TÉCNICO LISBOA

Aerial Sensor Networks for Early Wildfire Detection

André Miguel Martins Rocha

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor(s): Doctor Pedro Daniel Graça Casau
Doctor Rita Maria Mendes de Almeida Correia da Cunha

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Doctor Pedro Daniel Graça Casau
Member of the Committee: Prof. Alexandre José Malheiro Bernardino

October 2021

Resumo

Este trabalho tem o objetivo de fazer o projeto de uma rede de sensores aéreos composta por aviões de asa fixa não tripulados de forma a prestar vigilância e detetar sinais precoces de fogos florestais num certo território. Durante o trabalho um algoritmo foi adaptado para cobrir uma área uniformemente e distribuir os veículos da rede de acordo com uma função de risco florestal, dando prioridade a áreas de alto risco. O algoritmo é escalável para qualquer número de veículos e pode usar qualquer mapa de risco de incêndio que seja composto apenas por valores não negativos e limitados. Dois sistemas dinâmicos facilmente associados ao movimento de UAVs de asa fixa foram propostos, e simulações em Simulink e também em ROS foram implementadas de forma a simular um ambiente mais próximo da vida real. Por último, de forma a desenhar um workflow para ajudar à decisão do número de UAVs, altidude de voo e sensores, um modelo probabilistico da probabilidade de deteção de uma ignição foi formulado com o intuíto de maximizar essa probabilidade.

Keywords: Veículos aérios não tripulados, Controlo Cooperativo, Vigilância Autónoma

Abstract

This thesis aims to design an Aerial Sensor Network composed of fixed-wing unmanned aircraft in order to perform surveillance and detect early signs of a wildfire in a given territory. In this work, an algorithm is adapted to uniformly cover a given area and distribute the vehicles in the network depending on the fire hazard risk over the domain to be covered, prioritizing areas with a higher risk. This algorithm is scalable to any number of aircraft and can use any kind of fire hazard risk map that only contains bounded, positive values. Two different dynamical models associated with the movement of fixed-wing UAVs are proposed and are implemented with simulations in Simulink, as well as a ROS implementation to simulate an environment closer to a real world application. Lastly, a probabilistic model is formulated in order to design a workflow to help with the sizing of the fleet and the flight altitude to maximize the probability of detection of wildfires, both via the singular UAVs and the full sensor network.

Keywords: Unmanned Aerial Vehicles, Cooperative Control, Autonomous Surveillance

Acknowledgments

I would like to thank my family, for all the support during these stressful months, even when I was not the best person to be around of, as well as my supervisors who were a fountain of knowledge and patience whenever I needed any kind of help, and went above and beyond what was expected of them to help me complete this work.

Contents

List of Tables

List of Figures

Nomenclature

Notation

Bold Bold letters or symbols are either vectors or variables.

Non Bold Non Bold letters or symbols are scalar variables.

Mathematical Operations

$\langle f, g \rangle$ Inner product between two functions f and g defined as $\int_B f(x)g(x)dx$.

a $\cdot$ **b** Inner product between two vectors **a** and **b**.

Commonly used letters

$\boldsymbol{\lambda}$ Costates for the Optimal Control Problem.

μ Measure of the domain to be surveilled

μ_k Set of Fourier Coefficients for the Fourier Transformation of μ.

ϕ Metric for uniform coverage.

C^t Average probability distribution for the position of each agent in the network.

c_k Set of Fourier Coefficients for the Fourier Transformation of C^t.

f_k Set of Fourier Basis Functions.

k Wave vector for the Fourier Transform.

K Maximum wave number.

N Number of UAVs in the network.

Subscripts

j Denotes the j-th agent in the network.

k Denotes one wave vector.

Superscripts

$\top$ Denotes the transpose of a matrix or vector.

* Marks the variable is optimal.

Chapter 1

Introduction

1.1 Motivation

During 2017, the burnt area in Portugal saw a 428% increase relative to its mean values, as reported in [1]. In particular, the wildfire of Pedrógão Grande resulted in a burnt area of nearly 45.039 acres [1], causing nearly 500 million euros of estimated damage to the Portuguese government [2]. In fact, Portugal holds the largest percentage of burned area in Europe and is also the country in Europe with the highest number of ignitions per 1000 inhabitants. When compared to Spain, which has a similar climate and vegetation, Portugal has 6 times more fires per 1000 inhabitants [3].

While the cause of the previously mentioned fire was unrelated to human activity, the rural fire activity in Portugal is very often caused by humans. Between the years of 2011 and 2021, 48% of the fires with known causes were caused by inappropriate use of controlled fire as reported by Instituto da Conservação da Natureza e das Florestas in [4]. More specifically, 31% are caused by controlled burns of forest debris and agriculture excess that by negligence had gotten out of control. In addition to that, 23% of the fires are caused by criminal activity. This data is further corroborated in [5], which expanded the scope to all fire types instead of just rural and obtained similar proportions.

The consequences of human negligence and human criminal activity is that fires of small dimension start on the forest surface and spread until they become uncontrollable.

In light of these statistics, it is of extreme importance that an efficient method to cover an area under high risk of ignition and subject to frequent human activity is developed, so that the proper authorities are made aware of the ignitions before they become uncontrollable.

1.2 Literature Review

The ideal fire detection solution would be one that maintains continuous coverage of an area and is able to identify all sources of ignition. Geostationary satellites provide the first requirement and are already used to monitor environmental phenomena. However, due to the high orbit altitude, approximately 36000 km, the resolution of the obtained data is very low. One example of this is the VAS sensor of the

GOES satellite which produces samples of wildfires every three hours with a resolution between 7 and 14 Km on the infrared wavelength [6]. There is the possibility of reducing the satellite's orbit altitude to increase the spatial resolution but there are less availability of such services.

A lower altitude example is the data in gathered in [1], taken from the EOS-MODIS sensor with a resolution of 250m, but is limited to only getting 4 overpasses in a 24 hour period. In addition, data from such measurements can be corrupted by cloud cover [7].

Ground-based detection systems are much cheaper than satellite solutions and are not susceptible to corruption from cloud cover. These provide continuous coverage using visible and subvisible optical sensor, but have very low mobility and are limited to line-of-sight measurements, which can be obstructed by terrain features [8].

This work focuses on Aerial Sensor Networks (ASNs) consisting of fixed-wing Unmmaned Aerial Vehicles (UAVs) as a means to cover a given area. Unlike satellite solutions, the deployment of an ASN allows firefighting personnel to focus on a particular area of interest by obtaining high resolution data [9].

The problem of patrolling using an ASN can be roughly split into two separate fields of research: area coverage and resource allocation. For a given area to be covered, the area coverage fields deals with ways to optimally covering the allocated region, while the field of resource allocation deals with allocating the different vehicles of the ASN optimally between different subsets of the area to be covered.

For the purpose of this work, optimal allocation of resources means that regions of higher risk are to be monitored more often than areas with a lower risk so that resources are not being wasted at any given time in zones where an ignition is unlikely to start.

Many different measures of risk can be found in [10]. The different measures of risk are mostly based on the economical damage that a wildfire at any given location would cause, weighted by the probability of an ignition in said area. In the case of the Portuguese territory, different measures of risk have been considered in the past, including, for example:

- The approach in [11], which provides a method for the construction of an ignition probability map based on landscape and human-related variables;
- The Canadian Fire Weather Index (FWI), which combines the probability of an ignition with a wildfire propagation model in order to evaluate the risk of a wildfire, as described in [12]. This index is currently used by Instituto Português do Mar e da Atmosfera (IPMA) in Portugal [13];
- The method in [14], which builds a hazard map from a wildfire susceptibility and a wildfire probability maps.

Both the approaches in [11] and [14] have difficulty assessing the risk caused by human presence, something that is detrimental to any ASN using risk maps generated using these approaches, since the majority of the fires in Portugal is caused by humans (see [15], [5], [4]), as previously mentioned.

With the risk function for a given area, there needs to be a method to guide the agents in the ASN to places with higher risk values more often. The survey [16] describes a number of different strategies for both single and multi-robot area coverage. This problem is related to the so-called "lawnmower problem" which consists on the determination of a path that covers a given area and the "watchman's route

problem" which determines the best route to scan the given area. These problems become increasingly difficult as constraints are added to system, such as the presence obstacles, field of view (FOV) restrictions and dynamical constraints. While UAVs are usually not constrained by the presence of obstacles (if their altitude is high enough), they may be subject to dynamical constraints and the onboard sensors may be subject to FOV restrictions. Some of these problems are addressed in [17] and, notably, the authors in [18] follow an approach where point-mass vehicles are routed to areas of higher priority more often.

As stated before, these coverage algorithms can be used in cooperative control with different constraints and objectives. The work in [9] solves the vehicle routing problem optimally with respect to a cost function that minimizes the time it takes to cover a given area, while taking into account the limitations of the vehicles. In [19] and [20] a team of quad-copters is coordinated so as to perform continuous surveillance of a given area while avoiding collisions.

1.3 Objectives and outline

The objective of this work is to develop a real time trajectory generation algorithm that is capable of allocating resources from a ASN of fixed-wing UAVs dynamically based on the fire hazard risk map on a certain domain, with the goal of maximizing the probability of detecting an ignition. The algorithm can adapt to changes in the risk map so human activity can be taken into consideration. The evaluation of the human risk is not inside the scope of the project, but given the real time data feeds that ASNs are capable of providing, the same algorithm could serve as the basis of such future work.

In order to achieve this goal, the approach used in [18] was chosen as a base. The control law developed in the article does exactly what is necessary: it takes a risk map with very little constraints and guides a network of point-mass agents towards domain zones with high priority in real time. An advantage of this approach is that the risk map can be changed at any time during the mission and the network is able to adapt instantly and keep working as intended. The dynamical systems used in [18] can not be used directly in a real mission, however, because they do not take into account vehicle dynamics. In this work, the control law is adapted in order to take into consideration the dynamics of Fixed-Wing UAVs.

The work is divided into the following four chapters:

- **Uniform Coverage with Fixed-Wing Unmanned Aerial Vehicles:** In this section, all the concepts and variables needed to obtain the final control law are defined and explained, as well as the definition of the dynamical systems that describe the motion of an UAV. The goal of this section is to obtain the optimal control law for the desired systems;

- **Simulink Simulation:** In this section, the control laws obtained in the Theoretical Work section will be tested in a MATLab/Simulink environment for validation and proof that the adaptation of the original work was done properly;

- **ROS/Gazebo Simulation:** In this section, the physics simulator Gazebo working with the Robot Operating System (ROS) is used in order to simulate the model in a situation closer to real life to prove the real time capabilities of the control law;
- **Early Wildfire Detection:** In this section, a probabilistic model will be developed to take into account sensor capabilities, as well as to define exactly what needs to be detected and how it will be detected. The goal of this section is to use the data acquired in the previous chapters to provide a model that helps with the sizing of the ASN, namely the flight altitude and number of aircrafts.

Chapter 2

Uniform Coverage with Fixed-Wing Unmanned Aerial Vehicles

2.1 Theoretical Background

Given this work's objective of designing an algorithm for uniform coverage provided by an Aerial Sensing Network, it is important to first define what uniform coverage means and how to decide when the coverage is uniform.

In [18], the trajectory is deemed uniform if the dynamical system as a whole possesses ergodicity. Ergodicity is a property of dynamic and stochastic systems that relates the spatial averages of functions to its time averages over a trajectory. That means in a space with a known measure, such as a probability density function, the time average of the trajectory will converge to the space measure [21]. In other words, the time spent by an agent of a ergodic dynamical system on a certain space subset is the same as the measure of said subset.

As a result, if the trajectory of the network makes the system ergodic, such a trajectory is deemed uniform and results in an uniform coverage. The primary concern is then how far a system is from being ergodic and how to control the trajectory so the system is ergodic.

2.2 The metric for uniform coverage

Given a set of N agents for an ASN, let the following distribution C^t be defined:

$$C^t(\mathbf{x}) = \frac{1}{Nt} \int_0^t \sum_{j=1}^{N} \delta(\mathbf{x} - \mathbf{x}_j(\tau)) d\tau, \tag{2.1}$$

for each $\mathbf{x} \in \mathbb{R}^n$, where $\mathbf{x}_j : [0, t] \to \mathbb{R}^n$ represents the trajectory of the j-th agent for each $j \in \{1, 2, \ldots, N\}$ and $\delta(.)$ is the Dirac delta distribution.

Let g and f be two bounded function on B, the inner product between the two functions is defined as:

$$\langle f, g \rangle \ = \int_B f(x)g(x)dx. \tag{2.2}$$

The inner product of the distribution C^t with a bounded generic function f can be written as follows [18]:

$$\langle C^t, f \rangle \ = \frac{1}{Nt} \int_0^t \sum_{j=1}^{N} f(\mathbf{x}_j(\tau))d\tau. \tag{2.3}$$

The distribution C^t can be thought of as a probability density function because $\langle C^t, 1 \rangle \ = 1$. This means C^t can be seen as the probability density function function for the position of the agents in the ASN after a time period of t seconds.

Let μ be the measure of a rectangular space with the domain $U \subset \mathbb{R}^n$, with $\mu = 0$ outside of U. For μ to be a measure, it needs to be bounded and non-negative in its domain U. A necessary and sufficient condition to ergodicity is that the time average along a trajectory converges to the average spatial measure [21], so for the system to be ergodic, the convergence of C^t to μ is required. The condition for ergodicity is then:

$$\lim_{t \to \infty} \langle C^t, f \rangle \ = \ \langle \mu, f \rangle$$

for all bounded functions $f \ : \ U \ \rightarrow \ \mathbb{R}$. This is also the definition for weak convergence of the two functions [18].

2.2.1 Application to a 2D square domain

Given those generalised definitions, let us focus on the case of a 2D square domain, $U = [0, L] \times [0, L]$ for L a positive real number. This is convenient because the domain the ASN needs to cover is a 2D domain.

Let f_k be a Fourier basis functions for the 2D dimensional domain U so that,

$$f_k(\mathbf{x}) \ = \ cos(k_1 x)cos(k_2 y) \tag{2.4}$$

for each $\mathbf{x} = (x, y) \in \mathbb{R}^2$, with

$$k_1 = \frac{K_1 \pi}{L} \text{ and } k_2 = \frac{K_2 \pi}{L}$$

for each $K_1, K_2 = 0, 1, 2, ..., K$, where K is a positive integer, known as the maximum wave number. The subscript k denotes one instance the wave vector $\mathbf{k} = (k_1, k_2)$, for a total of $(1 + K)^2$ wave vectors, given that every combination of k_1 and k_2 needs to be considered. The function (2.4) was used because it satisfies the von Neumann boundary conditions of having a null gradient in the boundaries of domain U.

Using the Fourier basis functions f_k defined in (2.4), the Fourier coefficients for C^t and μ are given

by

$$c_k(t) = \frac{\langle C^t, f_k \rangle}{\langle f_k, f_k \rangle} = \frac{1}{Nt} \frac{\sum_{j=1}^{N} \int_0^t f(\mathbf{x}_j(\tau)) d\tau}{\langle f_k, f_k \rangle} \tag{2.5}$$

$$\mu_k = \frac{\langle \mu, f_k \rangle}{\langle f_k, f_k \rangle}, \tag{2.6}$$

respectively.

To quantify how far the current trajectory is from making the system ergodic, the distance between the two functions C^t and μ is measured using the following metric (cf. [18]):

$$\phi(t) = \left\| W_0^t - \mu \right\| = \sum_k \Lambda_k \left| s_k(t) \right|^2, \tag{2.7}$$

where $s_k(t) = c_k(t) - \mu_k$ and

$$\Lambda_k = \frac{1}{(1 + \|\mathbf{k}\|^2)^{\frac{3}{2}}}. \tag{2.8}$$

In [22] it is shown that $\phi(t)$ decays to zero if and only if C^t weakly converges to μ. This metric captures the deviation between the time averages and the space averages, and thus how far the system is from being ergodic. It will then be used as the metric for uniform coverage.

The fact that the metric for uniform coverage can be obtained from the Fourier coefficients of the desired measure μ means that there is the possibility of changing said coefficients during a mission, opening the possibility of changing the desired measure during the mission.

2.3 Fixed-wing airplane models

In order to minimize the metric for uniform coverage, the authors in [18] decided upon designing an control law. There were two different dynamical systems that were used in the paper. Both systems represent a point mass particle, one first order system, and one second order system. The systems are:

$$\dot{\mathbf{x}}_j = u_j$$

$$\ddot{\mathbf{x}}_j = -c\dot{\mathbf{x}}_j + u_j,$$

where $\mathbf{x}_j$ represents the position of the j-th particle, u_j is the control input and $c > 0$ is a damping factor.

Neither of these dynamical systems consider the properties and constraints of a fixed-wing aircraft. In the first system, the control input is the direct velocity of the aircraft and, without any constraints on the rate at which the velocity changes and what kind of changes are allowed by the aircraft, can lead to trajectories with very sharp turns, something that is shown in the results section in [18].

The second model could be usable if the ASN consisted of quad-rotor UAVs, which have a higher freedom of movement when compared to fixed-wing UAVs. For example, it is impossible for a fixed-wing UAV flying in a straight line to decelerate and change its direction without making a turn, something that would be possible with a quad-rotor and that the second order model allows.

In order to adapt the approach in [18] to an Aerial Sensor Network with Fixed-Wing UAVs, a different dynamical system needs to be considered that takes into account the movement constraints of the aircraft.

2.3.1 Dubin's Vehicle

In this work, it is considered that trajectories of the agents are always in the xOy plane and at a fixed flight altitude, where xOy is part of the NED (North-East-Down) local Inertial frame. As a result, the first system to be considered is that of a Dubin's Vehicle. This system is a described by simple model, which considers that a vehicle moves in a combination of circular arcs and straight lines, with the only control input being the angular velocity of the vehicle's heading angle.

