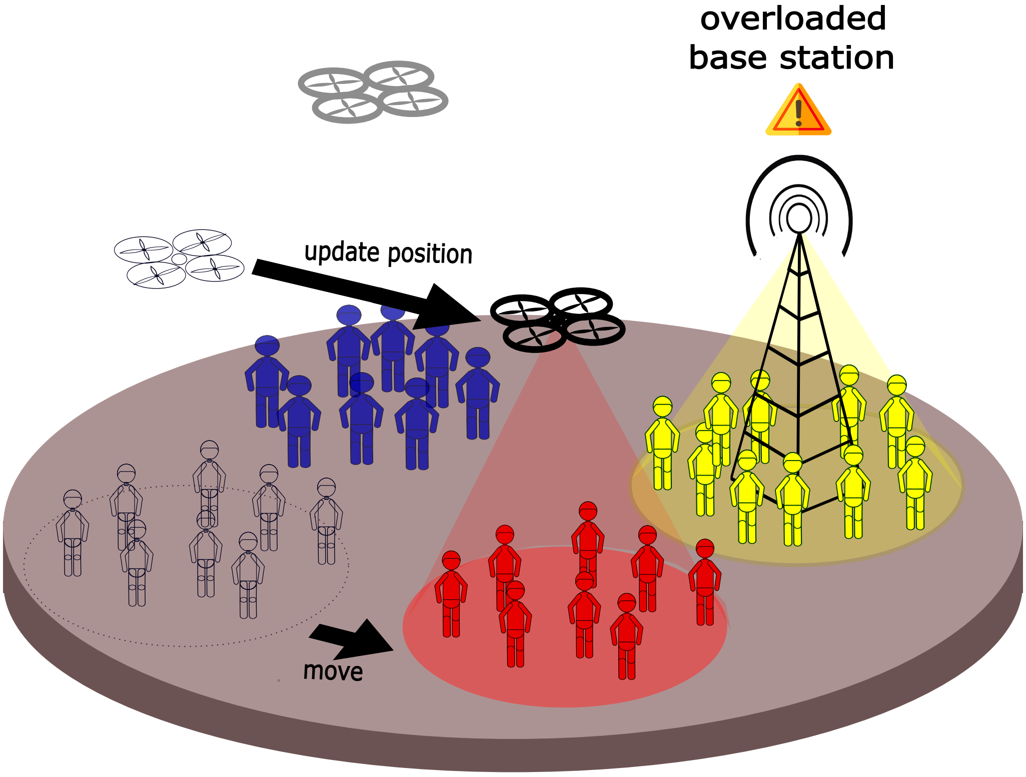

Figure 5.1: System model.

We consider the cellular communication system illustrated in Figure 5.1, where the users' requirements are not fully fulfilled or cannot be handled since the existing terrestrial base stations (TBSs) are damaged, overloaded or unavailable. To generalize our scenario, we consider both cases where a set D of UAV-mounted base stations (where D=d1,..,dk and k is the number of ABSs) denoted as aerial base stations (ABSs) are deployed either to support existing TBSs or to replace them. Let B be the set of all base stations (ABSs and TBSs) denoted as BSs (where B=b1,...,bi and i is the number of BSs), each BS $b \in B$ is located in the 3D space at (x_b, y_b, z_b), where (x_b, y_b) is the horizontal position of the ABS, and z_b represents its altitude, each ABS is equipped with a directional antenna, in addition, we suppose that the TBSs and ABSs operate at different frequency bands.

The set U (where U=u1,...,uN and N is the number of users) groups all the users of the covered area, for a user $u \in U$ to be considered connected, the received signal needs to be above a certain threshold T. Hence, the set U is divided into, a set U^t with the cardinality N^{ut} that represents the users successfully connected, and the set U^a with the cardinality N^{ua} that represents the non-connected users or the ones with a poor quality of service (QoS), Each user $u \in U$ is located at position (x_u, y_u, z_u). Assuming that the users are moving following a random walk mobility model, our objective is to find in real time the optimal placement of the ABSs in the 3D space in order to provide efficient connectivity to the ground users. Knowing that every ABS $d \in D$ has a maximum available bandwidth

B_d and limited resources, each ABS has a constraint C_A that represents the maximum number of users connected to that ABS.

5.3.1 Signal Propagation Model

In order to determine the signal power P_r that a user receives from an ABS, we adopt the air-to-ground propagation channel model given by [54], the approach is basically based on considering the links of line-of-sight (LoS) and non-line-of-sight (NLoS) between the ABS and ground user separately, each link has a probability of occurrence, and it depends on the elevation angle between the ABS and the user, the environment (suburban, urban, dense, etc), and the distance in between. The probability of having a LOS connection is given by [44]:

$$\frac{1}{1 + a * \exp(-b(\frac{180}{\pi} * \tan^{-1}(\frac{h}{d}) - a))} \tag{5.1}$$

where a and b are environment related constants and are given by [54], d denotes the euclidean distance between the ABS and the user, and h represents the ABS's altitude, moreover, the probability of having a NLoS is equal to:

$$P(NLoS) = 1 - P(LoS) \tag{5.2}$$

According to [54], the path loss model in decibels (dB) for LoS and NLoS links are respectively:

$$LoS = 20 * \log(\frac{4\pi f_c d_i}{c}) + \eta LoS \tag{5.3}$$

$$NLoS = 20 * \log(\frac{4\pi f_c d_i}{c}) + \eta NLoS \tag{5.4}$$

where f_c is the carrier frequency, d_i is the distance between the ABS and the user i, given by $d_i = \sqrt{h^2 + r^2}$. Furthermore, ηLoS and $\eta NLoS$ are the average additional losses for LoS and NLoS, respectively, they depend on the environment and they are given in [54]. The probabilistic mean path loss is given by:

$$L(h, d) = LoS \times P(LoS) + NLoS \times P(NLoS) \tag{5.5}$$

Furthermore, considering an ABS transmitting its signal with the power of P_t, the received power by the user is determined as:

$$Pr = Pt - L(h, d) \tag{5.6}$$

The data rate of the user $u \in U^a$ is calculated as:

$$r_u = b_{du} * \log_2(1 + \frac{P_r}{\sigma^2}) \tag{5.7}$$

where b_{du} is the communication bandwidth, and σ^2 represents the thermal noise.