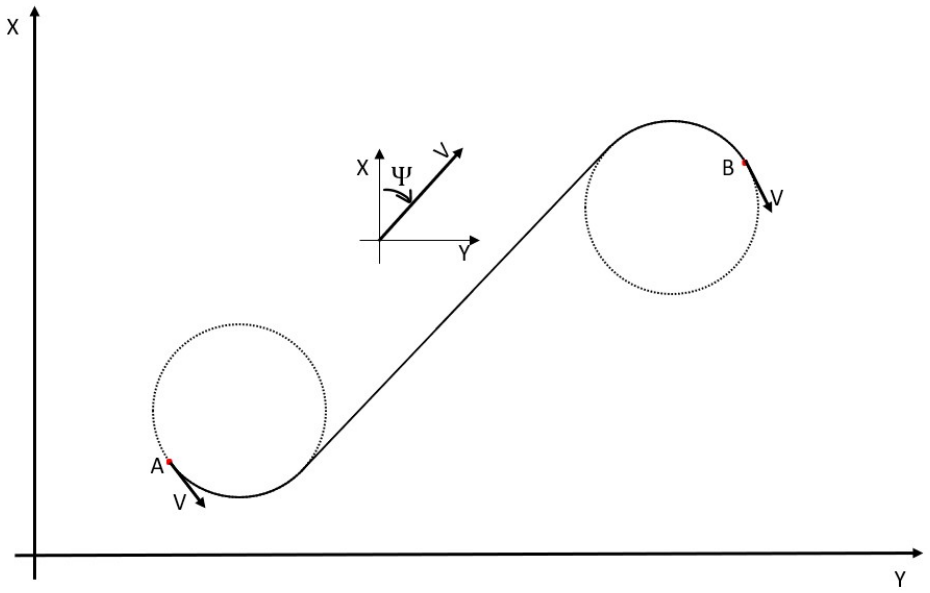


Figure 2.1: Illustration of a Dubin's Vehicle trajectory and of the local coordinate frame.

The Dubin's Vehicle motion is interesting because this kind of motion can be associated with the simple coordinated turn model, which directly relates the heading turn ratio with a fixed-wing's roll angle [23]. For the purpose of this work, the optimal path between two points isn't the goal, but the fact that any movement given by its equations are easily followed by a fixed-wing UAV. With the angles in accordance with the ones in Figure 2.1, the equations of motion for the j-th vehicle are the following:

$$\begin{aligned} \dot{\mathbf{x}}_j &= V_g \mathbf{M}(\psi_j) \\ \dot{\psi}_j &= u_j. \end{aligned} \tag{2.9}$$

where V_g is a constant scalar value for the vehicle's ground speed, $\mathbf{x}_j$ the vehicle's position, ψ_j is its heading angle, and $\mathbf{M}(\psi_j)$ the projection vector, given by:

$$\mathbf{M}(\psi_j) = \begin{bmatrix} \cos(\psi_j) \\ \sin(\psi_j) \end{bmatrix}. \tag{2.10}$$

2.3.2 Adapted Dubin's Vehicle

While the previous model generates trajectories that are easily followed by a fixed wing UAV, the control input is a scalar value, which has its own problems as discussed in Section 3.2. As a result, it would be ideal that a similar model be used but with a control input that is not a scalar value, but rather a vector. For this purpose, a model previously studied in [24] was considered.

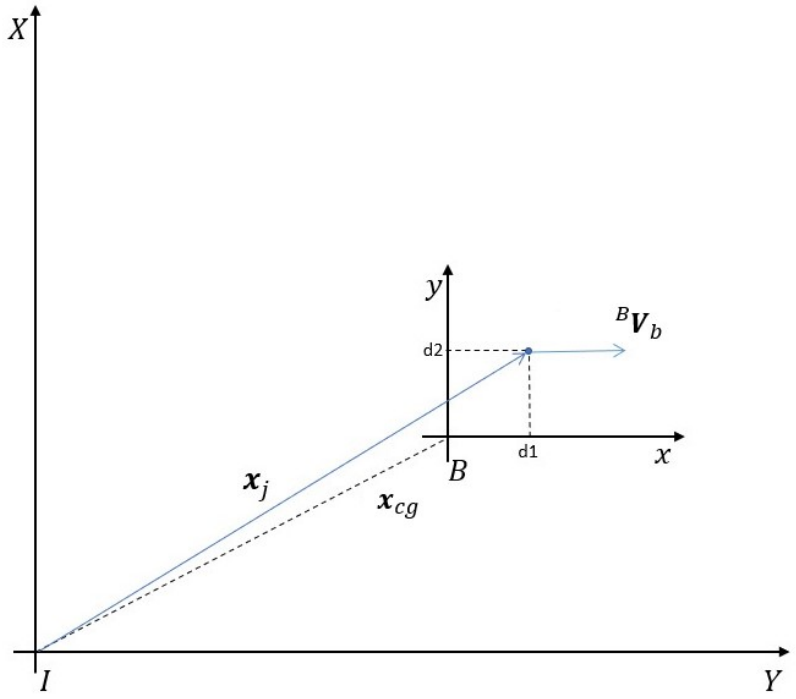


Figure 2.2: Adapted Vehicle's Coordinate Frame.

In this model, instead of tracking the center of mass of the vehicle, a point slightly ahead is considered. The same NED reference frame is used for the inertial frame and a rotating body frame centered on the vehicle's center of mass is considered for the vehicle body, with the linear speed always aligned with the body frame's xx axis so that ${}^B\mathbf{V}_b = [(v_b + \Delta v) \;\; 0]^\top$, where $v_b \in \mathbb{R}$ is a constant value and Δv is an input for the model that belongs to $[-\delta v, \delta v]$ for some positive δv. The position of the tracked point in the body frame is a parameter of the system, $\mathbf{d}_j = [d_1 \; d_2]^\top$.

As a result, the tracked dot's position, regarding the j-th agent in the network, $\mathbf{x}_j$ in the inertial frame is calculated using the following expression:

$$\mathbf{x}_j = \mathbf{x}_{cg} + \mathbf{R}(\psi_j)\mathbf{d}_j \tag{2.11}$$

and its derivative,

$$\dot{\mathbf{x}}_j = \mathbf{R}(\psi_j){}^B\mathbf{V}_b + \dot{\mathbf{R}}(\psi_j)\mathbf{d}_j \tag{2.12}$$

where $R(\psi_j)$ is the rotation matrix:

$$\mathbf{R}(\psi_j) = \begin{bmatrix} \cos(\psi_j) & -\sin(\psi_j) \\ \sin(\psi_j) & \cos(\psi_j) \end{bmatrix}$$

Assuming a model constraint for the maximum heading rate of vehicle, such that $\dot{\psi}_j \in [-u_{max}, u_{max}]$ and a maximum speed differential of $\Delta v \in [-\delta v, \delta v]$, and with $\mathbf{u}_j = [u_1 \ u_2]^\top$ the unitary control input vector, the system (2.12) can be re-written as follows:

$$\dot{\mathbf{x}}_j = \mathbf{A}(\psi_j) v_b + \mathbf{M}_2(\psi_j)\mathbf{u}_j$$

with $\mathbf{M}_2$ and $\mathbf{A}$ matrices that are functions of the heading angle ψ_j and the different parameters of the system such that,

$$\mathbf{M}_2(\psi_j) = \begin{bmatrix} \delta v \cos(\psi_j) & -u_{max}(d_1 \sin(\psi_j) - d_2 \cos(\psi_j)) \\ \delta v \sin(\psi_j) & u_{max}(d_1 \cos(\psi_j) - d_2 \sin(\psi_j)) \end{bmatrix} \quad \text{and} \quad \mathbf{A}(\psi_j) = \begin{bmatrix} \cos(\psi_j) \\ \sin(\psi_j) \end{bmatrix}$$

for each $\psi_j \in \mathbb{R}$.

In this model, the determinant of matrix $\mathbf{M}_2(\psi_j)$ is given by $\det(\mathbf{M}_2(\psi_j)) = \delta v \cdot d_1 \cdot u_{max}$ and it is different than zero for every heading value as long as long as δv, u_{max} nor d_1 are not zero themselves, making it so there are never problems with matrix inversion.

Another noteworthy fact is that both inputs are directly available in the first derivative of the UAV positions directly, which is very important in the next section.

2.4 Feedback Control Law

Calculating the optimal control law is done by solving the optimal control problem over a finite time horizon $[t, t + \Delta t]$ when Δt tends to zero. This approach was employed in [18] and is explained in [25]. In order to do this, it is more convenient to express the quantities from Section 2.2 without the time factor and the number of agents in the denominator, by multiplying the quantities by Nt as follows:

$$\begin{aligned} C_k(t) &= \frac{\sum_{j=1}^{N} \int_0^t f(\mathbf{x}_j(\tau))d\tau}{\langle f_k, f_k \rangle} = Ntc_k(t) \\ U_k(t) &= Nt\mu_k \\ S_k(t) &= C_k(t) - U_k(t) = Nts_k(t) \\ \Phi(t) &= \sum_k \Lambda_k \left| S_k(t) \right|^2 = N^2 t^2 \phi(t). \end{aligned}$$

To solve the optimal control problem, an appropriate cost function must be picked and the Hamiltonian equation formulated. For this purpose, the chosen cost function to minimise is the derivative of the metric for uniform coverage $\Phi(t+\Delta t)$, namely $\dot{\Phi}(t+\Delta t)$, which means that the goal is to make the next position

of the agent so that the metric for uniform coverage decays as much as possible, leading to the lowest possible value of the metric for uniform coverage. This derivative is as follows,

$$\dot{\Phi}(t+\Delta t) = \sum_{k} \Lambda_k S_k(t) \frac{dS_k(t)}{dt}. \tag{2.13}$$

From (2.13), it is known that the derivatives $S_k(t)$ are of fundamental importance:

$$\frac{dS_k(t)}{dt} = \frac{\sum_{j=1}^{N} f_k(\mathbf{x}_j(t))}{\langle f_k, f_k \rangle} - N\mu_k \tag{2.14}$$

$$\frac{d^2 S_k(t)}{dt^2} = \frac{\sum_{j=1}^{N} \nabla f_k(\mathbf{x}_j(t)) \cdot \dot{\mathbf{x}}_j(t)}{\langle f_k, f_k \rangle} \tag{2.15}$$

where $\nabla f_k(.)$ is the gradient vector of the Fourier basis function, given by:

$$\nabla f_k(\mathbf{x}_j(t)) = \begin{bmatrix} \frac{\partial f_k}{\partial x} \\ \frac{\partial f_k}{\partial y} \end{bmatrix} = \begin{bmatrix} -k_1 \sin(k_1 x(t)) \cos(k_2 y(t)) \\ -k_2 \cos(k_1 x(t)) \sin(k_2 y(t)) \end{bmatrix} \tag{2.16}$$

for each $\mathbf{x}_j(t) \in \mathbb{R}^2$. The derivatives are important and are used to extend the systems later to minimize the Hamiltonian.

From the derivatives of $S_k(t)$, it can be concluded that different dynamical systems might require further derivatives so that the control input is directly included, because it may or may not be directly in the first derivative of the position. In a system such as the simple first-order system in [18] or the adapted Dubin's Vehicle model, the control input is present in the first derivative of the trajectory in its entirety, and as a result accessible on the first derivative of the metric for uniform coverage while in systems like Dubin's Vehicle, the control input is only present on the second derivative of the metric for uniform coverage.

For all the models, the chosen cost function to minimize is:

$$J(t, \Delta t) = \dot{\Phi}(t + \Delta t) \tag{2.17}$$

In this specific case, without a Lagrangian function in the cost, the necessary conditions for optimality of the control problem in the interval $[t_0, t_f]$ are explained in [25] and are as follows:

$$\dot{\mathbf{x}}^*(t) = \frac{\partial \mathcal{H}}{\partial \lambda}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \tag{2.18}$$

$$\dot{\boldsymbol{\lambda}}^*(t) = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \tag{2.19}$$

$$\mathcal{H}(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \leq \mathcal{H}(\mathbf{x}^*(t), \mathbf{u}(t), \boldsymbol{\lambda}^*(t), t), \tag{2.20}$$

where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are generic system states and control inputs, $\boldsymbol{\lambda}(t)$ are the costates for the minimization and $\mathcal{H}(\mathbf{x}(t), u(t), \boldsymbol{\lambda}(t), t)$ is the Hamiltonian function. This, of course, for every $t \in [t_0, t_f]$ and every admissible control input. The * superscript means optimality, such as the optimal input $\mathbf{u}^*$. Finally, to

obtain the final value of the costates [25],

$$[\frac{\partial J}{\partial \mathbf{x}}(\mathbf{x}^*(t_f), t_f) - \boldsymbol{\lambda}^*(t_f)]^T \delta \mathbf{x}_f + [\mathcal{H}(\mathbf{x}^*(t_f), \mathbf{u}^*(t_f), \boldsymbol{\lambda}^*(t_f), t_f) + \frac{\partial J}{\partial t}(\mathbf{x}^*(t_f), t_f)]\delta t_f = 0 \tag{2.21}$$

2.4.1 Dubin's vehicle control law

Let us then consider the extended system for the Dubin's Vehicle, adding the equations for the auxiliary states $S_k(t)$ and $O_k(t) = \dot{S}_k(t)$:

$$\begin{aligned} \dot{\mathbf{x}}_j &= V_a \mathbf{M}(\psi_j) \\ \dot{\psi}_j &= u_j \\ \dot{S}_k(t) &= O_k(t) \\ \dot{O}_k(t) &= \sum_{j=1}^{N} \frac{\nabla f_k(\mathbf{x}_j(t))}{\langle f_k, f_k \rangle} \cdot \dot{\mathbf{x}}_j(t) \end{aligned} \tag{2.22}$$

Using the additional states, the cost function is then:

$$J(t + \Delta t) = \sum_k \Lambda_k S_k(t + \Delta t) O_k(t + \Delta t) \tag{2.23}$$

Now, let the costates be ${}^j\lambda_1(\tau) \in \mathbb{R}^2$ and ${}^j\lambda_2(\tau), {}^k\lambda_3(\tau), {}^k\lambda_4(\tau) \in \mathbb{R}$ for all N agents and for all k wave number vectors.

The Hamiltonian is defined for each $\tau \in [t, t + \Delta t]$ and written as follows:

$$\begin{aligned} \mathcal{H}(x_j, \psi_j, S_k, O_k, u_j, {}^j\lambda_1, {}^j\lambda_2, {}^k\lambda_3, {}^k\lambda_4, \tau) = V_a \sum_{j=1}^{N} {}^j\lambda_1 \cdot \mathbf{M}(\psi_j) + \sum_{j=1}^{N} {}^j\lambda_2 u_j + \sum_k {}^k\lambda_3 O_k \\ + V_a \sum_k {}^k\lambda_4 \frac{\sum_{j=1}^{N} \nabla f_k(\mathbf{x}_j)}{\langle f_k, f_k \rangle} \cdot \mathbf{M}(\psi_j) \end{aligned} \tag{2.24}$$

where the time dependencies on the states were omitted for brevity. Given the Hamiltonian (2.24), the dynamic equations for the costates are as follows:

$${}^j\dot{\lambda}_1(\tau) = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}_j} = -V_a \sum_k {}^k\lambda_4(\tau) \frac{\nabla^2 f_k(\mathbf{x}_j(\tau))}{\langle f_k, f_k \rangle} \mathbf{M}(\psi_j(\tau)) \tag{2.25}$$

$${}^j\dot{\lambda}_2(\tau) = -\frac{\partial \mathcal{H}}{\partial \psi_j} = -V_a \sum_{j=1}^{N} {}^j\lambda_1(\tau) \cdot \frac{\partial \mathbf{M}}{\partial \psi_j} - V_a \sum_k {}^k\lambda_4(\tau) \frac{\sum_{j=1}^{N} \nabla f_k(\mathbf{x}_j(\tau))}{\langle f_k, f_k \rangle} \cdot \frac{\partial \mathbf{M}}{\partial \psi_j} \tag{2.26}$$

$${}^k\dot{\lambda}_3(\tau) = -\frac{\partial \mathcal{H}}{\partial S_k} = 0 \tag{2.27}$$

$${}^k\dot{\lambda}_4(\tau) = -\frac{\partial \mathcal{H}}{\partial O_k} = -{}^k\lambda_3(\tau) \tag{2.28}$$

for each $\tau \in [t, t + \Delta t]$. In this case, the partial derivative of $\mathbf{M}$ with respect to ψ_j is the following:

$$\frac{\partial \mathbf{M}}{\partial \psi_j} = \begin{bmatrix} -\sin(\psi_j) \\ \cos(\psi_j) \end{bmatrix}.$$

To obtain the final values of the costates in this particular problem, where the time interval shrinks to zero, (2.21) is used and δt_f considered zero. The vector x^* is the optimal value for the states in (2.22) and λ^* is the optimal value for the costates, in this case a vector containing all the costates. As a result, the optimal final values for the costates are given as follows

$$\lambda^*(t_f) = \frac{\partial J}{\partial x}(x^*(t_f), t_f)$$

With $t_f = t + \Delta t$, the final values for the costates can be calculated as follows:

$$^j\lambda_1(t + \Delta t) =^j \lambda_2(t + \Delta t) = 0 \tag{2.29}$$

$$^k\lambda_3(t + \Delta t) = \Lambda_k O_k(t + \Delta t) \tag{2.30}$$

$$^k\lambda_4(t + \Delta t) = \Lambda_k S_k(t + \Delta t) \tag{2.31}$$

The control input u_j in this model is bounded, its upper and lower values being u_{max} and $-u_{max}$, respectively. As such, the optimization of the optimal control problem is done by minimizing the Hamiltonian, seen in (2.20). The control law is then given by:

$$u_j^*(\tau) = \underset{||u_j(\tau)|| \leq u_{max}}{\arg\min} \mathcal{H}(x_j, \psi_j, S_k, O_k, u_j, ^j\lambda_1, ^j\lambda_2, ^k\lambda_3, ^k\lambda_4, \tau). \tag{2.32}$$

for each $\tau \in [t, t+]$. The solution to that equation is trivial, as only the second term of the Hamiltonian contains the control input. As such, the optimal control law can be written as follows:

$$u_j^*(\tau) = -\frac{^j\lambda_2(\tau)}{||^j\lambda_2(\tau)||} u_{max}. \tag{2.33}$$

for each $\tau \in [t, t + D\Delta t]$.

Given the costate's dynamic equations and the final values, when Δt tends to zero, the value for $\lambda_2(\tau)$ can be approximated using basic calculus,

$$^j\lambda_2(t) = \lim_{\Delta t \to 0} {}^j\lambda_2(t + \Delta t)$$

where

$$\begin{aligned} ^j\lambda_2(t) &\approx {}^j\lambda_2(t + \Delta t) - {}^j\dot{\lambda}_2(t + \Delta t)\Delta t \\ &= 0 + [V_a \sum_{j=1}^{N} {}^j\lambda_1(t + \Delta t) \cdot \frac{\partial \mathbf{M}}{\partial \psi_j} + V_a \sum_k {}^k\lambda_4(t + \Delta t) \frac{\sum_{j=1}^{N} \nabla f_k(\mathbf{x}_j(t + \Delta t))}{\langle f_k, f_k \rangle} \cdot \frac{\partial \mathbf{M}}{\partial \psi_j}]\Delta t. \end{aligned}$$

The optimal control law can be rewritten as follows,

$$u_j^*(t) = \lim_{\Delta t \to 0} -\frac{{}^j\lambda_2(t)}{||{}^j\lambda_2(t)||} u_{max}. \tag{2.34}$$

Calculating the limit, and with the auxiliary variable $\beta_j(t)$ defined as

$$\beta_j(t) = V_a \sum_k \Lambda_k S_k(t) \frac{\sum_{j=1}^N \nabla f_k(\mathbf{x}_j(t))}{\langle f_k, f_k \rangle} \cdot \frac{\partial \mathbf{M}}{\partial \psi_j}, \tag{2.35}$$

the final control law for the Dubin's Vehicle is the following expression,

$$u_j^*(t) = -u_{max} \text{sign}(\beta_j(t)) \tag{2.36}$$

Just by looking at Equation (2.36) one of the features of this model is apparent, and that is that the optimal control law for a 1D bounded control input resulting from the minimization of an Hamiltonian with no quadratic terms only yields two possible results.