5.3.2 Energy Consumption Model for Rotary-Wing UAV

The UAV energy consumption is generally related to two factors, the first one is communication related energy, which means the energy needed for circuitry, signal processing and signal transmission and reception. However, this energy is usually neglected as it represents only 16% of the total energy of the UAV [44]. The second factor and the most important one is the propulsion energy needed to keep the UAV in the air and support its mobility.

Generally, the propulsion energy depends on the speed of the UAV, as shown in Figure 5.2, the plot of the power consumption along with its three components. So, as derived in [55], for a rotary-wing UAV flying with speed V, the propulsion power consumption is modelled as follows:

$$P(V) = P_0(1+\frac{3\,V^2}{U_{tip}^2}) + P_i(\sqrt{1+\frac{V^4}{4\,v_0^2}} - \frac{V^2}{2\,v_0^2}) + \frac{1}{2}\,d_0\,\rho\,s\,A\,V^3 \tag{5.8}$$

where P_0 and P_i represent the blade profile power and induced power in hovering status, respectively, U_{tip} denotes the tip speed of the rotor blade, v_0 is known as the mean rotor induced velocity in hover, d_0 and s are the fuselage drag ratio and rotor solidity, respectively, ρ and A denote the air density and rotor disc area.

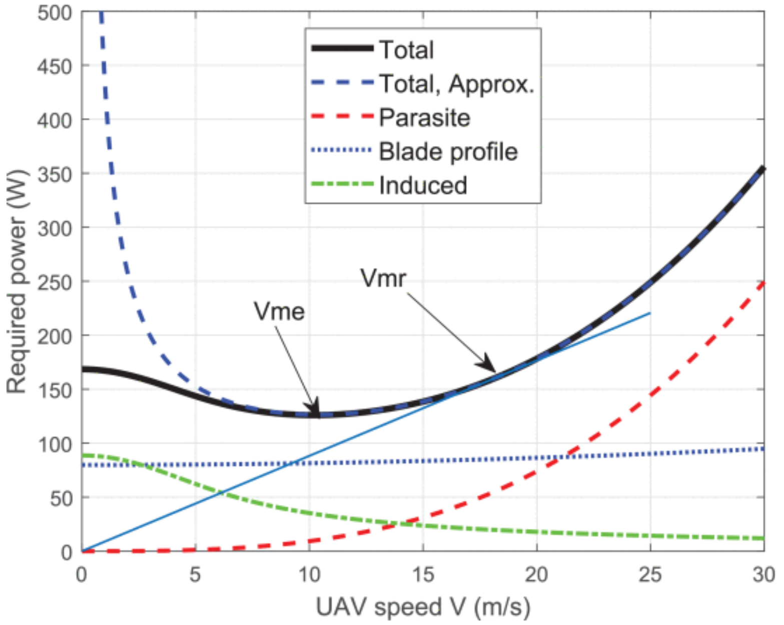

Figure 5.2: Propulsion power versus speed of rotary-wing UAV. Adapted from [55].

In order to estimate the coverage duration and the endurance of the UAVs, we assume that a UAV is provided with an engine that draws its energy from a battery with a capacity of B_c and a voltage V.

At each time t, the UAV consumes the power $P(t)$ from hovering with a certain speed, and it's calculated using Formula 5.8. So, the amount of energy consumed is calculated as follows:

$$E(t) = P(t)/V \tag{5.9}$$

Knowing that the battery's capacity decrease as the UAV keeps on hovering, hence, at each time t the amount of energy left in the battery is :

$$B_c(t) = B_c(t-1) - E(t) \tag{5.10}$$

5.4 Problem Formulation

Our objective is to determine the optimal position of the ABSs to provide efficient wireless coverage and to maximize the number of connected ground users with respect of the data rate r_u received by each individual $u \in U_a$. In this section, we formulate the problem as a mixed integer programming optimization problem.

We introduce l_d and c_{du} as decision variables, where:

$$l_d = \begin{Bmatrix} 1, & if\, ABS\, d\, is\, deployed \\ 0, & otherwise \end{Bmatrix} \tag{5.11}$$

And

$$c_{ud} = \begin{Bmatrix} 1, & if\, user\, u\, is\, connected\, to\, ABS\, d \\ 0, & otherwise \end{Bmatrix} \tag{5.12}$$

Assuming that the user u is connected to the ABS d, if the data rate experienced by the user satisfies its data rate is above the threshold $T(r_u \leq T)$.

The problem is formulated as follows:

$$Max\ N^{ut} \tag{5.13}$$

s.t

$$\sum_{d \in D} c_{du} \leq 1, \quad \forall u \in U \tag{5.14}$$

$$r_u \geq T, \quad \forall u \in U^t \tag{5.15}$$

$$\sum_{u \in U} c_{du} \leq C_A, \quad \forall d \in D \tag{5.16}$$

$$\sqrt{(x_{d_1} - x_{d_2})^2 + (y_{d_1} - y_{d_2})^2 + (z_{d_1} - z_{d_2})^2} > Dist, \quad \forall d_1, d_2 \in D \tag{5.17}$$

$$z_d \geq h_{min}, \quad \forall d \in D \tag{5.18}$$

The objective function 5.13 maximizes the number of connected ground users. The constraint 5.14 ensures that a user is served by only one ABS at a time, 5.15 guarantees that the users' data rates are acceptable if they are above a certain threshold T, 5.16 makes sure that the capacity of an ABS regarding the number of connected users is respected, constraint 5.17 enforce a safety distance between the set of all ABS, and 5.18 limit the altitude of the ABS to stay above the height h_{min}.

5.5 An Electrostatic Neurovectors Based 3D Deployment of Aerial Base Station for Small Cellular Networks

The problem formulated in the previous section is of high complexity, because of the non-linearity of some equations, and the difference in the decision variables type, the problem is an MINLP, which is an NP-hard problem [56]. In this section, we aim to come up with a solution to find the ABSs' optimal 3-D position to provide wireless coverage to a set of ground users. To this end, we propose an algorithm based on the notion of electrostatic forces, that uses neurovectors as a computational basis to determine the optimal position of the ABSs.

First, only an initial minimum number of ABSs is deployed, this number can increase after if it's necessary. Then, the set of all ABSs and users are considered to be electrically charged particles, and by using Coulomb's law the attraction and repulsion forces between them are calculated and exploited to move the ABSs until the electrical equilibrium is achieved [51] [52]. The force between two particles is defined as:

$$F_{ab} = \frac{q_a q_b}{||\vec{ab}||^2}\widehat{ab} \tag{5.19}$$

where $||\vec{ab}||$ is the magnitude of the vector $\vec{ab}$, and $\widehat{ab}$ its direction.

In order to determine the new positions of the ABSs that make them provide the intended quality of service, we calculate the shift vector $d(\vec{t})$ for each ABS d E D. the vector $d(\vec{t})$ has the same direction as the force vector obtained by summing all the forces exerted on the ABS d.

The first force exerted on the ABS is the attraction force exerted by the users to attract the ABS and get delivered the required services. knowing that the power transmitted by the ABS and received by the user has an inverse relationship with the distance between the ABS and the user, meaning that the farther the user is from the ABS, the weaker the signal. Hence, the attraction force exerted by one user will increase as the distance increases. In addition, the force's magnitude is also related to the number of non-connected users, so, if the number expands as the users distribute, the attraction force decreases and the ABS goes up to expand its coverage zone. The force can be formulated as follows :

$$\vec{F_{du}} = \alpha\widehat{du} \tag{5.20}$$

where $\alpha == \frac{||\vec{du}||-1}{||\vec{du}||\times N^{ua}}$. Hence, the attraction force exerted on ABS d by all the non connected users is given by :

$$F\vec{U}_d = \sum_{u \in U^a} \alpha\widehat{du} \tag{5.21}$$

The second force applied on the ABS is considered to satisfy constraint 5.17, which is the repulsion force exerted by other ABSs, these forces make sure that a safety distance between all the ABSs is always respected. The force applied on one ABS by another one has a proportional relation with the distance between the, therefore, as the distance between the two ABS d_1 and d_2 decreases the repulsion increase to keep them from colliding. For a more efficient distribution of the ABSs in the space, the repulsion force will also be related to the number of non-connected users, so it can be formulated as follows:

$$\vec{F_{d_1d_2}} = \beta \widehat{d_1 d_2} \tag{5.22}$$

where $\beta = \frac{N^{ua}}{\|\vec{d_1 d_2}\|}$. As we consider the repulsion forces exerted by all the ABSs, the force is written as:

$$F\vec{D}_d = \sum_{d_1 \in D} \beta \widehat{dd_1} \tag{5.23}$$

In order to fulfil constraint 5.18, we apply another force on the ABSs which is the repulsion force of the ground to make sure that the ABSs are always above a certain altitude limit, The force exerted on ABS d is expressed as:

$$F\vec{G}_d = \gamma \vec{r} \tag{5.24}$$

where $\gamma = \frac{N}{z_b^2}$, and $\vec{r}$ is the direction vector of the force which is orthogonal to the ground.

Taking into account all the forces exerted on an ABS, this last calculate the total force which is the sum of all the applied forces, when the QoS experienced by the users drops, the ABSs need to update their position to adapt to requirements, so each ABS calculate the sum vector $\vec{F}$ of all the exerted forces that we mentioned above, and move in that direction towards the new position. Figure 5.3 is an illustration of the drone and the forces exerted on it, the total force $\vec{F}$ applied on ABS d is calculated as:

$$\vec{F} = F\vec{U}_d - F\vec{D}_d + F\vec{G}_d \tag{5.25}$$

As shown in Figure 5.2 the power consumed by the UAV while flying increases as the speed of the UAV increases, so in order to minimize the consumed power, extend the battery's lifetime and provide wireless coverage for a longer time, we set a threshold to control the UAV's velocity. In that matter, the vector $\vec{F}$ would not be directly applied to the UAVs' position, instead the movement vector $d(\vec{t})$ is extracted.

$$d(\vec{t}) = \lambda \vec{F} \tag{5.26}$$

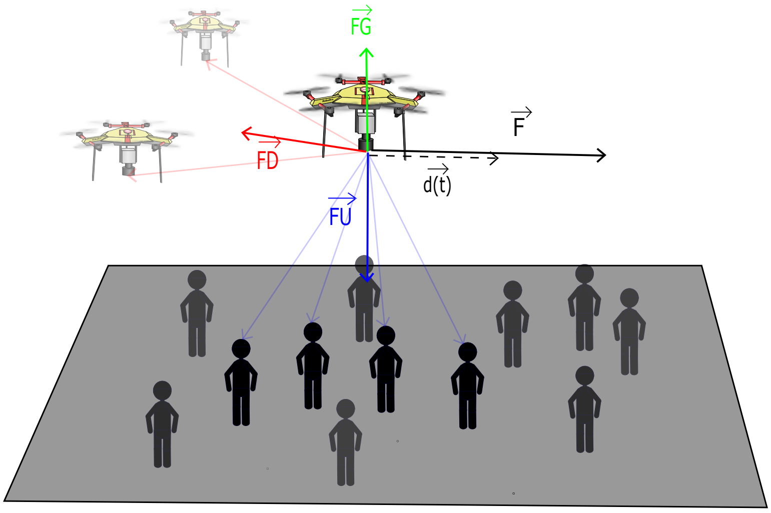

Figure 5.3: Vector model for electrostatic ABS wireless 3D coverage

Furthermore, to give our ABSs more flexibility and the ability to move in a more dense urban environments with high buildings that they have to make their way around them in order to get to the users, we add an additional force $F\vec{O}_d$, which is a repulsive force exerted on the ABS by the obstacle to keep it away, so it only depends on the distance between the ABS and the obstacle so the smaller the distance, the stronger the force. given an obstacle o, the force is formulated as follows :