As it stands, the control input can only assume values of its bounds and is not a smooth function, something that can prove difficult in the control of a real aircraft, due to the aircraft needing to make very aggressive maneuvers to try and follow an input, if it is possible at all.

2.4.2 Adapted Dubin's vehicle control law

The approach to obtaining the optimal control law for the adapted Dubin's vehicle is exactly the same as the approach used in the previous chapter, only the system used is different and there is one less costate.

Let us then consider the extended system for the Adapted Dubin's Vehicle, adding equations for the auxiliary states $S_k(y)$ and $O_k(t) = \dot{S}_k(t)$:

$$\begin{aligned} \dot{\mathbf{x}}_j(t) &= \mathbf{A}(\psi_j) v_b + \mathbf{M}_2(\psi_j) \mathbf{u}_j \\ \dot{S}_k(t) &= O_k(t) \\ \dot{O}_k(t) &= \sum_{j=1}^N \frac{\nabla f_k(\mathbf{x}_j(t))}{\langle f_k, f_k \rangle} \cdot \dot{\mathbf{x}}_j(t) \end{aligned} \tag{2.37}$$

Recovering the matrix definitions,

$$\mathbf{M}_2 = \begin{bmatrix} \delta v \cos(\psi) & -u_{max}(d_1 \sin(\psi) - d_2 \cos(\psi)) \\ \delta v \sin(\psi) & u_{max}(d_1 \cos(\psi) - d_2 \sin(\psi)) \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} \cos(\psi) \\ \sin(\psi) \end{bmatrix}$$

Using the cost function in (2.23),

$$J(t) = \sum_k \Lambda_k S_k(t) O_k(t)$$

and introducing analogous costates ${}^j\lambda_1(\tau) \in \mathbb{R}^n$ and ${}^k\lambda_2(\tau), {}^k\lambda_3(\tau) \in \mathbb{R}$ for all N agents and K wave

vector harmonics, the Hamiltonian equation can be written as follows,

$$\mathcal{H}(x_j, S_k, O_k, u_j, {}^j\lambda_1, {}^k\lambda_3, {}^k\lambda_4, \tau) = \sum_{j=1}^{N} {}^j\lambda_1 \cdot [\mathbf{A}v_b] + \sum_{j=1}^{N} {}^j\lambda_1 \cdot [\mathbf{M}_2(\psi_j)\mathbf{u}_j] \tag{2.38}$$

$$+ \sum_k {}^k\lambda_2 O_k$$

$$+ \sum_k {}^k\lambda_3 \frac{\sum_{j=1}^{N} \nabla f_k(\mathbf{x}_j)}{\langle f_k, f_k \rangle} \cdot [\mathbf{M}_2(\psi_j)\mathbf{u}_j + \mathbf{A}v_b] \tag{2.39}$$

for each $\tau \in [t, t + \Delta t]$, where the time dependencies were again omitted for brevity.

The costate dynamics obtained from the Hamiltonian are as follows:

$${}^j\dot{\lambda}_1(\tau) = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}_j} = -\sum_k {}^k\lambda_3(\tau) \frac{\nabla^2 f_k(\mathbf{x}_j(\tau))}{\langle f_k, f_k \rangle} \mathbf{M}_2(\psi_j(\tau))\mathbf{u}_j(\tau) \tag{2.40}$$

$${}^k\dot{\lambda}_2(\tau) = -\frac{\partial \mathcal{H}}{\partial S_k} = 0 \tag{2.41}$$

$${}^k\dot{\lambda}_3(\tau) = -\frac{\partial \mathcal{H}}{\partial O_k} = -{}^k\lambda_2(\tau) \tag{2.42}$$

for each $\tau \in [t, t + \Delta t]$.

The process to obtain the final value of the costates is analogous to the process to obtain the values for the previous model, namely based on (2.21) and δt_f considered zero. As a result, the optimal final values for the costates are given as follows

$$\lambda^*(t_f) = \frac{\partial J}{\partial x}(x^*(t_f), t_f)$$

with $t_f = t + \Delta t$, the final values for the costates can be trivially calculated.

$${}^j\lambda_1(t + \Delta t) = (0, 0)^T \tag{2.43}$$

$${}^k\lambda_2(t + \Delta t) = \Lambda_k O_k(t + \Delta t) \tag{2.44}$$

$${}^k\lambda_3(t + \Delta t) = \Lambda_k S_k(t + \Delta t) \tag{2.45}$$

The control input is defined in this model as a vector with maximum norm equal to one, with the amplitudes of velocity and heading turn rate dictated by parameters and as such not included in the input. As a result, the boundaries for the input are $\mathbf{u}_j(t) \leq 1$. The optimal control input is then

$$\mathbf{u}_j^*(\tau) = \underset{||u_j(\tau)|| \leq 1}{\arg\min}\ \mathcal{H}((x_j, S_k, O_k, u_j, {}^j\lambda_1, {}^k\lambda_3, {}^k\lambda_4, t) \tag{2.46}$$

The Hamiltonian in this case contains two terms with $\mathbf{u}_j$, and as such, the minimization of the equation given the bounded input is trivial again, given $\mathbf{u}_j$ is never quadratic or higher. For a cleaner expression, let's define the auxiliary variable $\beta_j(\tau)$ so that

$$\beta_j(\tau) = \mathbf{M}_2^T(\psi_j(\tau))^j\lambda_1(\tau) + \mathbf{M}_2^T(\psi_j(\tau))\sum_k {}^k\lambda_3(\tau)\cdot\frac{\sum_{j=1}^N \nabla f_k(\mathbf{x}_j(\tau))}{\langle f_k, f_k\rangle}$$

The transpose of $\mathbf{M}_2$ appears so that the input $\mathbf{u}_j$ is isolated and still a column vector in the Hamiltonian, using the properties of the inner product. The following identity was used:

$$\mathbf{q}^T\mathbf{W}\mathbf{u} = \mathbf{u}^T\mathbf{W}^T\mathbf{q}$$

where **q** and **u** are generic column vectors of the same dimension and **W** is a generic square matrix of appropriate size. As a result, $\beta_j(\tau)$ is a column vector.

Given the bounded nature of the input, the optimal control law can be written as

$$\mathbf{u}_j^*(t) = -\frac{\beta_j(t)}{||\beta_j(t)||} \tag{2.47}$$

Knowing the costate dynamic equations and the final values, when Δt tends to zero, the $\beta_j(t)$ can be written with first order accuracy with basic calculus,

$$\beta_j(t) = \lim_{\Delta t\to 0}\beta_j(t+\Delta t)$$

where

$$\begin{aligned}\beta_j(t+\Delta t) &= \mathbf{M}_2^T(\psi_j(t+\Delta t))\times[{}^j\lambda_1(t+\Delta t) + \sum_k {}^k\lambda_3(t+\Delta t)\cdot\frac{\sum_{j=1}^N \nabla f_k(\mathbf{x}_j(t+\Delta t))}{\langle f_k, f_k\rangle}]\\ &= \mathbf{M}_2^T(\psi_j(t+\Delta t))\times\sum_k \Lambda_k S_k(t+\Delta t)\cdot\frac{\sum_{j=1}^N \nabla f_k(\mathbf{x}_j(t+\Delta t))}{\langle f_k, f_k\rangle}\\ &- \mathbf{M}_2^T(\psi_j(t+\Delta t))\times[{}^j\dot{\lambda}_1(t+\Delta t) + \sum_k {}^k\dot{\lambda}_3(t+\Delta t)\cdot\frac{\sum_{j=1}^N \nabla f_k(\mathbf{x}_j(t+\Delta t))}{\langle f_k, f_k\rangle}]\times\Delta t\end{aligned}$$

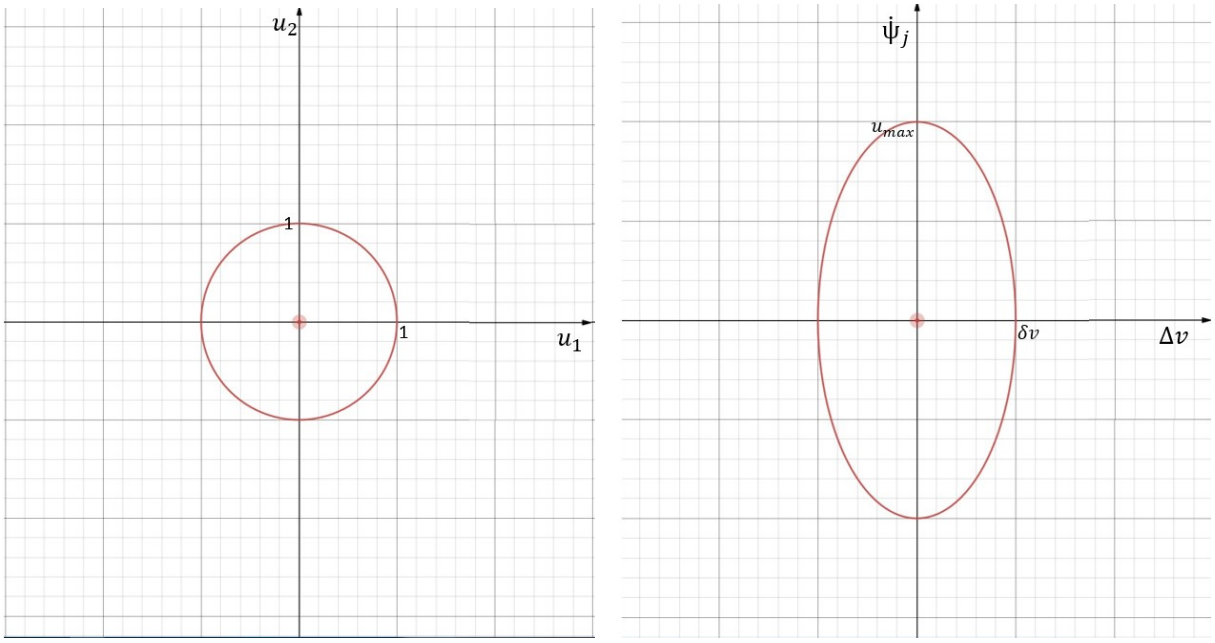
This stems from the following first order approximation for a general $f(t)$ function,

$$f(t) = f(t+\Delta t) - \dot{f}(t+\Delta t)\Delta t$$

Computing the limit yields

$$\beta_j(t) = \mathbf{M}_2^T(\psi_j(t))\sum_k \Lambda_k S_k(t)\frac{\sum_{j=1}^N \nabla f_k(\mathbf{x}_j(t))}{\langle f_k, f_k\rangle}. \tag{2.48}$$

Unlike the previous model, the control input now has multiple dimensions, and as such, considering all the functions inside are smooth, the control input is smooth as well. The input is constrained to the unitary circle, with its effects translated to Δv and $\dot{\psi}$ in the model depending on the chosen parameters, as depicted in Figure 2.3.



(a) Possible Control Inputs

(b) Possible effect on the model

Figure 2.3: Possible control inputs and its correspondent effect in the model.

Chapter 3

Trajectory Generation Simulink Simulation and Results

In this chapter, simulation results for the models that are described in the previous chapter will be shown, as well as the parameters chosen for further work and performance comparison between the models.

3.1 Target probability density function

3.1.1 Fire Hazard Risk

The concept of uniform coverage based control laws obtained in Section 2.4 is the convergence of the time average along a trajectory into the desired spatial measure. The spatial measure is then one of the inputs of the algorithm.

As previously mentioned in Section 2.2, the only restriction applied to this measure is that it needs to be bounded, without infinite values in the space domain that needs to be surveilled.

For the purpose of this work, it is convenient to think of this measure as something that relates to fire hazard risk. A measure that would make sense is the Fire Hazard Risk Index, a value calculated daily by IPMA. According to the methodology document publicly available [13], this value is a number from one to five, encompassing the different risk levels from reduced to maximum, calculated using various data such as weather prediction and the amount of fuel on the ground. An example of such a map is given in Figure 3.1.

It is to note that the pixel resolution in IPMA's calculations is only of one kilometer, and would not be enough for the purpose of monitoring small to medium sized areas, if needed. However, the classification method can be utilized to make custom risk maps based on the knowledge of the owner of the surveillance system.

On a more practical level, there is no information regarding the fire hazard probability or fire rate and its relation to the risk index beside the fact that a moderate index would have a higher probability than a

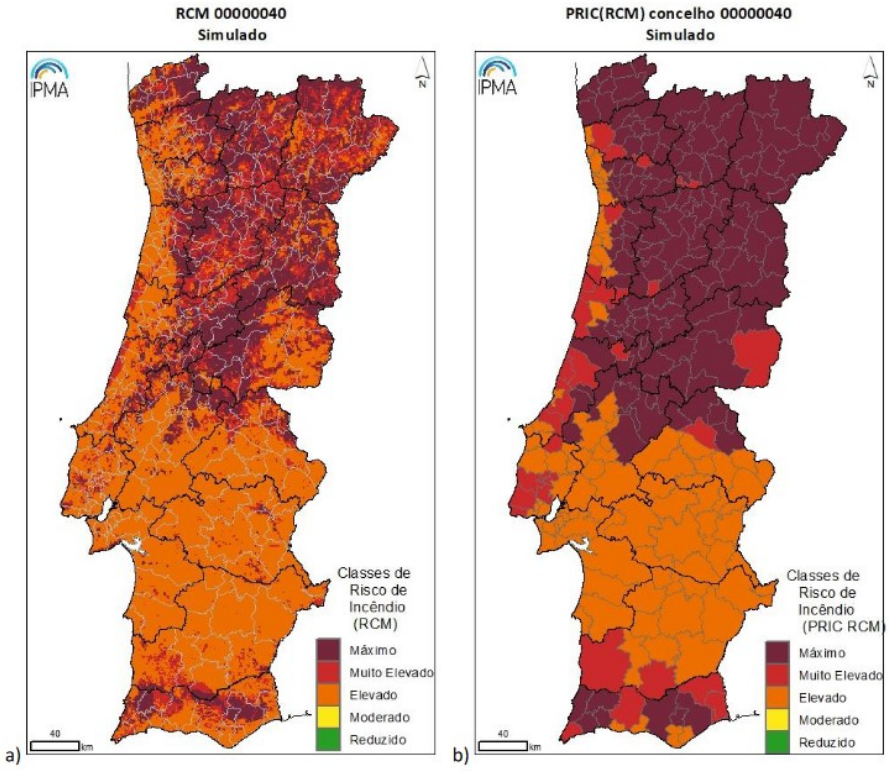


Figure 3.1: Simulated fire hazard index by pixel resolution (a) and county (b) by IPMA [13]

reduced index, for example. Consequently, relative numbers are the easiest to use and a very practical choice that can be easily tweaked.

The fact that the Fourier Transform is applied to the measure and only its coefficients are used in the algorithm makes it so the relative numbers must not be so close together as to be indistinguishable in the reconstruction of the original function. The closer the relative numbers, the easier it is for the sinusoidal nature of the reconstruction to mix the different risk zones.

For the purpose of the simulation, the values for relative risk from Table 3.1 are proposed and were used.

Table 3.1: Relative risk levels for custom risk map.

Classification	**Risk (Relative)**
Reduced	1
Moderate	2
High	4
Very-high	8
Maximum	16

To understand the effect of these numbers in the context of this work, assigning the classifications *Reduced* and *Moderate* to two area zones of the same size would mean that, ideally, the UAV network would spend twice as much time monitoring the *Moderate* area as it would monitoring the *Reduced* area. Additionally, the results would be the same if instead of those classifications, any two adjacent classifications were used because each classification has a relative risk twice that of the last classification.

Depending on the number of classifications used in the entire surveillance area, the relative risk can be seen as a measure of time spent per area.

3.1.2 Designing the target probability density function

Considering that further in the work there will be a probability analysis, it is convenient that the measure for the Fire Hazard Risk is expressed as a probability density function with a volume of one in the entire domain.

Let $U = [0, L] \times [0, L] \subset \mathbb{R}^2$ be the square target domain for surveillance and $\mu : \mathbb{R}^2 \to \mathbb{R}$ the target probability density function. Using an auxiliary function, $risk : \mathbb{R}^2 \to \mathbb{R}$ which assigns a relative risk number to every point in the domain, we can have the target probability density function defined as

$$\mu(x, y) = \frac{risk(x, y)}{\iint_U risk(X, Y) dX dY} \tag{3.1}$$

for every $(x, y) \in U$. The function is zero for every (x, y) outside of the domain U.

The Fourier coefficients usable by the control law are recovered from (2.5):

$$\mu_k = \frac{\langle \mu, f_k \rangle}{\langle f_k, f_k \rangle}. \tag{2.5}$$

The number of harmonic wave vectors is an important factor and depends on the desired resolution and can heavily affect computation time. With the highest wave number K, the number of integrations to obtain the Fourier coefficients from (2.5) alone is $(K + 1)^2$ for each time step.

The resolution, on the other hand, is heavily influenced by the characteristics of the UAV used for surveillance. There is no point in having a one meter resolution if the UAV can only make coordinated turns with a ten meter radius, for example.

For simulation purposes, the domain picked is a $2000m \times 2000m$ domain with a cell resolution of $100m \times 100m$.

The risk distribution used is shown in Figure 3.2.

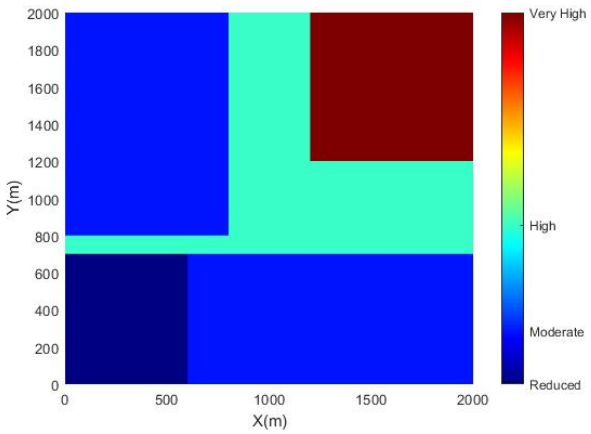


Figure 3.2: Base risk function for simulation

The function used has only four levels, with the maximum being *Very High*.

3.1.3 Disadvantages brought by Fourier symmetry and how to solve them

While Figure 3.2 shows a very reasonable probability risk map, the Fourier Transform needs to be applied it to have any use in the control law.

However, one of the characteristics of the Fourier Transform when the Fourier Basis in (2.4) are used is its periodicity, or rather the symmetry on the edges. This is showcased in Figure 3.3 (a). Additionally, both control laws heavily depend on the gradient of the basis function to decide what the next input is. The result is that if no other modification is made, the UAV would very quickly leave the intended surveillance domain and keep working ”properly” according to the decrease in metric, as the metric does not differentiate between the periodic areas.

In [18], this is not a problem for one of the models developed in the article, the single integrator model from Chapter 2, a model in which the control input directly controls the linear velocity of the particle, because there is no lag in the response and the boundary conditions of the Fourier base function would make it so the closer to the boundary, the closer to zero the gradient would be. In this case, there would never be a linear velocity value that would point to the outside of the domain while the particle is on the boundary.

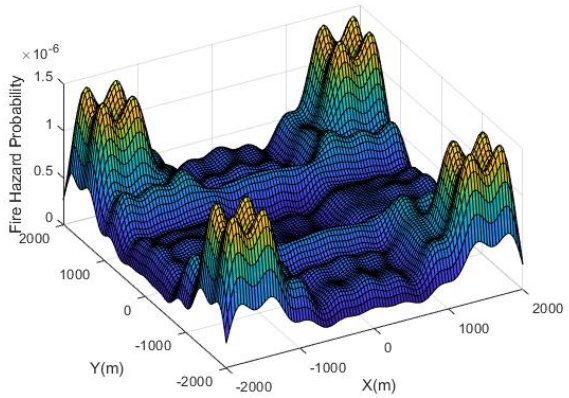
In the case of this work, the control input effect on the position is delayed, and even if the gradient vector gets close to zero in the boundary of the domain, there would often be cases where the UAV would not be able to turn swiftly enough and the maneuver would lead it outside of the domain. When outside of the domain, the metric and the gradient would keep functioning as if nothing is wrong and the UAV would stay in that periodic zone until it eventually left again. For illustration, Figure 3.3 (b) shows what happens if no countermeasures are employed when running the simulations detailed in Section 3.3. In the figure, the plotted squares represent the different reflections of the $2000 \times 2000m$ domain to be covered.

There are two proposed solutions to this issue:

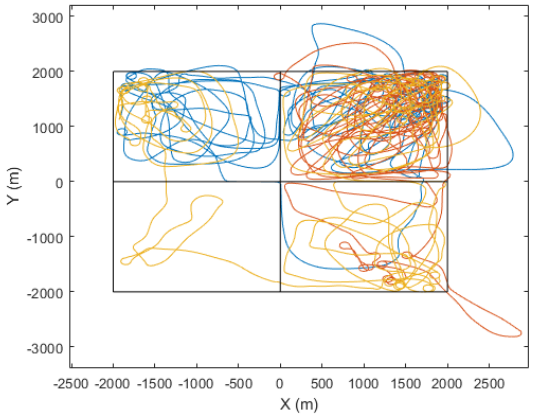
- Forcing the vehicle to turn back to the initial domain, effectively overpowering the control law,
- Extending the domain of the Fourier Transform with the desired distribution surrounded by zero values.

The first proposed solution, while working, would make it so every time the UAV would be outside the domain, the trajectory would stop being optimal in the sense that it would not be controlled by the optimal control law anymore. This would result in an undesirable increase in the metric for uniform coverage. Testing this solution alone showed that often times the vehicle would go outside the domain again right after coming back, creating a loop that made the UAV stay in the boundary of the domain for long periods of time.

The second proposed solution requires an increase in the number of harmonics used in the Fourier Transform to keep the shape of the desired probability density function. This is not a problem with



(a) The Fourier Transform of the Fire Hazard Risk Distribution for $K = 15$ in a larger domain.



(b) The trajectories obtained by running the simulation from Section 3.3 without countermeasures for the agents leaving the domain.

Figure 3.3: The Fourier Transform of the Fire Hazard Risk Distribution for $K = 15$ in a larger domain and the trajectories obtained by running the simulation from Section 3.3 without countermeasures for the agents leaving the domain.

current hardware, either on-board or off-board, as the increase in harmonics increases the computation time linearly and not exponentially, as shown in Section 3.3.1.

There is another issue regarding the second solution, in the sense that the Fourier Transform is only an approximation of the risk probability density function and not the exact function, meaning that a value of zero on the outside would never be quite zero. Depending on the parameters of the model, there could still be an small chance that the vehicle would find itself outside the domain, albeit a small one when the parameters or the model are well chosen, something that is discussed in Section 3.4.1.

Consequently, it makes sense that a combination of both is used. The domain of the distribution is extended to $3000m \times 3000m$ with the previous function centered in this domain, and should the UAV leave the extended domain, it is forced inside. Since the probability on the boundaries of the extended domain is approximately zero and increasing approaching the actual coverage domain before the extension, it is very unlikely that the UAV would be stuck in the extended domain border in a loop. Figure 3.4 is the graphical representation of the extension of the domain.

In Figure 3.5, the Fire Hazard Probability and its Fourier Transform with $K = 15$ harmonics are represented. The small rectangle with the minimum cell resolution was inserted to make and example of how small changes influence the Fourier Transform. It is obvious that without a considerable amount of harmonics, the $100m$ width rectangle would not be very influential in the final result.

With this in mind, it would be useful in a real life example to ignore very small zones of smaller risk near relatively bigger high risk zones and just assign a similarly high risk to the smaller zone. In contrast, a very small zone of high risk is still very important and should be extended in the risk function to account for the Fourier Transform, otherwise the transform might decrease the risk the algorithm takes into account.

It is, in conclusion, imperative to understand that the Fourier approximation of the Fire Hazard probability density function is the important factor in the algorithm, not the pre-transform distribution.

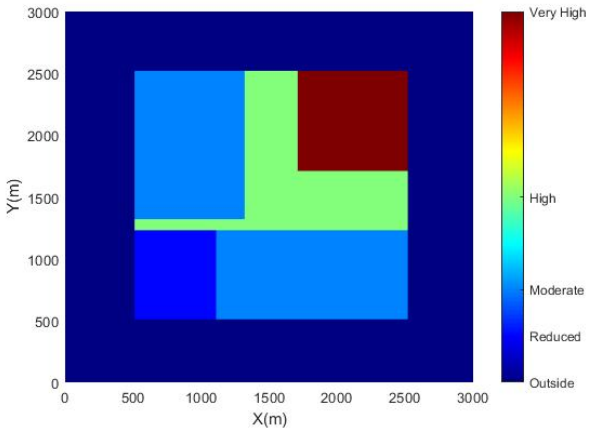
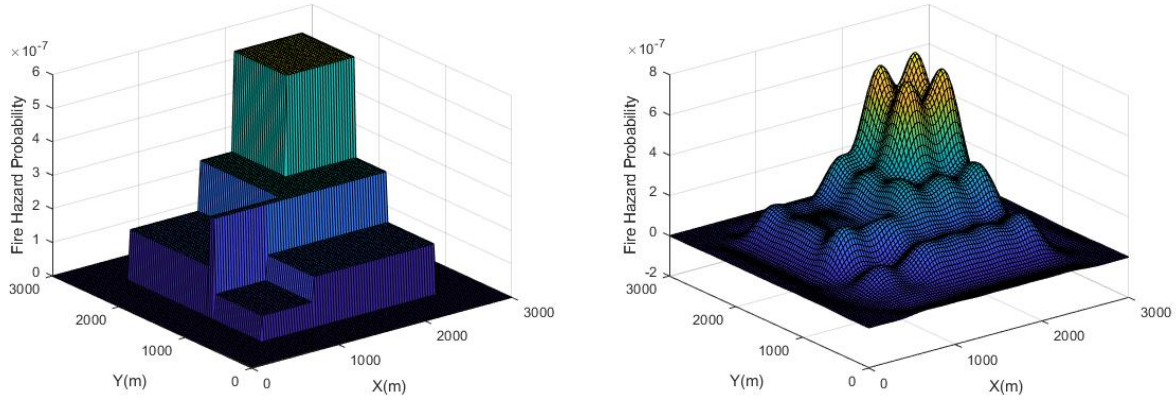


Figure 3.4: Risk function with an extended domain.



(a) Fire Hazard probability

(b) Fourier Transform of the Fire Hazard probability

Figure 3.5: Fire Hazard probability function and its Fourier approximation with $K = 15$.

3.2 Simulink Simulations

The Simulink simulations will only take into the account the models presented in Sections 2.3.1 and 2.3.2. While they do not represent the full workings of an aircraft, the simulations will not take into account any inner-loop controllers as there is no set aircraft to be used in the project.

The design of the inner-loop controller is outside of the scope of this thesis, but the algorithm can output either heading angles, heading rate, linear velocities or position waypoints. With this variety of variables, many off-the-shelf solutions can be used, such as a PX4 controller [26].

The block diagram in Figure 3.6 shows the basic version of the full controller when taking into account the aircraft inner-loop controllers. The section Simulink simulations will focus on the highlighted area. Every obtained result uses the ode4 solver with a fixed step size of $0.1s$

The Simulink model is a simple integrator block preceded by an *Interpreted MATLab function* block containing all the calculations necessary to obtain the control law.

The Fourier coefficients of the desired distribution as well as the inner product between the Fourier basis function and itself were calculated using two nested one dimensional *integral* functions.

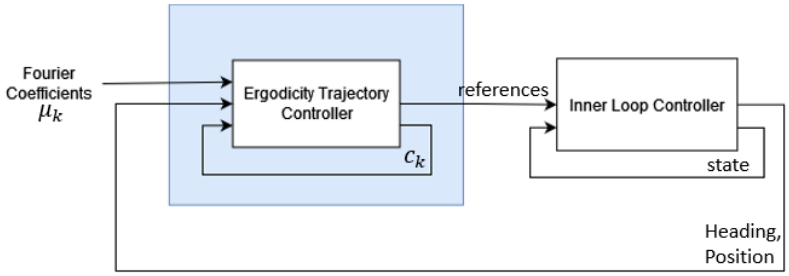


Figure 3.6: Control Scheme of a full aircraft

Name	Cruise speed	Altitude	Endurance
Albatross [27]	18 ms/	-	up to 5 hours
Jabali [28]	39 m/s	up to 6500m	up to 8 hours
PD-2 UAS VTOL [29]	-	up to 5000m	up to 10 hours
STREAM C VTOL [30]	28 m/s	up to 5000m	up to 10 hours
Bramor ppX [31]	16 m/s	up to 5000m	up to 3.5 hours

Table 3.2: Compilation of different Fixed-Wing surveillance drones and some of its characteristics.

Regarding basic vehicle characteristics for parameters such as trim speed, heading turn rate and flight time, an in depth market research, compiled in Table 3.2 showed commercial UAVs for surveillance have an endurance of hours. One specific case is the hybrid VTOL/Fixed-Wing aircraft Jabalis from Helvetis, with an endurance of up to eight hours flying at $39m/s$. Turn rates, considering the type of aircraft, are expected to go up to around $0.5 - 0.8rad/s$.

3.3 Dubin's Vehicle Model

For the Simulink simulation for the Dubin's Path Model, we considered an ASN composed of 3 UAVs with a maximum heading rate of $u_{max} = 0.5rad/s$ and a constant vehicle speed of $Va = 30m/s$ based on the UAV market research. The simulation was run for a period of $T = 3600s$, i.e., one hour.

The obtained trajectory is highlighted in Figure 3.7, with the different coloured lines depicting the different UAVs.

Straight away, it is evident that the solutions for the symmetry problem were effective. In fact, the fail-safe measure in the case the UAV was found outside of the bigger domain was never executed. The UAVs still spent some time on the outside of the actual surveillance domain, but only in the beginning of the simulation due to the nature of the control law prioritizing the lower frequency harmonics first, due to their weighting decreasing as the harmonic's number increases (Λ_k decreases as the index k increases, from equation 2.8).

The density of the lines is also higher in higher risk zones, and much lower in lower risk zones. The two corners in the $y = x$ diagonal are evidence of this, as the lower corner has the lowest Fire Hazard risk and the higher corner has the highest Fire Hazard risk.

It is also important to note that while the control input can only be one of two values, that the vehicles

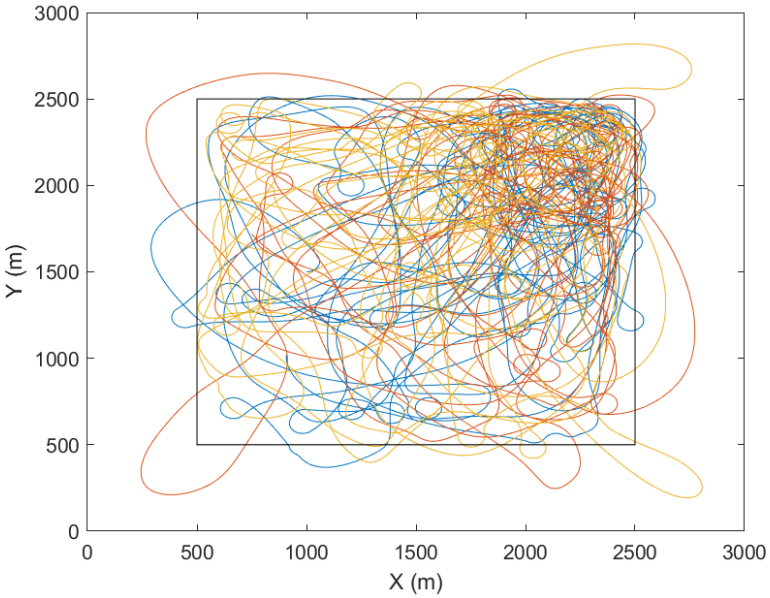


Figure 3.7: Obtained trajectory for 3 UAVs using the Dubin's Vehicle model.

do not move in perfect circles every time. This makes sense, as the control law is calculated every $0.1s$. For example, if in the period of 2 seconds the inputs were always the opposite of the previous one, the end result would have been something very similar to a straight line.

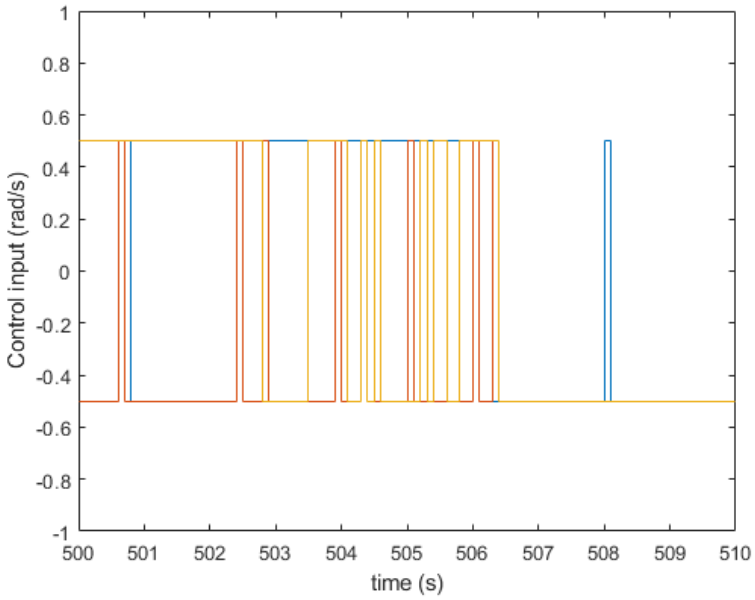
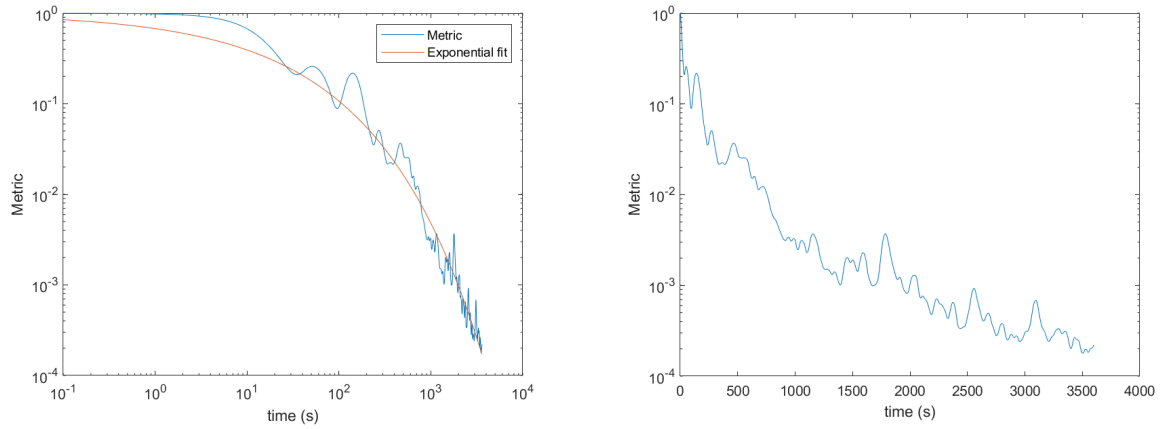


Figure 3.8: Control input sequence for the three UAVs in $t = [500, 510]s$.

In Figure 3.8, the control input sequence for all the UAVs is presented in a small excerpt of the simulation. The control input is, as can be expected, very erratic, and changing values very quickly, from the positive to negative end of the spectrum. Measures to try and overcome this issue were studied in Section 3.3.4

Some controllers can control a UAV solely given a reference turn rate, as it is very easy to translate

to a roll angle reference in a fixed-wing UAV. Regardless of the controller, any reference input would be hard to follow due to the discontinuities of the reference. This is not the only draw back, of course. Given the way the input is defined, the servos in any aircraft would need to be constantly changing their positions from one extreme to the other, leading to reduced vehicle lifespan, higher maintenance, and lower endurance due to constant use of the servo motors.



(a) metric for uniform coverage with both log scales

(b) metric for uniform coverage with y log scale

Figure 3.9: metric for uniform coverage representation for a one hour simulation.

In regards to the metric for uniform coverage in Figure 3.9, the first thing that needs to be said is that the graph is normalized against the metric in starting point, $\Phi(t = 0)$. This make it easier to compare the metric data against other models and variables, as the starting metric is decided by the size of the domain, the risk function and the initial vehicle positions.