$$F\vec{O}_d = \delta\widehat{do} \tag{5.27}$$

where $\delta = \frac{1}{\left\|\vec{du}\right\|^2}$. Hence, the total force $\vec{F}$ applied on ABS d is calculated as :

$$\vec{F} = F\vec{U}_d - F\vec{D}_d + F\vec{G}_d - F\vec{O}_d \tag{5.28}$$

5.6 Neuronal Modeling

For the UAVs to position themselves to provide wireless coverage for the ground users and more essential to avoid any potential collision between them, each UAV needs to estimate two groups of vectors. In the first group, regarding the users' positions, for the UAV to determine the force vector $F\vec{U}_d$ that enables it to cover a certain number of users, it needs to estimate the reference vectors of the users' positions u_i . Each vector is encoded by a neurovector. Then, the vector $F\vec{U}_d$ is calculated according to Equation 5.21.

The second class of vectors, associated with the UAVs' positions, at each movement the UAVs should always respect a safe distance in between to avoid any possible collisions, also, to let the UAVs be distributed in a way to provide a more efficient coverage to a maximum number of users. Each UAV needs to estimate the vectors of the other UAVs' positions relative to its own position, each force is calculated individually according to Equation 5.22, and the vector $F\vec{D}_d$ is calculated by summing the neurovectors encoding the UAVs positions d_i . However, as the force is repulsive the inverse of the vector $\breve{}\vec{FD}_d$ is calculated.

After the neurovector representing $\vec{FG}_d$ is determined, the neurovector F is calculated according to Equation 5.25. Using the graphical representation proposed in chapter 2, we presented in Figure 5.4 the neuron model that allows a group of UAVs to provide cellular wireless coverage.

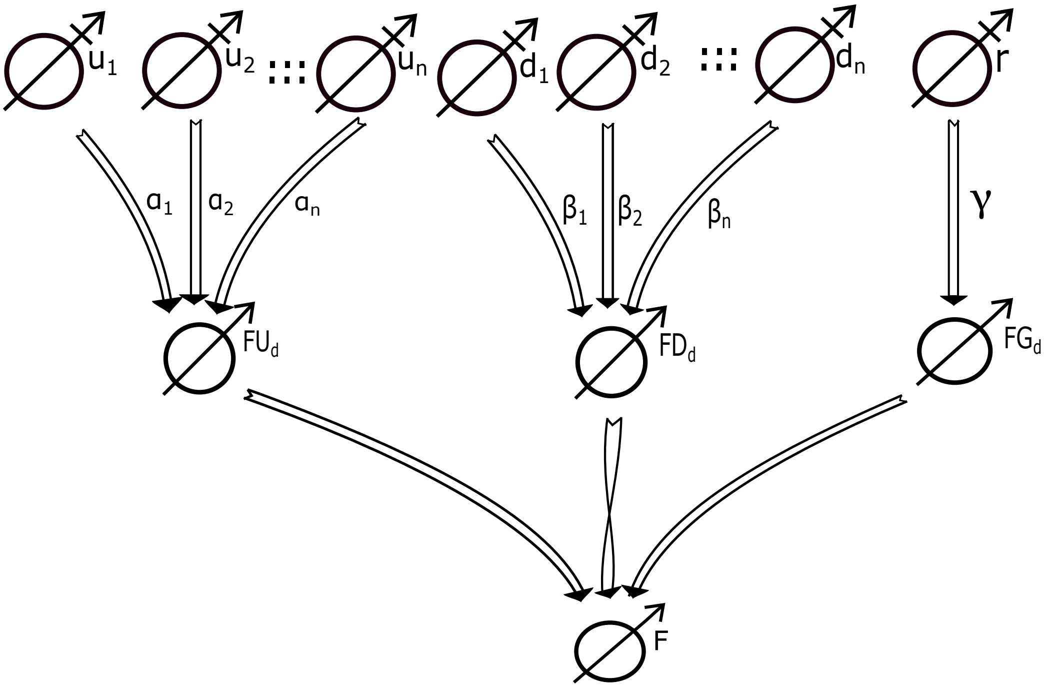

Figure 5.4: Neuron model for wireless coverage.

5.7 Simulations and Results

In this section, we present a simulation scenario and numerical results to evaluate the performance of our proposed solution.

5.7.1 Simulation Scenario

We consider a simulation area of 600 x 600 m, the users are homogeneously and randomly distributed in the area and move according to a random walk model where every user moves at constant speed v picked randomly from [1.25 m/s, 1.5 m/s] [57], according to a heading defined by the angle θ, the equation of the movement is defined as follows:

$$\begin{aligned}\Delta x &= v * \cos(\theta)\Delta t\\ \Delta y &= v * \sin(\theta)\Delta t\end{aligned} \tag{5.29}$$

The users initially move according to θ that changes slightly with each step conforming to the formula $\Delta\theta = 0.3 - 0.6\beta$, where β is a random number picked between 0 and 1, This alteration allows the users to distribute in the whole area. After a fixed amount of time n, the value of θ is replaced with a new one, picked randomly between 0 and 2π [30].

Table 5.1: System parameters for the signal propagation model

Parameter	Notation	Value
ABS maximum capacity	C_A	30
ABS transmit power	P_t	20 dBm
Carrier frequency	f_c	2 GHz
Bandwidth	B_d	100 MHz
thermal noise power	σ	-80 dB
SNR threshold	T	2dB
a from (1)	-	9.61
b from (1)	-	0.16
Additional pathloss under LoS	ηLoS	1
Additional pathloss under NLoS	$\eta NLoS$	20

Table 5.2: System parameters for the energy consumption model

Parameter	Notation	Value
Blade profile power	P_0	79.86 W
Induced power	P_i	88.62 W
Tip speed of the rotor blade	U_{tip}	120 m/s
Mean rotor induced velocity	v_0	4.03 m/s
Fuselage drag ratio	d_0	0.6
Air density	ρ	1.225 kg/m3
Rotor solidity	s	0.05
Rotor disc area	A	0.503 m2
Battery capacity	B_c	5870 mAh
voltage	V	15.2 V
UAV's speed	-	10.21 m/s

For the signal propagation model, we used parameters used for the suburban environment summarized in Table 5.1 [39]. For the energy consumption model, the parameters are summarized in Table 5.2 [59]. For the neurovectors we used a population size of (10×10).

First, to evaluate the efficiency of our solution, we deploy the ABSs according to the "force" algorithm from [51] which is also based on electrostatic forces next to our neurovectors-based solution, in order to compare the performances of the two solutions.

To further evaluate the performance of the solution in more dense urban areas where there are buildings for the ABSs to overcome, we add a few obstacles in the simulation area that the users can go around, and see the behaviour of the ABSs.

5.7.2 Numerical Results

In this section we evaluate the performance of our proposed solution in terms of user satisfaction, energy consumption and ABSs deployment efficiency.

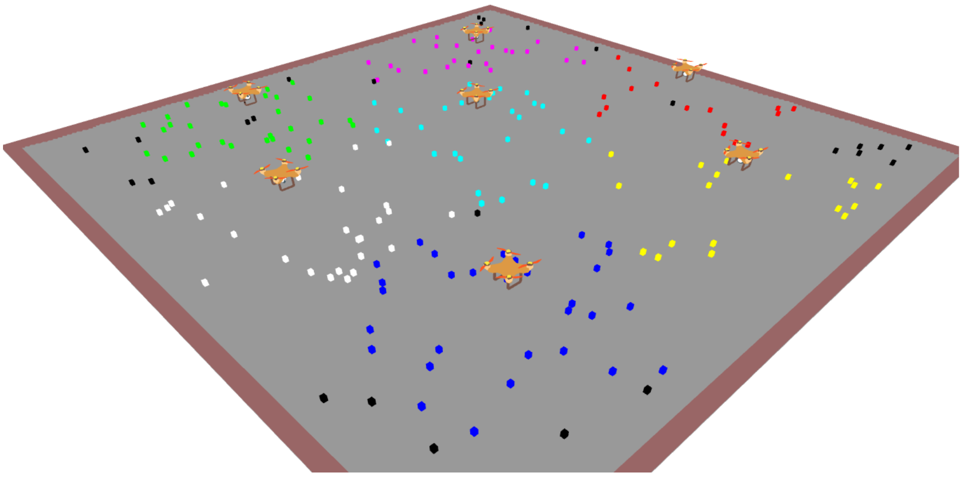

Figure 5.5: simulation snapshot showing the distribution of the UAVs.

In Figure 5.5, we can see the distribution of the UAVs in the simulation area to cover the users. The colored dots represent the connected users, which means the users who have the same color are connected to the same UAV, and the black dots represent the non-connected users or the one with poor quality of service.

To show the adaptation of the ABSs with the users' movement, we track the data rate, the percentage of connected users and standard deviation to evaluate the difference between the users' received data rate. We deploy a fixed number of ABSs $k = 7$ to provide wireless coverage for a given number of users $N = 200$. At first the users are grouped in the middle of the simulation area, and the ABSs are deployed to provide coverage. Then as the users start to distribute in the area the ABSs move according to them to maintain connectivity.

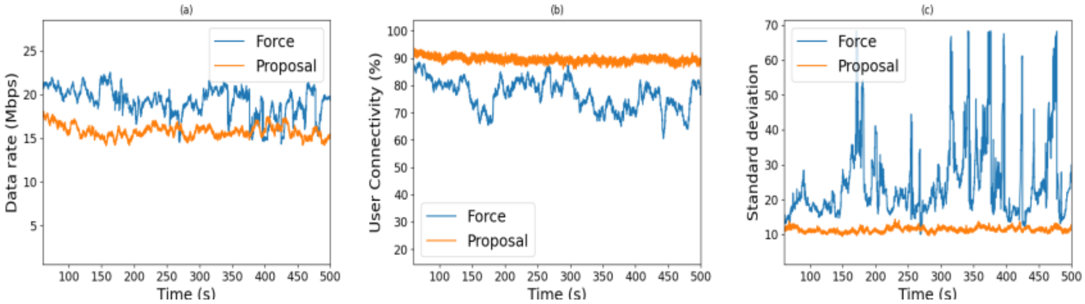

Figure 5.6: Numerical results from simulations of our solution and the "force" algorithm :Data rate (a), User connectivity (b), Standard deviation of users' data rate(c).

Figure 5.6 is a representation of the results obtained from monitoring the data rate, users connectivity and standard deviation for a period of time as the users move and distribute throughout the whole simulation area. In Figure 5.6.a, we show the average data rate received by the users, the data rate achieved by our proposed solution is slightly underneath the one achieved by the force algorithm, however, our solution did maintain almost the same data rate all the time compared to the force algorithm where it kept bouncing up and down.

Figure 5.6.b represents the percentage of connected users, we can see that our solution sustained the same gain of around 90% constantly, which means that more than 180 users out of 200 were connected at all times, as for the other 20, they represents the isolated users who get out from the ABS coverage zone, so, if a user leaves an ABS's coverage area and enter to another one's coverage zone, he will automatically be connected to that new zone, but if its new destination is not covered by any ABS, the user will be disconnected, as the attraction force of a number of users grouped in one spot would be stronger against one or two isolated users. In contrast to the force algorithm, the user connectivity percentage was very unstable as the users moved, and decreased to reach 60%, which means 80 users received a non-acceptable data rate which is quite high, that because our model allows the ABSs to have more freedom in the 3D space, so it can adjust its altitude to connect more users as they distribute.