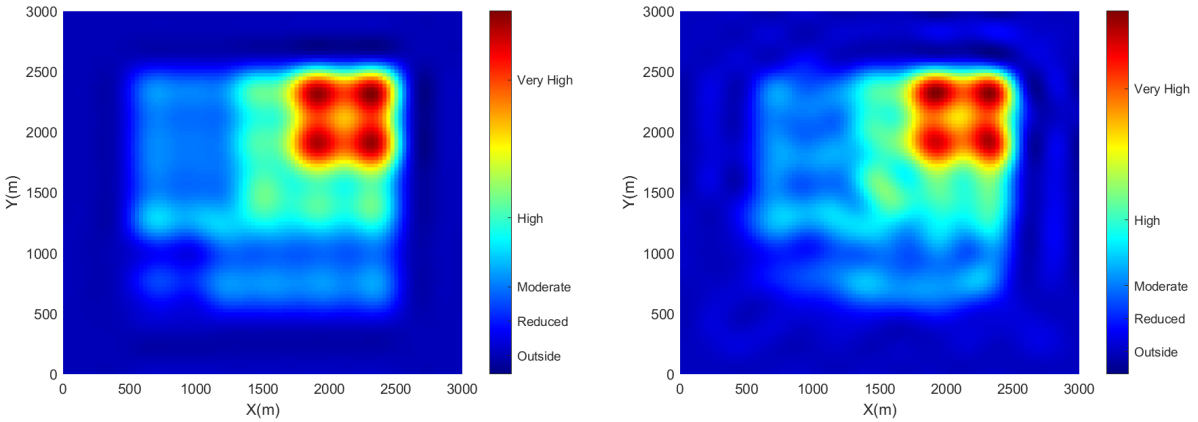
In terms of the shape of the graph, it is clear the function isn't monotonous. That would be impossible using this uniform coverage algorithm. On average, the metric appears to decrease exponentially in Figure 3.9 (a). This result is in line with the results obtained in [18], so it can be concluded that there wasn't any anomaly in application of the algorithm when changing the model.

The noise has clear peaks which immediately decrease, almost in a recurring fashion. This is the result of the time average being used. This is better explained when thinking of a uniform distribution in the entire domain. Even if the algorithm was perfect, and the distance between the functions was 0, the next movement of the UAV would increase the distance between the distributions, and it would only reach 0 again once the rest of the domain was visited again.

While the metric for uniform coverage is important, the shape of the obtained distribution is even more so. In Figure 3.10, the desired distribution and the obtained distribution are compared. The values do no perfectly align with the values from Figure 3.4, the distribution pre-Fourier Transform, but overall the different risk zones can be differentiated in the image.

3.3.1 Effect of the number of harmonics in the computational time

The process of running the simulations can be divided into two different stages: the setup stage, in which the Fourier coefficients of the desired distribution μ_k are obtained, as well as defining the initial



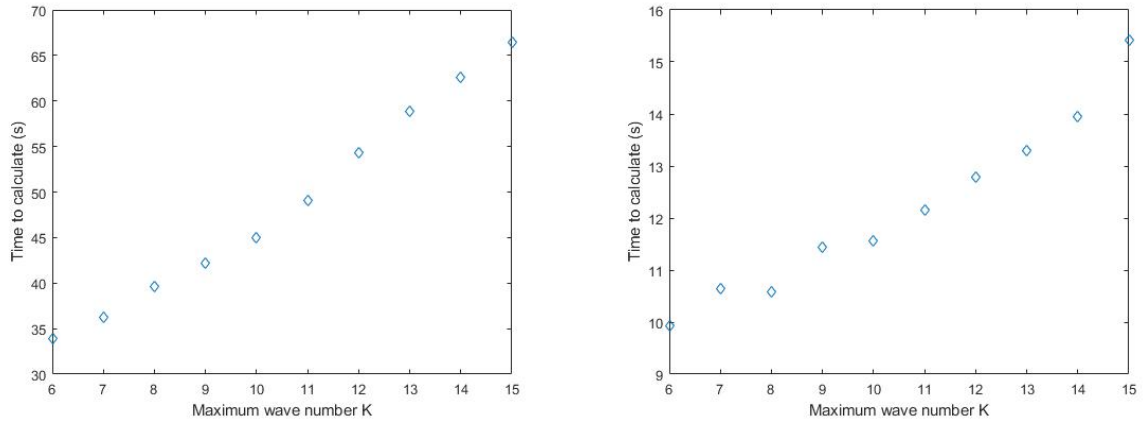
(a) Heatmap of the desired probability density function

(b) Heatmap of the obtained probability density function

Figure 3.10: Visualization of the desired probability density function's and the obtained probability density function's Heatmap.

conditions, and the simulation stage, where the Simulink model is run.

The number of harmonics used is directly related to computation time, both in setup and in simulation. For the setup stage, each harmonic adds two 2D integrals, one for the denominator and one for the numerator in (2.5). For the simulation, each harmonic effectively adds one state, which means one additional integral.



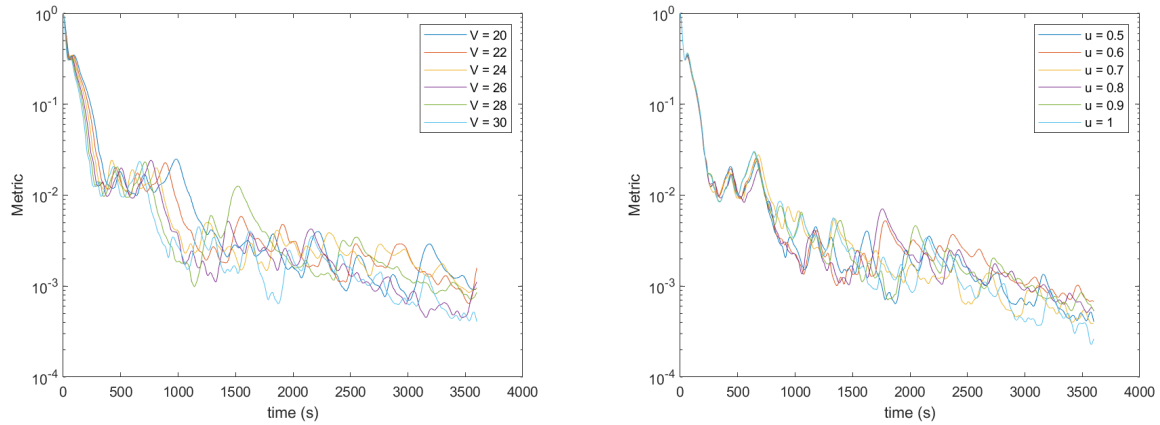
(a) Computational time in the setup stage for different values of K.

(b) Computational time in the simulation for different values of K.

Figure 3.11: Effect of the number of harmonics in the computational time, in the setup stage and in the simulation.

In Figure 3.11, the results for the computational time using different values of K are shown. The increase in computational time in the setup stage is linearly related to K, which is surprising because the number of harmonics is equal to $(1 + K)^2$. This means that increasing the maximum number of harmonics is not such a detriment to computational time. The same can be said to the time it takes for the simulation to run, even if there is a slight inconsistency on the lower number of harmonics. This can be attributed to the way Simulink calculates the integrals, which depends heavily on threshold values and numerical precision.

3.3.2 Effect of the constant speed and heading turn rate



(a) Effect of the linear speed on the Dubin Model.

(b) Effect of the heading turn rate on the Dubin Model.

Figure 3.12: Effect of the linear speed and heading turn rate on the Dubin Model.

In order to see how the basic parameters affected the metric, the simulation was run for different values of each of the parameters. Values from $V = 20m/s$ to $V = 30m/s$ were tested, as well as values from $u = 0.5rad/s$ to $u = 1rad/s$. The results were compiled in Figure 3.12.

The obtained results were very similar. There were no drastic differences in the metric for uniform coverage's behaviour, only that with each increment added to each variable, the averages seem to decrease slightly. This is more evident in the case of the linear speed.

In conclusion, the higher the linear speed and/or heading turn rate, the better the results. This is to be expected, of course, as with a higher linear speed, more of the domain can be covered in a shorter time span.

In the case of the heading turn rate, it can be concluded that a higher rate is better, as the minimization of the Hamiltonian was given for the bounded control input and the control law is directly proportional to the boundary values. The higher the absolute value of the maximum heading turn rate, the more negative the derivative of the metric for uniform coverage is allowed to be, resulting in a better performance.

Ultimately, the values for the linear speed and heading turn rate will be determined by the maximum the selected vehicle is able to provide.

3.3.3 Effect of the number of UAVs

The difference in the number of UAVs used in the simulation is very impactful.

In Figure 3.13 the results for varying the number of aircrafts between 1 and 5 in terms of the metric for uniform coverage are shown.

Invariably, increasing the number of UAVs increases the decay rate of the metric, although it is clear that there are two different groups of results, $N = 1, 2$ and $N = 3, 4, 5$. The clear difference between the results can be attributed to the area of the domain in combination with the speed at which the UAVs are allowed to move. The results clearly show the difference in the metric between 2 and 3 UAVs is

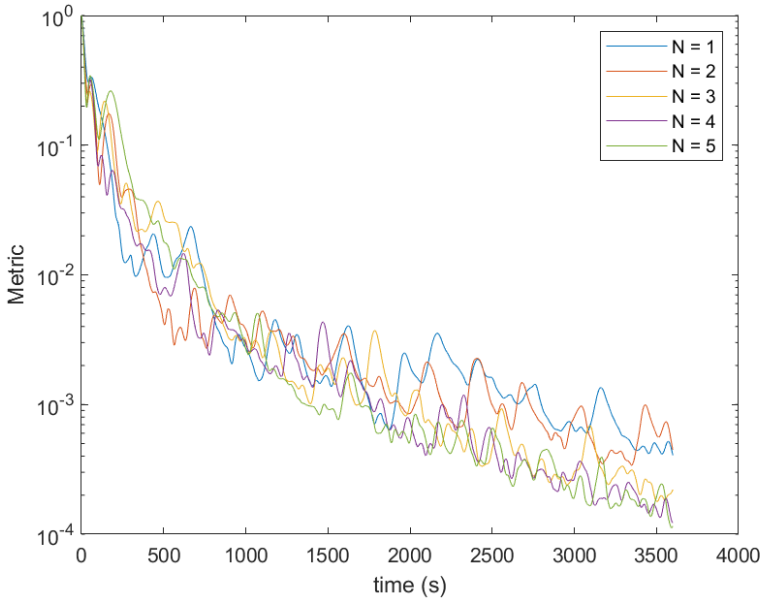


Figure 3.13: Effect of the number of UAVs in the metric for uniform coverage.

considerably bigger than the difference between 3 and 4 UAVs. It can be concluded that for these particular combination of parameters, the ideal size of the fleet would be 3 UAVs, as the performance would decrease heavily with less than 3 and stay relatively the same after 3 UAVs.

The difference between the results in the second group is the height of the cyclical peaks. The higher the number of UAVs, the lesser the height of the peak is. This is to be expected, as the metric is averaged over the number of aircrafts in the simulation and the cyclical increase in the metric is due to the problem constraints. The increase in UAVs also means the frequency of the up and down cyclical behaviour is higher, as more aircrafts would lead to a decrease in the time it would take to reach the minimum value in the metric for uniform coverage due to triple the coverage, even if it is average.

3.3.4 Regularization of the control input

In order to try and solve the issue of the irregular input, the control law from (2.36) was modified. Instead of using the $sign$ function, a saturation function was used, so that the regularized control law is defined as follows:

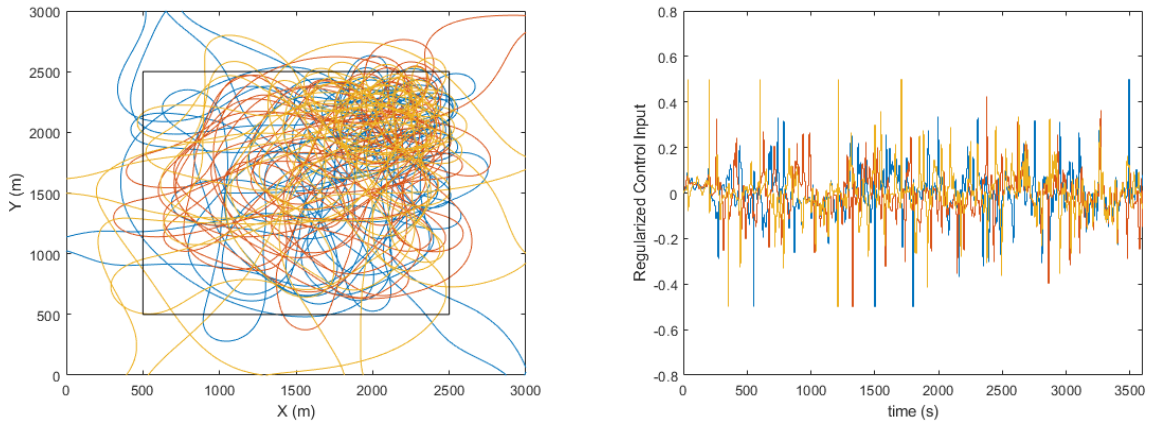
$$u_j(t) = \begin{cases} -u_{max} & \text{if } \gamma\,\beta_j(t) \geq u_{max} \\ -\gamma\,\beta_j(t) & \text{if } ||\gamma\,\beta_j(t)|| < u_{max} \\ u_{max} & \text{if } \gamma\,\beta_j(t) \leq -u_{max} \end{cases}, \tag{3.2}$$

where $\gamma > 0$ is a constant to tune the slope of the linear part of the saturation function.

The constant γ is then a parameter to be changed, however given the way the quantity $\beta_j(t)$ is defined in (2.35), it is nearly impossible to predict even its magnitude order. Furthermore, if the target distribution μ changes, or the domain changes its size, the order of magnitude of this quantity will also change because $S_k(t)$ is directly related to the Fourier coefficients of μ. The ideal value for γ would

be one that would allow for the control input to be almost always in the linear zone but have its values as close to the saturation limits as possible, meaning the higher the slope, the closest to optimal the regularized control law is.

While the order of magnitude is hard to predict, the equation for $\beta_j(t)$, (2.35), resembles the equation for the metric for uniform coverage, (2.7), so as a first approximation, γ was set to the inverse of the order of magnitude of the metric at $t = 0$. In the particular case of this simulation, this value is $\gamma = 10^{14}$.



(a) Trajectory

(b) Regularized control input

Figure 3.14: Trajectory and control input obtained for the regularized control input with $\gamma = 10^{14}$.

The trajectory of the different agents and their regularized control input throughout the simulation is showcased in Figure 3.14. The most noticeable effect of this specific regularization is that the agents spend a lot of time outside of the domain to be covered, even going so far as to frequently leave the extended domain. This is because the γ parameter is too low, causing the system to not have a control input high enough to make the needed turn. With the regularization with a low slope, the control input is much closer to zero than it is to the saturation point, which would be the optimal control input. This can be seen in Figure 3.14 (b), where the control input is now always smooth and rarely ever goes above 0.2, when the saturation point is 0.5. The values in the saturation zone are not due to the regularization but to the method employed to keeping the agents inside the extended domain.

With the poor results for the trajectory, it is expected that the obtained metric for uniform coverage is also inferior when compared to the optimal result in Figure 3.9. This is evident from Figure 3.15, with the normalised metric never dropping below 10^{-3} when the optimal result almost reaches the 10^{-4} values.

In order to try improve this, the slope was increased to $\gamma = 10^{15}$ because there was still the option to do so, with the control input being so far from the saturated values.

The trajectory result for this value of γ is showcased in Figure 3.16 (a), and it seems that the agents spend considerably less time outside of the domain to be covered, even if it is longer than if the input was optimal, and can still leave the extended domain. The regularized control input for the simulation is showcased in Figure 3.16 (b), and it can be seen that it reaches the saturation values very often, meaning that increasing the value of γ would only make it so the control input saturates more often, which would defeat the point of adding the saturation in the first place.

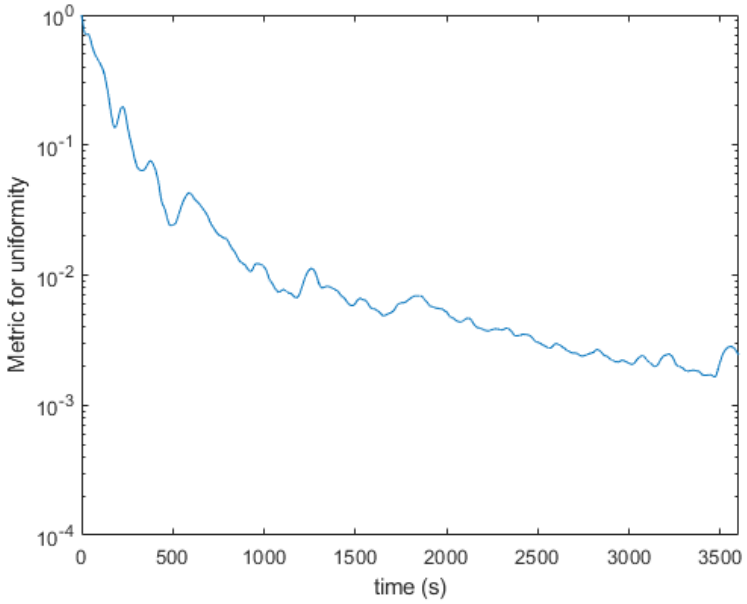
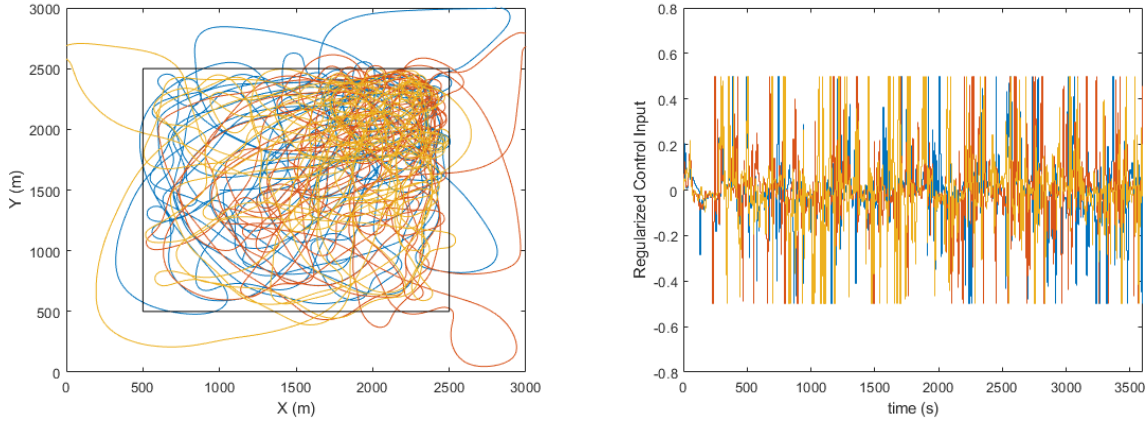


Figure 3.15: Metric for uniform coverage for a regularized input with $\gamma = 10^{14}$.