In Figure 5.6.c, we display the standard deviation that indicates by how much the data rate received by the users differs from the average data rate, for our solution the standard deviation is steady and equal to 10, which means that the users have a stable, close to average data rate the whole simulation, on the contrary, the force algorithm marked a noticeable gap in the data rate experienced by the users that go up to 70.

From these results, we can deduce that our neurovectors-based solution delivers quite good results compared to the force algorithm and it's much more robust and can adapt to the users' mobility while keeping a stable data rate experience, and that because our ABSs can move freely in the 3D space and adjust their altitude according to the users' distribution.

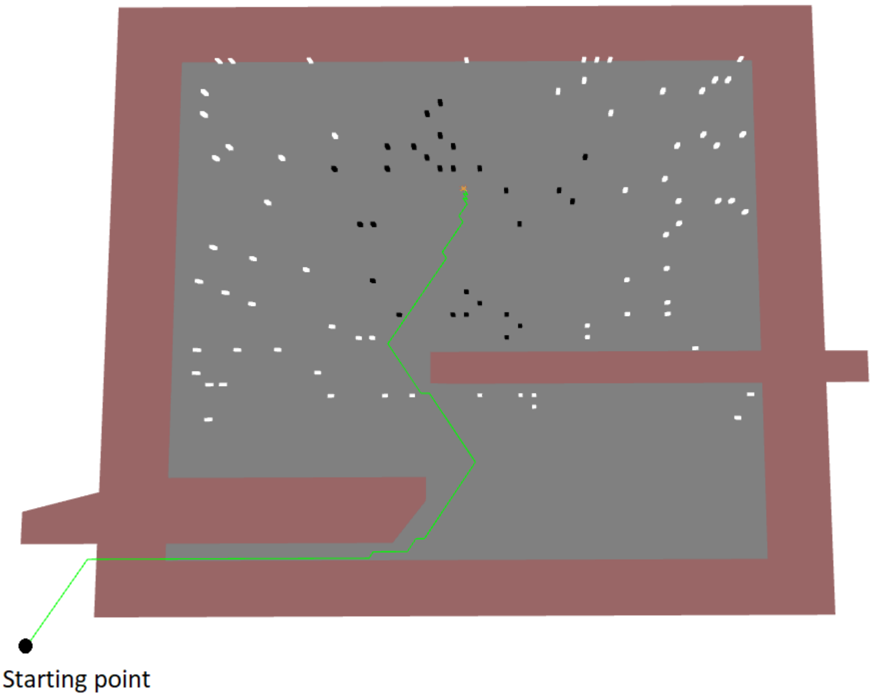

Figure 5.7: Simulation snapshot showing the path of an ABS moving through obstacles.

For the next experience, we set a few obstacles in the simulation area, on the way of the ABS to the users. These obstacles could be buildings in real life, so, their height is quite high, and the ABS won't be able to make its way above the building. Figure 5.7, shows the path of the ABS going from the starting point till getting to the users to provide wireless coverage, the ABS did navigate its way through the obstacles without collapsing, and with leaving a safety distance in between.

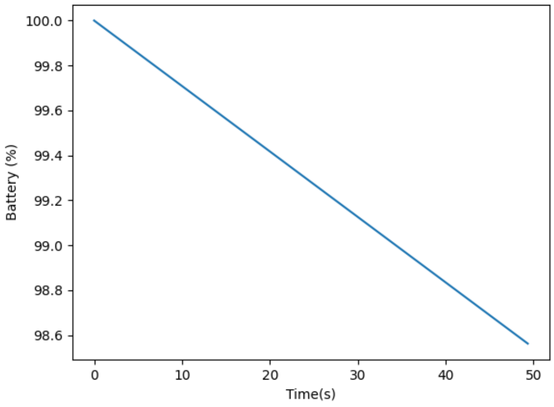

Figure 5.8: Average battery life of one UAV through time.

Figure 5.8, represents the average battery life through time of one UAV while hovering and providing wireless coverage, the graph is a straight line which means at each time t the same amount of energy is consumed, and that because as the UAVs' velocity is constant and set to 10.21 m/s, which is the speed that leads to minimum power consumption according to Figure 5.2, hence, battery endurance is maximized.

5.8 Conclusion

In this chapter we have studied the deployment of a fleet of UAVs mounted base station in the 3D space to provide wireless coverage to a set of ground users in an area where there is no service or where the existing infrastructure is damaged or overloaded, we formulated the problem and proposed a solution based on electrostatic forces where each UAV moves according to a force which is the sum of an attraction force exerted by the users to get connected, a repulsion force from the other UAVs to avoid collision, and the repulsion force exerted by the ground to keep it up in the air, in addition to another repulsion force comes from obstacles if exist. This solution is structured into a vector model and then represented using neurovectors.

The simulation results which are based on various system parameters were compared to an existing work from the literature, they showed close results analyzing the data rate and a superior performance regarding its ability to adapt to the users' movements. We also tested and confirmed its ability to overcome obstacles and adjust with the environment.

CONCLUSION AND PERSPECTIVES

Conclusion

Our objective in this thesis was to present a bio-inspired computing model using artificial neurons for drones to use and navigate in the 3D space. In this context, we were interested in proposing several neuron models to let drones navigate in different flying scenarios. The first one focuses on the bat's stealthy hunting style, which enables the animal to approach its prey without being noticed. To replicate the movement camouflage tactic, we have offered two models: the first with regard to a fixed point in the environment and the second with respect to a point at infinity. To do this, we used a neural representation for three-dimensional vectors using the discretization of the sine-matrix, a periodic function (MS). In the second scenario, we constructed two neuron models that allow a UAV to track a target, the first focused on tracking a ground target while considering the height limit that should be respected. The other one assumes that the target can fly and move in 3D space, and the UAV should track the target while keeping a constant distance. These models were tested and the results showed that the UAV was successful in tracking the target in both cases. In the last scenario, a neuron model that allows a fleet of UAVs to provide wireless coverage for a set of ground users while considering their data requirements and their mobility, the model was tested and compared to existing work from the literature, and the result has showed better results considering the stability of the provided data rate. Thanks to neurovectors, we were able to create neural architectures that isomorphically represent the vector equations of motion enabling concealment during hunting. These structures accurately mimic camouflage even when there are a low number of neurons (16 neurons per vector). This novel method of vector representation may pave the way for neural modeling of additional problems arising from vector differential equations. The models we provide, however, suffer from an idealization of the signals conveyed between the populations of neurons and an absence of realism in the simulation of the neurons involved (simulations without noise).