(a) Trajectory

(b) Regularized control input

Figure 3.16: Trajectory and control input obtained for the regularized control input with $\gamma = 10^{15}$.

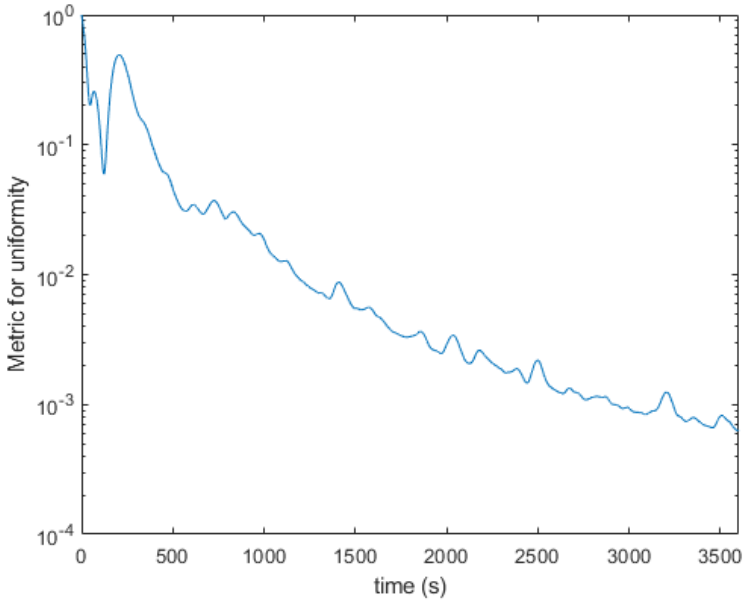


Figure 3.17: Metric for uniform coverage for a regularized input with $\gamma = 10^{15}$.

Regarding the metric for uniform coverage, shown in Figure 3.17, the values dropped bellow the 10^{-3} mark, but were not quite as low as the ones obtained from running the simulation with the optimal control law, in Figure 3.9.

Considering that it would be counter productive to increase the value of γ any more, and that the performance of the model using this regularization technique were not up to par with the optimal control law, it can be concluded that this approach, while able to fix the discontinuities in the control input, do not provide good enough results.

Furthermore, even if the obtained results were good, there is no formula that can generate the ideal γ value, and since it varies between distributions μ, should a method such as this be employed, it would remove the option of changing the distributions during the mission, effectively blocking one of the most valuable features of this algorithm, accounting for human activity in real time if needed.

3.4 Adapted Dubin's Vehicle Model

When the Adapted Dubin's Vehicle Model is introduced in Section 2.3.2 the conclusion that the y coordinate of the point in the body frame (d_2) doesn't affect the determinant of the final matrix is very important, as it can be set to zero and the system will still work perfectly fine. As such, with this d_2 value, point is always situated in the xx axis of the body frame.

For this simulation, the base speed of the vehicle was set to $v_b = 30m/s$ and its maximum heading turn rate set to $u_{max} = 0.5rad/s$ the same values used in the previous simulation.

For the values specific to this model, the distance d_1 was set to $2m$, and the speed differential was set to $\delta v = 5m/s$. The effects of varying the distance of the point will be studied further into this work, in Section 3.4.1.

Choosing the value for δv needs to be done carefully because it directly relates to the minimum speed of the UAV. In this case, the minimum speed would be $V_{min} = v_B - \delta v = 25m/s$. The value V_{min} needs to be above the aircraft stall speed. The time span of the simulation was, again, one hour, $T = 3600s$.

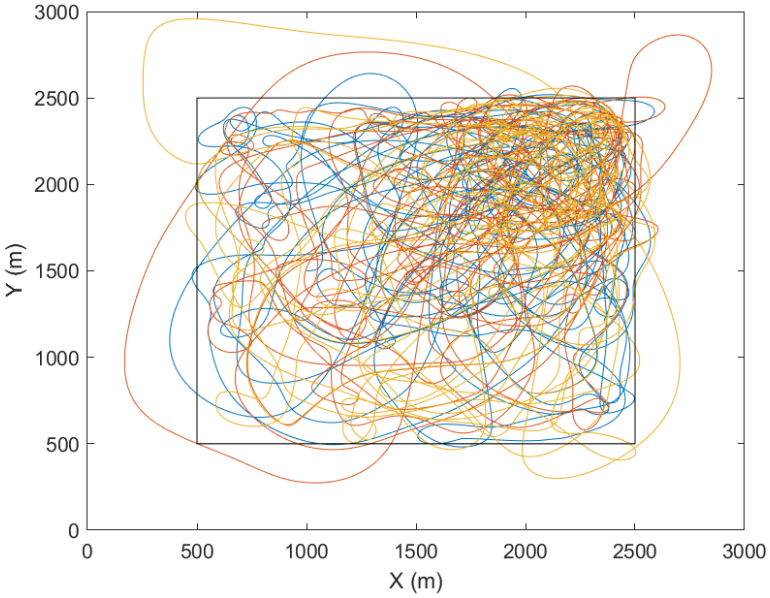


Figure 3.18: Obtained trajectory for 3 UAVs using the adapted dubin's path model.

Simulating three UAVs using this model produced the trajectories in Figure 3.18, with the different coloured lines depicting the different UAVs.

The trajectories still have an acceptable shape, without any sharp edges, unobtainable in the model. With the chosen parameters, the UAV stayed inside the extended domain and was never forced inside the domain with a non-optimal input.

The density of the lines also clearly shows the difference in surveillance time between the different areas on the domain, with the top right being the densest zone, corresponding to the area with the highest Fire Hazard Risk.

A small sample of the control input for one UAV over the course of 10 seconds is shown in Figure 3.19. The control input is smooth, even if the plot chosen was a *stairs* function instead of the regular *plot* to account for the discrete solver used.

Given that u_2 represents the heading turn rate, it is easy to see why the input would result in trajectories that are very easy to follow by a fixed-wing UAV. This values are small for the most part, with the transition being slow. It does not have the steep transitions from the previous model and so there should not be any agility imposition in the UAV, meaning a wider variety of aircrafts can be picked for usage.

The evolution of the metric for uniform coverage in this simulation, shown in Figure 3.20, exhibits an exponential behaviour in logarithmic scale, confirming that the system is working as expected. The result is, once again, not a monotonous function but decreases on average as the simulation time increases.

The metric once again shows cyclical behaviour after an initial transitory period, likely due to having covered the higher weighted Fourier bases, the lower frequency ones.

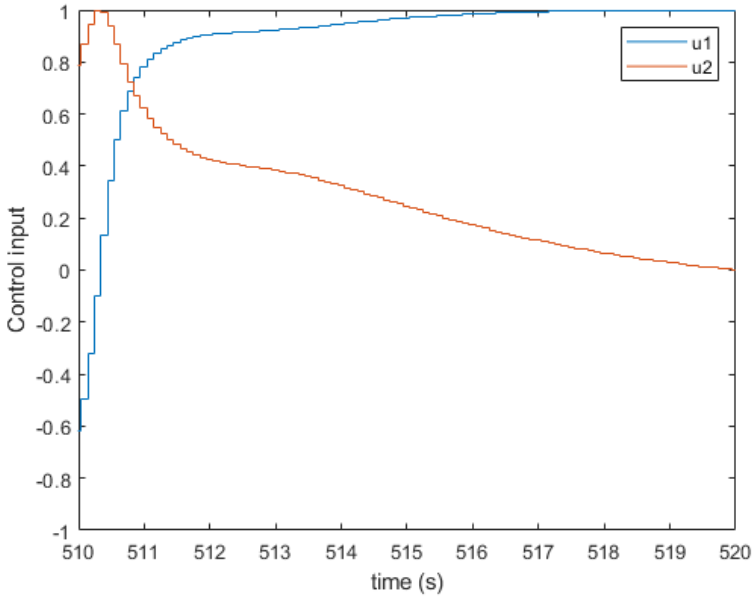
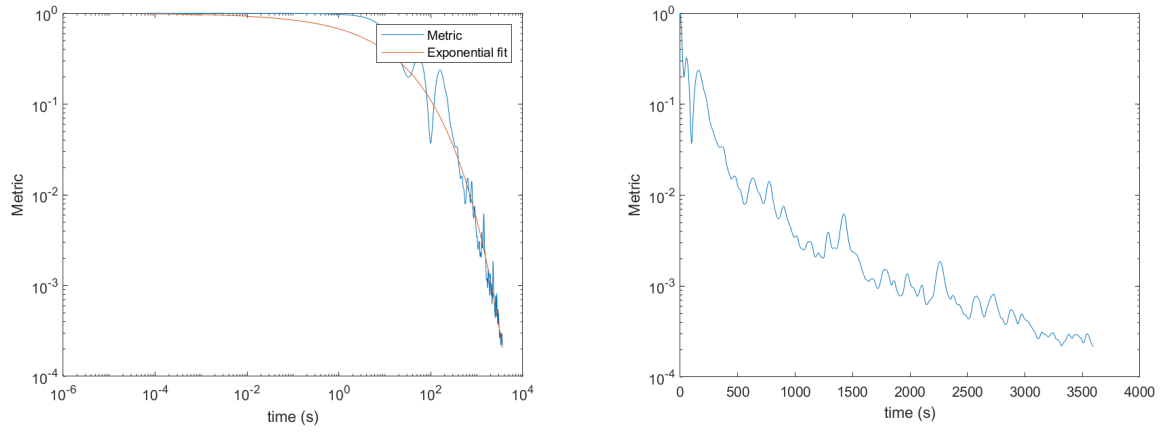


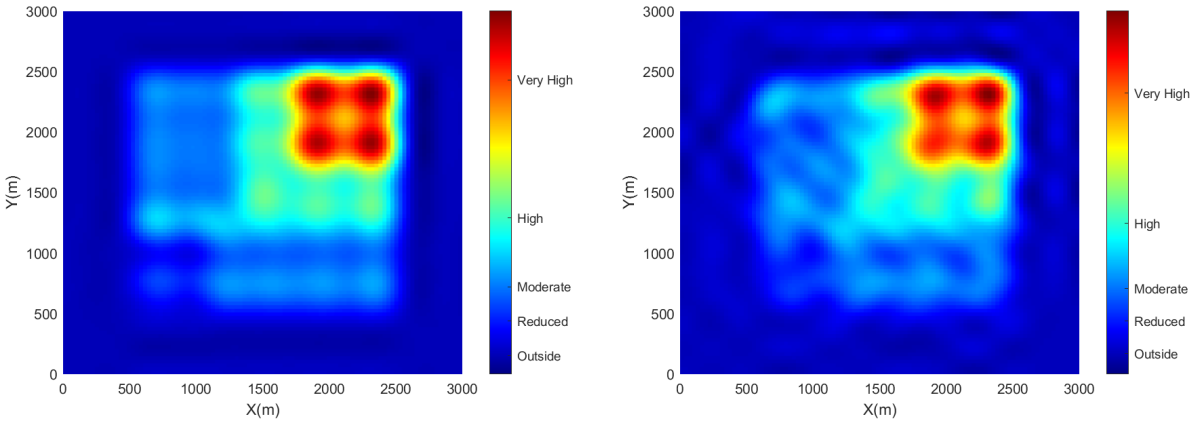
Figure 3.19: Control Input for one UAV in $t = [510, 520]s$. .



(a) metric for uniform coverage with both log scales.

(b) metric for uniform coverage with y log scale..

Figure 3.20: metric for uniform coverage representation for a one hour simulation with the Adapted Dubin's Vehicle Model.



(a) Heatmap of the desired probability density function

(b) Heatmap of the obtained probability density function.

Figure 3.21: Visualization of the desired probability density function's and the obtained probability density function's Heatmap on the Adapted Dubin's Vehicle Model.

The heatmap of the probability density function obtained at the end of one hour of simulation is shown in Figure 3.21. The heatmap shows that the *Very High* and *High* areas were appropriately covered, even if the edges of the *High* Risk zone don't exactly have the desired shape, at around $\mathbf{x} = [1500,\ 2000]^\top$.

The *Moderate* areas, on the other hand, are considerably messier, with the obtained distribution having plenty of darker blue spots, associated with a *Reduced* classification. The edges can clearly be noticed, but the content is not entirely uniform. This is a case where a slightly higher simulation time would be very beneficial, as the finer details only get worked on after the first basis converge to acceptable results due to the weighting.

The shape of the surveillance square domain is also very blurry, due to the vehicle spending a considerable amount of the beginning of the simulation doing a short sweep on the extended domain. This is due to the parameters of the system, and will be discussed next.

Overall, the obtained result is still well within the acceptable domain. Even if the shapes are not perfectly defined and uniform in detail, it is clear the higher risk areas are being surveilled appropriately and the rest would, with a higher running time, converge to the desirable uniformity.

3.4.1 Effect of parameter variation

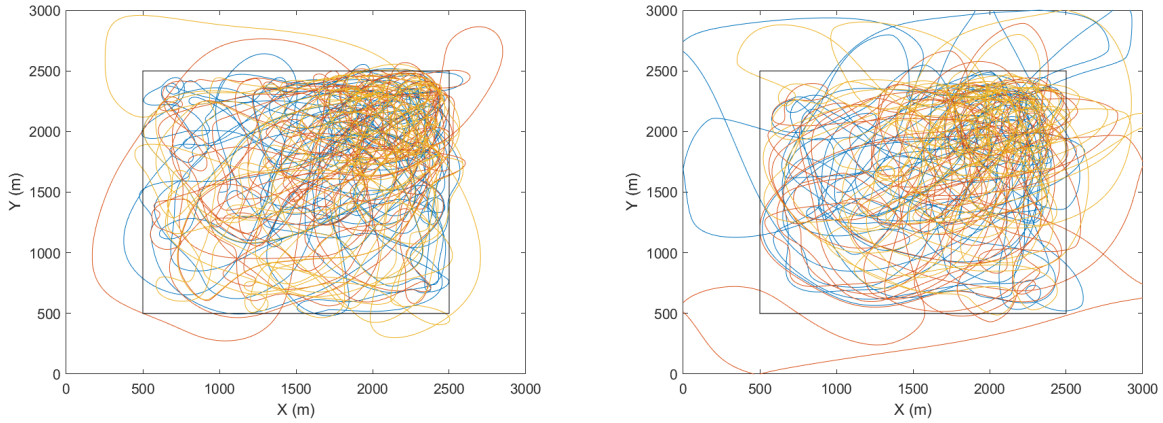
The system has one defining characteristic that makes it so multiple inputs are allowed, and that is the tracking of a point on the body frame and not the center of the body frame itself.

As a result, the position of the point directly influences the amount of control over the UAV. While the control of the linear speed differential is important, it is inevitably the behaviour of the heading turn rate input that makes the trajectory.

The control input component related to the heading turn rate is the second column of matrix $\mathbf{M}_2$, which is directly proportional to d_1, the x coordinate of the point in the body frame. As a result, the closer to the origin, the less effect the control law equation will have on the heading turn rate component due to the input vector being unitary. If the vector norm of the columns of $\mathbf{M}_2$ is very different, one component

will have a bigger weighting.

In conclusion, it is better if δv and $d_1 \cdot u_{max}$ are close, or at the very least in the same order of magnitude.



(a) Trajectory for $d_1 = 2m$.

(b) Trajectory for $d_1 = 0.5m$.

Figure 3.22: Effect of the tracked point's position in the trajectory of the Adapted Dubin's Path Model.

To exemplify the lack of control, the same simulation was run but for $d_1 = 0.5m$ instead of the previous $2m$. The resulting trajectory is shown in Figure 3.22, and it is clear that the behaviour is completely unacceptable.

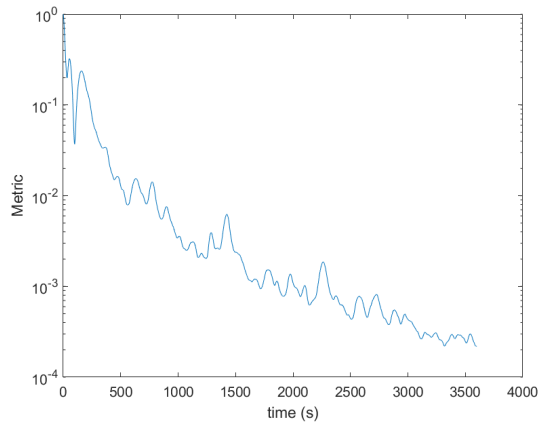
Before any further analysis, Figure 3.22 (b) shows the UAV going outside the extended domain, applying the second fail-safe measure, of forcing the aircraft back inside. This figure proves the designed algorithm is working properly.

Both simulations use the same parameters except the different distance, with $u_{max} = 0.5rad/s$ and $\delta v = 5m/s$. To expand on the previous concept of similarity in values, the product between the heading turn rate and the distance is, in the case of simulation (a) $d_1 \cdot u_{max} = 1$ and $d_1 \cdot u_{max} = 0.25$ for simulation (b). Comparing both of these values to the value of δv makes it clear why the parameter is not at all well chosen. In this case, for there to be a significant change in the control input related to the turning rate in (b), the value in the multiplying vector from the control law would need to be much higher than the value related to the velocity due to the normalization of the vector.

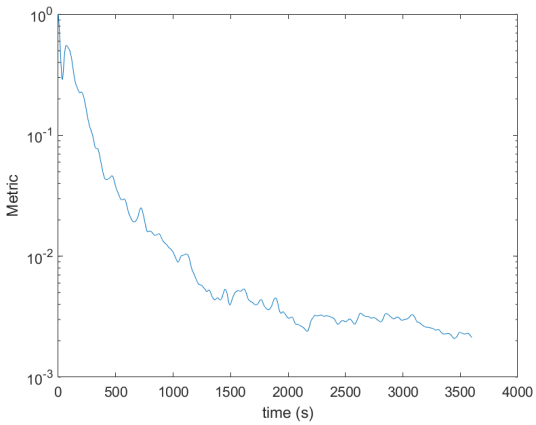
The result of the bad choice of parameter is clear from Figure 3.22 (b), in which the UAV has difficulty turning and is less agile. The effective consequence of bad parameterization is almost an attenuation in the turn rate, which explains the wider curves produced by the model.

The direct effect of a bad trajectory was translated to the metric for uniform coverage, shown in Figure 3.23. While the graphs look similar at first glance, the order of magnitude on the yy axis is of notice, as the value in (a) is almost an order of magnitude lower than that of figure (b).

This figure shows that, while the system still works for bad parameterised systems, the metric for uniform coverage would stabilize at a higher value. All the results shown up to this point suggest there is a point after which the metric for uniform coverage starts decaying at a much slower rate. The results from Figure 3.23 suggest that this point is influenced by the freedom the UAV is afforded on the model,



(a) metric for uniform coverage for $d_1 = 2m$.



(b) metric for uniform coverage for $d_1 = 0.5m$.

Figure 3.23: Effect of the tracked point's position in the metric for uniform coverage of the Adapted Dubin's Path Model.

something that was never previously showcased in this work. In this case, while the metric is still low, the obtained trajectory is not acceptable. In conclusion, a decrease in the metric for uniform coverage while necessary, is not sufficient to ascertain if the model is well parameterized or not.

While a bigger distance can mean better results, it can neither be so distant that it doesn't make sense to calculate the distribution on an unrealistic point nor should it overpower the velocity component, as the input would quickly depreciate into a graph similar to the Dubin's Path as u_1 would be reaching zero after normalising.

3.5 Model performance comparison

Since both the previous models had the same initial condition, same domain, and same risk function, it is possible to overlap the metric for uniform coverage graphs.

In Figure 3.24, the two metrics are overlapped and look very similar. The Dubin's Vehicle model has a slightly lower average value, but the results could have easily been passed as eachother's, as they have no noticeable differences. With no noticeable differences in performance, it is up to a matter of preference as to which model to use.

The Dubin's Path model is very simple to parameterise. Every parameter that the model has, the adapted model has as well. On the other hand, the control input is not smooth and can lead to issues with the aircraft and possibly bring difficulty in the control of the aircraft.

The Adapted Dubin's Path model has the flexibility of having more parameters and it is not out of the realm of possibility that there is some combination of values that would perform significantly better than the Dubin's Path mode, although none were found during the development of this work.

The biggest difference between the two models is then the fact that the control input in the Adapted Dubin's Model is smooth. This is exactly the issue that needed to be solved and the reason this model was applied, and the fact that there is no drop in performance is a positive aspect of the Adapted Dubin's Vehicle model.

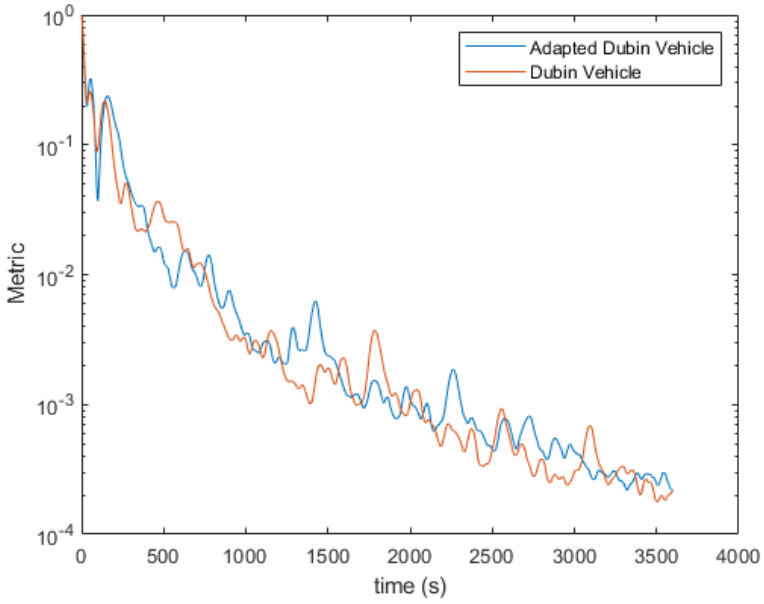


Figure 3.24: Comparison between the metric for uniform coverages of the Dubin's Vehicle and the Adapted Dubin's Vehicle.

Chapter 4

ROS/Gazebo Simulation

As a proof of concept regarding the capability for the real time application of the algorithm, a simulation was run using the Robot Operating System (ROS) paired with the physics simulator Gazebo 9.

In order to control the aircraft, the open-source software solution PX4 was chosen due to its ease of use and proven good results. The communication between the aircraft and the control station (the computer, in this case) was done via the ROS package MAVROS, a MAVLink interface using a Python wrapper.

The UAV in used for this simulation is a standard fixed-wing aircraft with VTOL capabilities. This is one of the example aircrafts with a pre-built physical model to be used with the open-source solution for flight control, PX4. The aircraft is shown in Figure 4.1.

A ROS simulation with Gazebo is particularly important because it's a highly reliable combination which means a successful simulation should lead to a positive real life experience after some amount of troubleshooting.

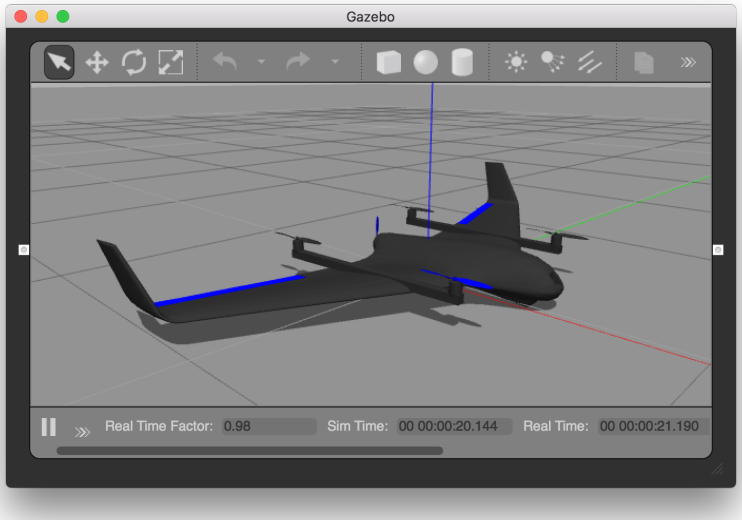


Figure 4.1: Simulated UAV in gazebo.

4.0.1 UAV Control Scheme

The development of an autopilot or inner loop controller is outside the scope of this work. As such, the standard PX4 controller modes were used.

Out of the box, PX4 comes with two relevant control modes: Mission Mode and Offboard mode. Mission mode implies the trajectory is known before hand, as it functions by pre-loading the aircraft with a list of way-points to achieve. Considering the ultimate goal of this work is real-time trajectory generation, this option was not particularly desirable.

On the other hand, the Offboard mode allows the ground control station to send references to the aircraft in real time, with the maximum time difference between messages being 0.5 seconds. This is the chosen mode for this project.

With Offboard flight mode, it is possible to control a fixed-wing UAV using either position waypoints or attitude references, including a thrust set-point [32]. These are the limitations without for the control using PX4 without writing a new flight mode, which would also be outside of the scope of this project.

While the attitude controller would make sense given the main input of the algorithm is the heading turn rate, the need for the thrust input would involve writing another layer of control to transform the linear speed into thrust references. This would be very difficult without knowing the physical model. As a result, the chosen approach was to control the aircraft sending waypoints in real time.

Another disadvantage of the PX4 controller is that there is no way to directly control the speed with fixed-wing UAV aircrafts outside of setting the trim air-speed as one of the MAVROS parameters. This is deterimental to the adapted Dubin's vehicle model, as even if waypoints are generated using its input, the UAV would not be able to follow them properly, either never reaching or reaching too soon and orbiting.

As a result, only the Dubin's Path Vehicle model was simulated using ROS. While this is not ideal, it is the simulation of the worst case scenario, a non-smooth control input and an off-the-shelf controller without further tuning.

4.1 Sensors and data usage

In order to calculate the control law for each step of time, the only thing the aircraft needs is the current time, the UAV location and its heading. For all of these measures, the only sensor needed is the GPS sensor.

The GPS sensor in the PX4 stack delivers its measurements filtered with various Kalman filters using data from various other sensors in the aircraft, and as such the results are very accurate. The sensor outputs both the UAV position and its Ground Speed, which will be used to calculate the heading. The heading angle is calculated with the following expression:

$$\psi = atan2(V_y, V_x)$$

where V_x and V_y are the linear speed components in the inertial frame.

In terms of memory required, at any given point in time, the control station only needs $(1+K)^2+1$ floating point numbers, which is slightly over 1kB of memory for the case of $K=15$ which produces very good results.

In order to obtain the waypoint to send to the UAV, a Python script using the MAVROS library for communication was used. The algorithm works in the following way:

1. Receive GPS data;
2. Calculate the distribution coefficient's integral using a trapezoidal integral function with step time equal to 1/10 of the receiving rate;
3. Calculate the optimal control input;
4. Calculate the next waypoint with the state equations using a trapezoidal integral function with step time equal to 1/10 of the receiving rate;
5. Send waypoint.

In order to calculate the next waypoint, the dynamic equation is simulated assuming perfect conditions and no delay in the response of the UAV.

The time between GPS messages by default in the PX4 simulation environment is unrealistically low, so to counteract this the algorithm ignores any message received until 0.1 seconds after the last one. This results in the algorithm sending waypoints at a frequency of 10 Hz. In this case, the waypoint sent assumes the aircraft will have a perfect movement in the next 0.1 seconds, and the calculated value is obtained using 10 steps of trapezoidal integrations.

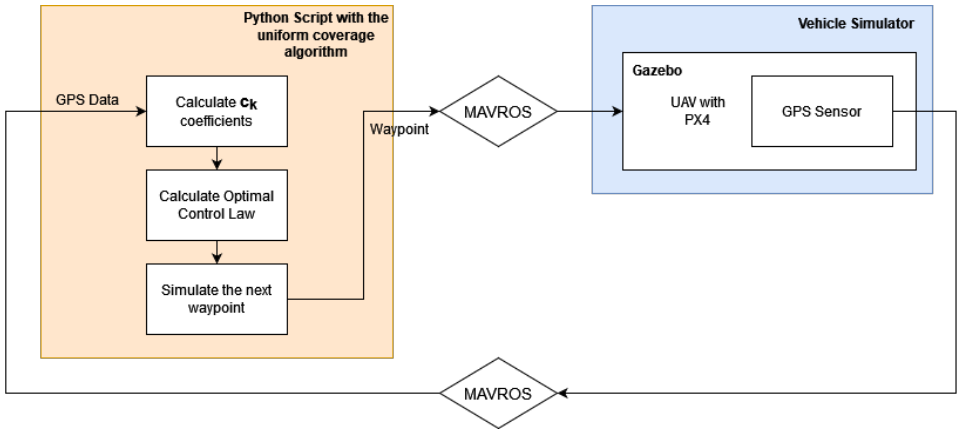


Figure 4.2: Graph summary of the ROS implementation architecture.

The designed architecture is summarised in Figure 4.2.

4.2 Results of the ROS simulation

For this simulation, using the Dubin's Vehicle Model, the aircraft speed was set to $25m/s$ instead of the previous simulation's $30m/s$ due to vehicle limitations. In terms of maximum heading turn rate, the script used the previous value of $u_{max}=0.5rad/s$ with positive results.

The simulation time was extended to around 6000 seconds from the previous 3600 seconds because of the decrease in speed and number of aircrafts.

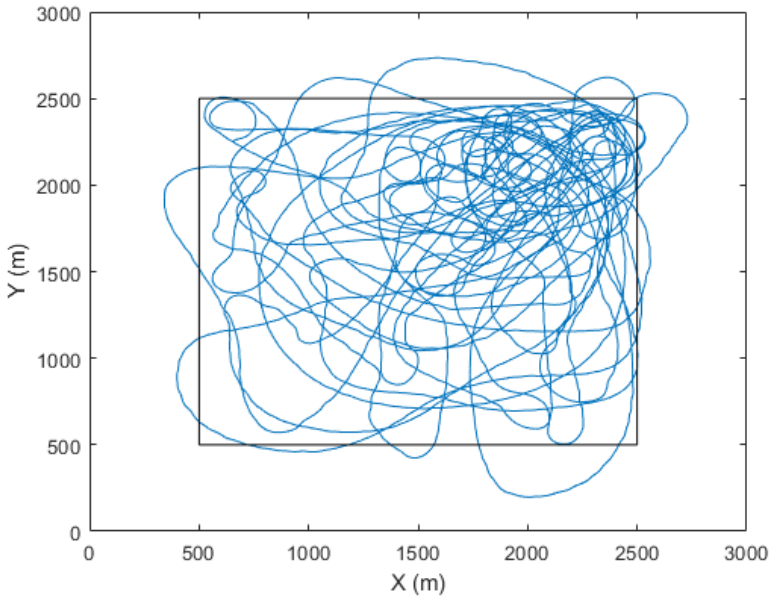


Figure 4.3: Trajectory obtained in 6000 seconds of gazebo/ROS simulation.

The trajectory obtained is present in figure 4.3. Looking closely, it is clear to see that the non-smoothness of the control input affected the trajectory quite a bit. None of the curves is completely smooth, with every turn being made up of very small turns. Unlike the Simulink simulations, using real values for the position and angle of the aicraft lead to the input not switching as fast, meaning there were longer periods of the same control input, which resulted in imperfections in the trajectory. The Simulink simulation didn't take into account the real UAV dynamics and as such it didn't matter that the input changed with a very high frequency, unlike in a real example.

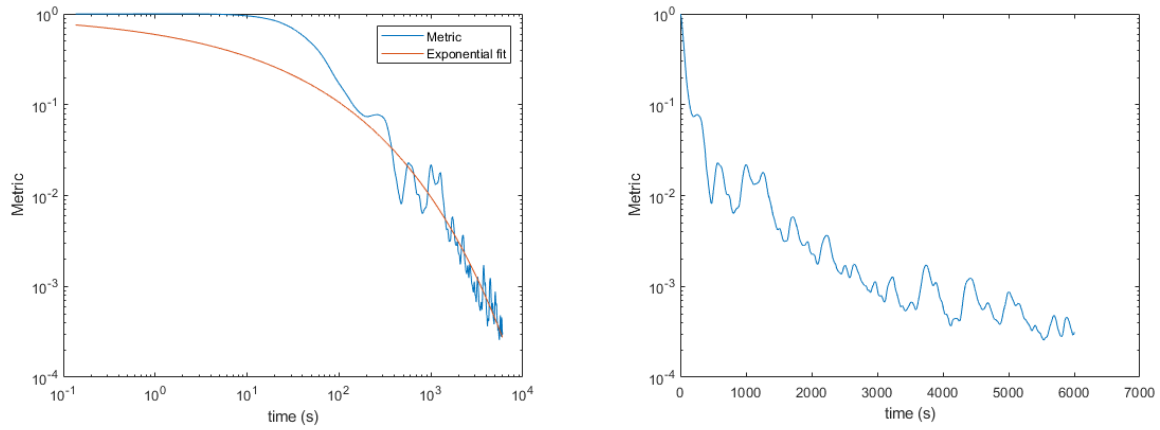
Another issue that caused imperfections in the trajectory is that changing the control input from $-u_{max}$ to u_{max} didn't result in the vehicle turning rate being changed to the exact value straight away.

In terms of the trajectory density, however, there is a clear differentiation in the different risk zones, with a clearly high density in the top right and very low density in the bottom left, the highest and lowest risk zones, respectively. This bodes well for the real life performance of the algorithm, as this was just one UAV.

The UAV spent very little time on the domain extension, and the fail-safe of forcing the aircraft back into the extended domain was never put into effect. This mean the UAV was always controlled with the optimal control input in (2.33).

The metric obtained, shown in Figure 4.4, shows a very similar behaviour to the ones obtained in the Simulink simulations, albeit with a smaller decrease rate to the a lower speed and number of aircrafts. An exponential fit was still possible in the logarithmic scale, proving the algorithm is working properly.

There is an unfortunate behaviour of the cyclical peaks being very high, and that is the result of the simulation having only one aircraft. In the Simulink simulations, the height of the peaks decreased at a

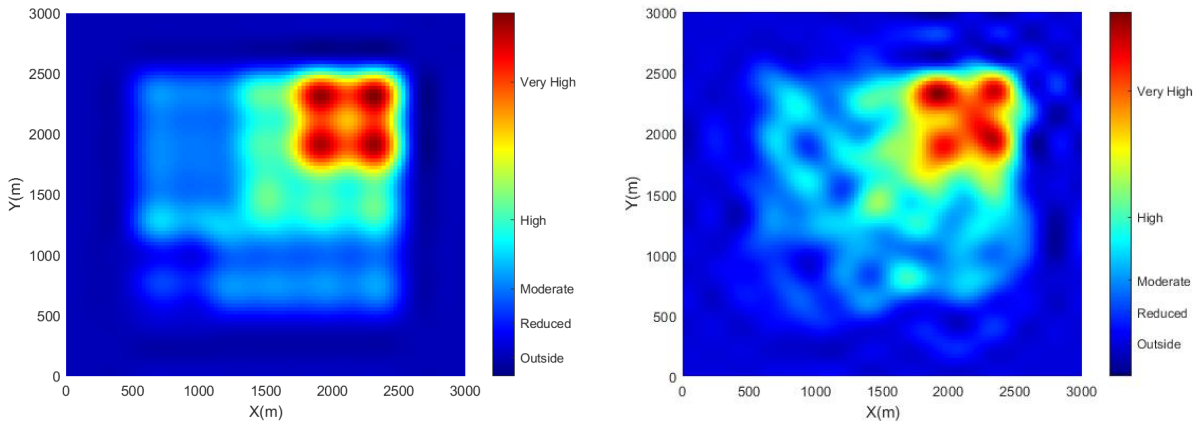


(a) Uniformity metric with both log scales.

(b) Uniformity metric with y log scale..

Figure 4.4: Uniformity metric representation for a 6000 seconds simulation in gazebo/ROS.

similar rate as the rest of the metric, which did not seem to be the case here. Nonetheless, the average of the metric still decreases with time, proving the real-time implementation to be successful.



(a) Heatmap of the desired probability distribution

(b) Heatmap of the obtained probability distribution.

Figure 4.5: Visualization of the desired probability distribution's and the obtained probability distribution's Heatmap on ROS/Gazebo simulation.

One area in which the implementation could have produced better results is the actual shape of the obtained probability distribution function. In Figure 4.5, it is very clear that the *Moderate* areas were not as surveilled as expected. This is to be expected due to the simulation being of a single UAV and not having an optimal response to the control input, but the results are still far from optimal.

The coverage of the *Very High* risk area is very good, with the square on the top right almost perfectly defined. The *High* risk area, while not as clearly defined, still shows an acceptable coverage.

While the cold spots in the *Moderate* area are not ideal, it is important to understand that this is calculated using the UAV's position, not the actual sensing range. With a sensor attached to the aircraft, the cold spots would also be monitored due to close proximity to the warmer spots in the close vicinity. This is a drawback of the way the probability distribution is calculated, because it does not take into account the actual area that is being surveilled, only the UAV's position.