Perspectives

This work could pave the way for neural modeling of other problems emerging from vector differential equations, it will allow the solving of many problems using a biologically plausible approach, and it will also reduce, using real-time neuromorphic computing, the computational time of tasks that require complicated calculations.

BIBLIOGRAPHY

[1] John McCarthy. What is artificial intelligence. *URL: http://www-formal. stanford. edu/jmc/whatisai. html*, 2004.

[2] ING DANIEL SURGENT. Spiking neural networks, perspective. *URL: https://invenio.nusl.cz/record/41755/files/content.csg. pdf*, 2010.

[3] Samanwoy Ghosh-Dastidar and Hojjat Adeli. Third generation neural networks: Spiking neural networks. In Wen Yu and Edgar N. Sanchez, editors, *Advances in Computational Intelligence*, pages 167–178, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[4] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

[5] Kenji Suzuki. *Artificial neural networks: Architectures and applications.* BoD–Books on Demand, 2013.

[6] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, 1952.

[7] Andrzej Kasiński and Filip Ponulak. Comparison of supervised learning methods for spike time coding in spiking neural networks. 2006.

[8] David Heeger. Integrate and fire model of spike generation. *J. Neurosci*, pages 1–4, 1997.

[9] Petr Lansky and Susanne Ditlevsen. A review of the methods for signal estimation in stochastic diffusion leaky integrate-and-fire neuronal models. *Biological cybernetics*, 99(4):253–262, 2008.

[10] Eugene M Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

[11] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition.* Cambridge University Press, 2014.

[12] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory.* Psychology Press, 2005.

[13] Tim VP Bliss, Graham L Collingridge, and Richard GM Morris. Introduction. long-term potentiation and structure of the issue. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 358(1432):607, 2003.

[14] Stuart Zola-Morgan and Larry R Squire. Memory impairment in monkeys following lesions limited to the hippocampus. *Behavioral neuroscience*, 100(2):155, 1986.

[15] Edvard Moser, May-Britt Moser, and Per Andersen. Spatial learning impairment parallels the magnitude of dorsal hippocampal lesions, but is hardly present following ventral lesions. *Journal of neuroscience*, 13(9):3916–3925, 1993.

[16] Samuel Frazer Cooke and Timothy VP Bliss. Plasticity in the human central nervous system. *Brain*, 129(7):1659–1673, 2006.

[17] Tim VP Bliss and Graham L Collingridge. A synaptic model of memory: long-term potentiation in the hippocampus. *Nature*, 361(6407):31–39, 1993.

[18] Peter V Massey and Zafar I Bashir. Long-term depression: multiple forms and implications for brain function. *Trends in neurosciences*, 30(4):176–184, 2007.

[19] Russell E Nicholls, Juan Marcos Alarcon, Gaël Malleret, Reed C Carroll, Michael Grody, Svetlana Vronskaya, and Eric R Kandel. Transgenic mice lacking nmdar-dependent ltd exhibit deficits in behavioral flexibility. *Neuron*, 58(1):104–117, 2008.

[20] Gaël Malleret, Juan M Alarcon, Guillaume Martel, Shuichi Takizawa, Svetlana Vronskaya, Deqi Yin, Irene Z Chen, Eric R Kandel, and Gleb P Shumyatsky. Bidirectional regulation of hippocampal long-term synaptic plasticity and its influence on opposing forms of memory. *Journal of Neuroscience*, 30(10):3813–3825, 2010.

[21] Mark Bear, Barry Connors, and Michael A Paradiso. *Neuroscience: Exploring the Brain, Enhanced Edition: Exploring the Brain.* Jones & Bartlett Learning, 2020.

[22] Hideaki Ogasawara, Tomokazu Doi, and Mitsuo Kawato. Systems biology perspectives on cerebellar long-term depression. *Neurosignals*, 16(4):300–317, 2008.

[23] Henry Markram, Joachim Lübke, Michael Frotscher, and Bert Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297):213–215, 1997.

[24] Bin Chen and Kenneth Y Kwan. *Neural circuit and cognitive development: Comprehensive developmental Neuroscience.* Academic Press, 2020.

[25] Daniel R Kunkle and Chadd Merrigan. Pulsed neural networks and their application. *Computer Science Dept., College of Computing and Information Sciences, Rochester Institute of Technology*, 2002.

[26] Wenzhe Guo, Mohammed E Fouda, Ahmed M Eltawil, and Khaled Nabil Salama. Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems. *Frontiers in Neuroscience*, 15:638474, 2021.

[27] Edgar D Adrian and Yngve Zotterman. The impulses produced by sensory nerve-endings: Part ii. the response of a single end-organ. *The Journal of physiology*, 61(2):151, 1926.

[28] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity.* Cambridge university press, 2002.

[29] Wolfgang Maass and Christopher M Bishop. *Pulsed neural networks.* MIT press, 2001.

[30] Mohamed Zennir. *Nouvelles approches en robotique pour la navigation autonome dans les environnements partiellement connus.* PhD thesis, 12 2017.

[31] Mandyam V Srinivasan and Matthew Davey. Strategies for active camouflage of motion. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 259(1354):19–25, 1995.

[32] Akiko Mizutani, Javaan S Chahl, and Mandyam V Srinivasan. Motion camouflage in dragonflies. *Nature*, 423(6940):604–604, 2003.

[33] Michael Hopkin. Dragonfly flight tricks the eye. *Nature*, Jun 2003.

[34] Kaushik Ghose, Timothy K Horiuchi, PS Krishnaprasad, and Cynthia F Moss. Echolocating bats use a nearly time-optimal strategy to intercept prey. *PLoS biology*, 4(5):e108, 2006.

[35] Alessandro Rucco, A. Pedro Aguiar, Fernando Lobo Pereira, and João Borges de Sousa. A predictive path-following approach for fixed-wing unmanned aerial vehicles in presence of wind disturbances. In Luís Paulo Reis, António Paulo Moreira, Pedro U. Lima, Luis Montano, and Victor Muñoz-Martinez, editors, *Robot 2015: Second Iberian Robotics Conference*, pages 623–634, Cham, 2016. Springer International Publishing.

[36] Tiago Oliveira and Pedro Encarnação. Ground target tracking for unmanned aerial vehicles. 08 2010.

[37] Malavathu Srinivasan, Shaowu Zhang, M Lehrer, and Thomas Collett. Honeybee navigation en route to the goal: Visual flight control and odometry. *The Journal of experimental biology*, 199:237–44, 02 1996.

[38] Oleksandr Andryeyev and Andreas Mitschele-Thiel. Increasing the cellular network capacity using self-organized aerial base stations. In *Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, DroNet '17, page 37–42, New York, NY, USA, 2017. Association for Computing Machinery.

[39] R. Irem Bor-Yaliniz, Amr El-Keyi, and Halim Yanikomeroglu. Efficient 3-d placement of an aerial base station in next generation cellular networks. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–5, 2016.

[40] Georgios Amponis, Thomas Lagkas, Maria Zevgara, Georgios Katsikas, Thanos Xirofotos, Ioannis Moscholios, and Panagiotis Sarigiannidis. Drones in b5g/6g networks as flying base stations. *Drones*, 6(2), 2022.

[41] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Wireless communication using unmanned aerial vehicles (uavs): Optimal transport theory for hover time optimization. *IEEE Transactions on Wireless Communications*, 16(12):8052–8066, 2017.

[42] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communications Letters*, 20(8):1647–1650, 2016.

[43] Akram Al-Hourani, Sithamparanathan Kandeepan, and Simon Lardner. Optimal lap altitude for maximum coverage. *IEEE Wireless Communications Letters*, 3(6):569–572, 2014.

[44] Mohamed Alzenad, Amr El-Keyi, Faraj Lagum, and Halim Yanikomeroglu. 3-d placement of an unmanned aerial vehicle base station (uav-bs) for energy-efficient maximal coverage. *IEEE Wireless Communications Letters*, 6(4):434–437, 2017.

[45] Junting Chen and David Gesbert. Optimal positioning of flying relays for wireless networks: A los map approach. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, 2017.

[46] Zdenek Becvar, Michal Vondra, Pavel Mach, Jan Plachy, and David Gesbert. Performance of mobile networks with uavs: Can flying base stations substitute ultra-dense small cells? In *European Wireless 2017; 23th European Wireless Conference*, pages 1–7, 2017.

[47] Mohammadsaleh Nikooroo and Zdenek Becvar. Optimal positioning of flying base stations and transmission power allocation in noma networks. *IEEE Transactions on Wireless Communications*, 21(2):1319–1334, 2022.

[48] Mansi Peer, Vivek Ashok Bohara, Anand Srivastava, and Gourab Ghatak. User mobility-aware time stamp for uav-bs placement. In *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–6, 2021.

[49] Rozhina Ghanavi, Elham Kalantari, Maryam Sabbaghian, Halim Yanikomeroglu, and Abbas Yongacoglu. Efficient 3d aerial base station placement considering users mobility by reinforcement learning. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2018.

[50] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Merouane Debbah. Drone small cells in the clouds: Design, deployment and performance analysis. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2015.

[51] Rania Islambouli and Sanaa Sharafeddine. Autonomous 3d deployment of aerial base stations in wireless networks with user mobility. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2019.

[52] Jan Plachy and Zdenek Becvar. Energy efficient positioning of flying base stations via coulomb's law. In *2020 IEEE Globecom Workshops (GC Wkshps*, pages 1–6, 2020.

[53] Zhongliang Zhao, Pedro Cumino, Christian Esposito, Meng Xiao, Denis Rosário, Torsten Braun, Eduardo Cerqueira, and Susana Sargento. Smart unmanned aerial vehicles as base stations placement to improve the mobile network operations. *Computer Communications*, 181:45–57, 2022.

[54] Akram Al-Hourani, Sithamparanathan Kandeepan, and Abbas Jamalipour. Modeling air-to-ground path loss for low altitude platforms in urban environments. In *2014 IEEE Global Communications Conference*, pages 2898–2904, 2014.

[55] Yong Zeng, Jie Xu, and Rui Zhang. Energy minimization for wireless communication with rotary-wing uav. *IEEE Transactions on Wireless Communications*, 18(4):2329–2345, 2019.

[56] Kai Zhou, Mustafa R Kılınç, Xi Chen, and Nikolaos V Sahinidis. An efficient strategy for the activation of mip relaxations in a multicore global minlp solver. *Journal of Global Optimization*, 70(3):497–516, 2018.

[57] Muhammad Moazzam Ishaque and Robert B. Noland. Behavioural issues in pedestrian speed choice and street crossing behaviour: A review. *Transport Reviews*, 28(1):61–85, 2008.

[58] Cheng Lyu, LF Abbott, and Gaby Maimon. Building an allocentric travelling direction signal via vector computation. *Nature*, 601(7891):92–97, 2022.

[59] Yong Zeng, Jie Xu, and Rui Zhang. Energy minimization for wireless communication with rotary-wing uav. *IEEE Transactions on Wireless Communications*, 18(4):2329–2345, 2019.