Overall, this real time implementation of the Dubin's Path Vehicle can be said to be a success. Taken into consideration the conditions of the simulation, mainly the fact that the PX4 controller is not tuned to the specific problem or the fact that none of the algorithm's control inputs was directly passed into the PX4 controller, the results were in line with the ones obtained in the Simulink simulations.

The time between messages was also very small, giving breathing room for communication issues in a real implementation caused by possible channel latency and/or signal attenuation. The application does not required a very large bandwidth as only four float numbers are required from the aircraft at any given time, and the data used was already expected to be received in the ground station anyway, and thus does not bottleneck the channel unexpectedly, giving room for sending video data in near real time for further image processing for the actual fire detection.

Chapter 5

Early Wildfire Detection

5.1 Ignition Definition

In order to estimate of the probability of detection of an ignition during a surveillance mission, it is important to define what is to be considered an ignition.

For the purpose of this work, an ignition is to be the precursor to an uncontrollable fire, a small flame that might be contained easily and in a timely manner. With the most common fire causes mentioned in Section 1.1, this can be associated with a Crawling or Surface fire.

A Crawling or Surface fire is, as the name indicates, a fire caused by the burning of ground level vegetation and other fuels such as grass, trash debris, and pieces of wood from surrounding trees. Examples of these types of fire are shown in Figure 5.1

Figure 5.1: Two examples of surface fires in a grassland, (a) in Yellowstone National Park 1988 and (b) in the Cedar Creek Ecosystem Science Reserve. [33]

Surface fires burn at a temperature lower than crown fires, meaning fires in tree canopies, and spread slowly compared to the other forms of more intense fires. These burn with a surface temperature of around 400ºC to 500ºC, but sloped terrain and strong wind can speed up the spread of the fire [33].

Knowing the temperature at which a surface fire usually burns is extremely useful, as it can be used to estimate the power and the radiating power of a fire and thus select the most appropriate equipment to detect it.

Let an ignition then be defined as a surface fire burning at 500ºC with a small area. The area can be

changed according to further studies of the fire spread inside the covered zone, so as to determine how big or small a surface fire can be before it spreads and becomes uncontrollable. For the purpose of this work, an area of 5 squared meters was chosen.

Infrared (IR) sensors often function within certain wavelengths. As such, choosing between the different bands is crucial.

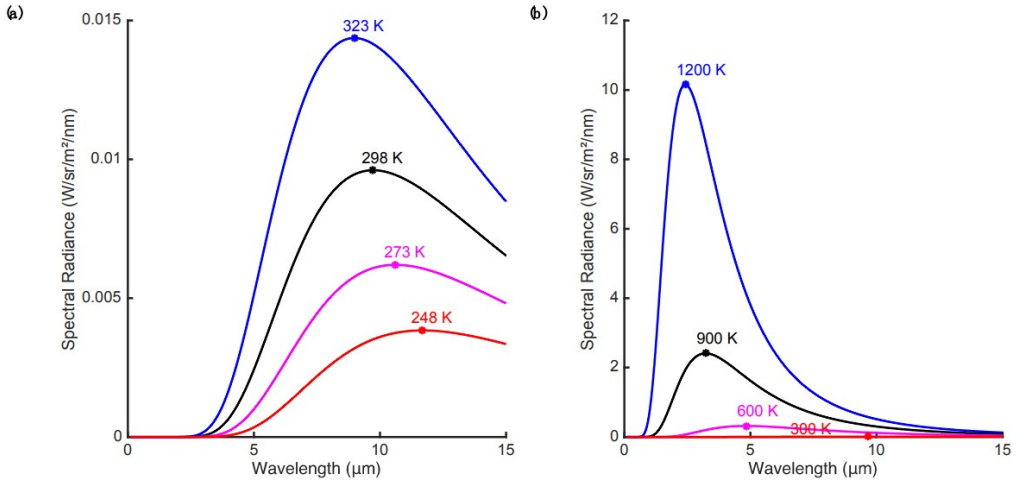


Figure 5.2: "Planck's law curves for ideal blackbody radiation at moderate temperatures (-25 to +50 ℃); (b) Higher temperature sources radiate much more intensely (note the large change in ordinate scale) and peak spectral response is shifted toward smaller wavelengths." [34]

All blackbody matter in thermal equilibrium emits electromagnetic energy following Planck's Law, which describes the Radiance of the energy as a function of the matter's absolute temperature for each wavelength [34]. The radiance can be seen as the portion of total power emitted at a certain wavelength.

In Figure 5.2, two graphs are plotted as a comparison between the peaks and wavelengths of low temperatures and temperatures associated with fire, from around $600\,K$ to $1400\,K$[33]. It can be seen that low temperature sources have their peak radiance at a wavelength of around $10\mu m$, while sources like fire have their peak radiance between 2 and $5\mu m$.

In [34], it is stated that experimental evidence supports the assumption that in the Thermal Infrared Range (TIR), between 0.75 and $15\mu m$, fire radiation can be modelled using the Planck's Law despite not being exactly a blackbody.

Using TIR sensors is extremely useful due to the fact that they only detect the radiation power of certain wavelengths and as such ignore most of what could limit visibility, like trees and thick smoke. In the case of fire detection, Medium Wavelength Infrared (MWIR) Sensors are exactly what is needed. MWIR sensors work in wavelengths between 3 and $8\mu m$ [34]. Incidentally, the energy radiated peaks inside that range, and the sensor will ignore lower temperature readings due to them being outside of the wavelength bounds imposed by the sensor.

5.1.1 Average power radiated by an ignition

The power emitted by a fire source is extremely complicated to estimate. The more complete models have multiple specific parameters and very complex equations. For the purpose of this work, the simplest model for the flame power will be used, as it is accurate enough to provide results that are not completely unrealistic. Consequently, the power radiated by a flame will be estimated with the Stefan–Boltzmann law, one of the simplest models which estimates the power radiated by a blackbody [35]. This power is a function of the source's temperature and body area, hence the importance of the values gathered previously.

The Stefan–Boltzmann law is given by the following expression,

$$P(T, A) = \sigma A T^4 \tag{5.1}$$

where $\sigma = 5.670374419 \times 10^{-8} W.m^{-2}.K^{-4}$ is the Stefan–Boltzmann constant, T is the body's absolute temperature and A is the body's radiating area.

With the temperature of $T = 752.15\,K\,(500\,C)$ and a surface area of $A = 5\,m^2$, an ignition radiates the following power in the surface, P_{0i}:

$$P_{0i} = 1.0131 \times 10^5\, Watts$$

5.2 Probability of Detection

Out of every measure already mentioned throughout this work, none is more important than the probability of detection of an ignition. A high probability of detection means the network is functioning as intended and the odds of having an uncontrollable catastrophe are reduced. It is then of extreme importance that said probability is maximized. For this reason, there needs to be a model for the probability of detection when employing the Aerial Sensing Network.

The simulations performed in Section 3 provide the average probability density function for the position of any of the UAVs in the Aerial Sensing Network. This is function C^t in (2.1). The function is constantly changing throughout the simulation, so it is important that an opportune moment is chosen to calculate the probability of detection over a mission. This is the last obtained value of the function, obtained using the Fourier Coefficients and the Fourier basis function in (2.4). An example of this would be the probability density function function in Figure 3.21 (b).

Let a UAV position (x_v, y_v) be a point in the coverage domain U. With the last moment of the simulation taken into consideration, the probability density of at least one UAV in the network being in (x_v, y_v) is $P_v(x_v, y_v)$, defined as

$$P_v(x_v, y_v) = C^{t=3600}. \tag{5.2}$$

The probability density function of an ignition is the target distribution function of the algorithm, μ. As

such, let (x_i, y_i) be a point in the coverage domain U. The probability of an ignition happening in (x_i, y_i) is $P_i(x_i, y_i)$, defined as

$$P_i(x_i, y_i) = \mu(x_i, y_i). \tag{5.3}$$

Lastly, the probability density function of detection of an ignition in (x_i, y_i) by a UAV in (x_v, y_v) needs to be defined. This is the sensor model.

5.2.1 Sensor Model

The model for the probability of detection depends entirely on the sensor used. However, it was not feasible to obtain any sort of sensor for this work to model said probability. Every sensor is different and the modern sensors also have different performance depending on the software used to process the images obtained.

The objective is to provide a general workflow to test and parameterise the UAV fleet before the full ASN is deployed. Consequently, a general sensor model is used as an example to generate data. In a real work environment, this is the step that would require further fine tuning in the form of obtaining the sensing effectiveness of the sensor used in conjunction with the image processing software.

The model used for this part was proposed in [36] and takes advantage of the fact that electromagnetic radiation propagates spherically in a vacuum and in a free medium such as air. The result of this is that the power dissipation is proportional to the inverse of the distance travelled squared. As such, the noisy measurement obtained by the IR sensor can be modelled by the following equation,

$$P_r(x_v, y_v, x_i, y_i) = \frac{P_{0i}}{4\pi R(x_v, y_v, x_i, y_i)^2} + n_i \, Watts \tag{5.4}$$

where the parameter n_i is a random variable following a Gaussian distribution and $R(x_v, y_v, x_i, y_i)$ is the distance between the ignition point and the sensor position, which is the vehicle position, given by

$$R(x_v, y_v, x_i, y_i) = ||(x_v, y_v, h) - (x_i, y_i, 0)|| \tag{5.5}$$

where h is the flight altitude of the vehicle, a constant parameter over the surveillance mission.

The proposed model for the probability of detection is based on whether the power received by the sensor is higher than a certain threshold. Consequently, let $P_{sens}(x_v, y_v, x_i, y_i)$ be the probability of the sensor at (x_v, y_v) detecting an ignition at (x_i, y_i) be given by the the following expression

$$P_{sens}(x_v, y_v, x_i, y_i) = \{P_r(x_v, y_v, x_i, y_i) > c\}. \tag{5.6}$$

The important variables to determine for this model are then the standard deviation of the white noise added to the received power and the threshold at which a detection is deemed to have failed.

These parameters would change considerably for each different sensor and software combination used. However, for the purpose of this work some values were estimated in order to showcase the

paremeterisation workflow.

The threshold c is the first to be decided. To obtain this value, a distance from the ignition at which the sensor has a 50% chance of detection needed to be chosen. The various aircrafts compiled in Table 3.2 show that missions at 5000 to 6500 meters above sea level are possible, so it is assumed that the sensors would work at that range.

The value chosen for the 50% detection mark was then $5000m$. This value could easily be changed to accommodate a specific sensor. The threshold for the used sensor model is then

$$c = \frac{P_{0i}}{4\pi 5000^2} = 3.2247 \times 10^{-4}\, Watts. \tag{5.7}$$

The standard deviation, however, is a much more complex number to estimate. It directly affects the shape of the probability function. It can be interpreted as a combination of sensor noise, sensor resolution, software effectiveness and even further power losses due to an imperfect medium with the presence of smoke and foliage.

The cumulative normal distribution has a sigmoid function shape, which means the sensor probability function does as well. The magnitude of the standard deviation could not be much lower than 10^{-4}, the magnitude of the threshold, as it would turn the probability function would have a slope so steep it would almost be a step function, 1 before 5000m and 0 after 5000m. The value chosen for the standard deviation is $\sigma = 5 \times 10^{-5}$ and makes it so there is a decrease of detection probability from 0.9 to 0.1 from around 4500m to 5500m, a much more realistic behaviour. This value is of course easily adjustable by testing the IR sensor in use.

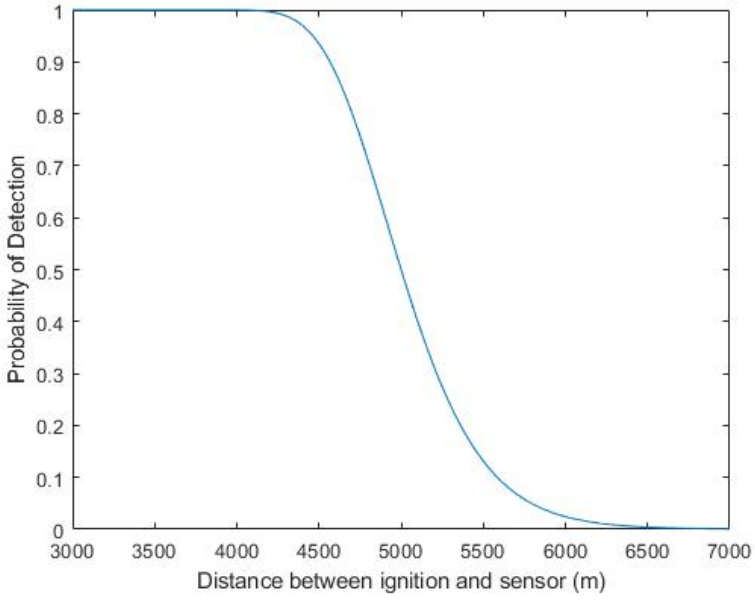


Figure 5.3: Probability of the sensor detecting an ignition at a distance of R meters from the sensor.

In Figure 5.3 the probability of detection of an ignition is plotted in function of the distance to the sensor, with a threshold c and additive noise $n_i \rightarrow \mathcal{N}(0, \sigma^2)$.

Not only are the sensors limited by distance, they are also limited by their field of view. To take in

consideration the FOV of the sensor, it will be assumed that the sensor is always pointing straight down from the aircraft and has a conic field of view. The radius of the cone on the ground can then be written as follows,

$$r = h \arctan(\frac{\gamma}{2}),$$

where h is the flight altitude and γ is the angle of the cone of view. This angle can range between different values even in the same sensor, but as it increases the resolution of the obtained image also decreases due to more area being covered with the same amount of pixels. The value used throughout this section is $\gamma = 24º$, a value that sits in the middle of the range in sensors like [37].

The value of γ can be changed depending on the circumstances. In a more complete model, P_{sens} would have a component that depended on this angle and it could be used as a parameter for the optimization. However, with the lack of a physical sensor for testing, this was not achieved.

Ultimately, the sensor model to be used it P_s, defined as follows,

$$P_s(x_i, y_i, x_v, y_v) = \begin{cases} P_{sens}(x_i, y_i, x_v, y_v) & \text{if } ||(x_i, y_i) - (x_v, y_v)|| \leq r \\ 0 & \text{otherwise} \end{cases}. \tag{5.8}$$

5.2.2 Probability function

With a well defined sensor model, vehicle position probability density function and ignition probability density function, the overall probability of one UAV in the network detecting an ignition can be the integral of the product of those three over all possible vehicle and ignition positions,

$$P_d = \iint_U \iint_U P_s(x_v, y_v, x_i, y_i) P_i(x_i, y_i) P_v(x_v, y_v) dx_v dy_v dx_i dy_i. \tag{5.9}$$

The detection of an ignition is an independent event for each aircraft in the network, and as such, for the whole network, the important measure is the probability of at least one aircraft detecting the ignition. That is the complimentary probability of none of the UAVs detecting the ignition. Consequently, for a network of N UAVs, the joint probability of detection is given as

$$P_{joint} = 1 - \prod_{i=1}^{N} (1 - P_d). \tag{5.10}$$

5.3 Parametric Study

In order to apply the probability functions from equations 5.9 and 5.10, the Simulink simulation for the Adapted Dubin's Vehicle model was used. The data extracted from the simulation were the final Fourier coefficients of the agent average position probability function in order to build the function from (5.2).

The parametric study from this section has the end goal of aiding the user in choosing the right number of agents for the surveillance network. With an approximated probability of detection, the overall cost of the network, including buying the UAV, their surveillance software and maintenance can be

balanced out by the expected economical damage that a fire would cause. It would then be up to the user to decide how many UAVs to buy and at which altitude the surveillance mission would take place at.

5.3.1 Influence of the flight altitude

The altitude of the agents is not evident just by analyzing the equations. While a decrease in altitude would mean the sensor is more likely to detect an ignition, it also means that it is covering less area at once due to the FOV restrictions.

To clarify the effect of the flight altitude, the probability density function for the average position of the UAVs (P_v) was the one shown in Figure 3.21, for a network of 3 agents, and the probability density function for the ignition location (P_i) the one shown in Figure 3.4.

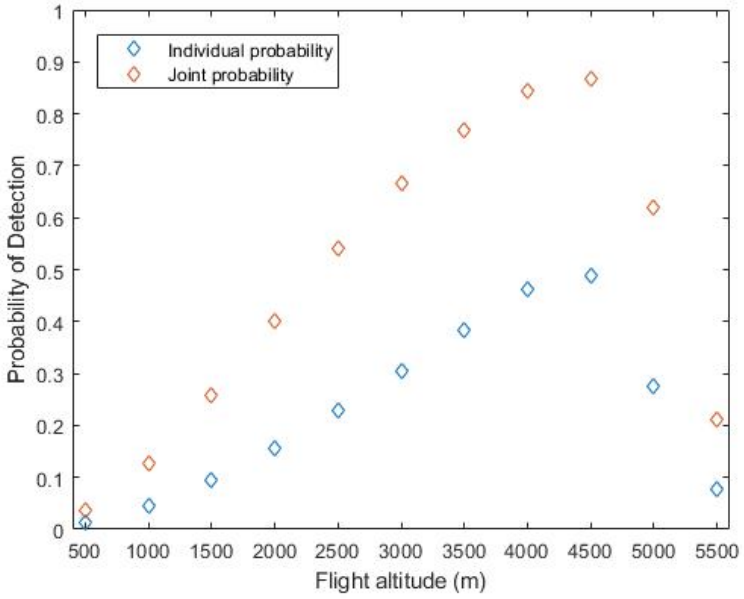


Figure 5.4: Probability of detecting an ignition in function of the altitude.

In Figure 5.4 the results for flight altitudes from 500m to 5500m are shown. The probability of detection shows a clear peak at an altitude of 4500m.

The most important conclusion to take from this graph is that the sensor model is a limiting factor in the detection probability due to both range limitations and FOV restrictions. Considering these restrictions, there is an optimal flight altitude, in this case being somewhere close to $4500m$.

The increase in the probability of detection until the $4500m$ altitude mark is explained by the FOV restrictions. At those altitude, the distance between the sensor and the ignition would almost always be inside the region where the probability of detection is near 1 in Figure 5.3. Increasing the altitude in those ranges increases the area covered by the agents at any given time, thus increasing the probability of detection.

There is a sharp decrease in the probability of detection at the 4500m mark, which is the distance at which the sensor probability of detection is around 90%. By design of the model, after that distance,

the sensor performance starts decreasing rapidly, with a detection probability of around 10% at 5500m. This is important because the flight altitude is the minimum distance at which the sensor would be able to sense the ignition, and it would happen when the aircraft would be positioned directly upwards of the ignition. Consequently, the maximum flight altitude is directly affected by the sensor used, in this case anything above 4500m makes little sense.

5.3.2 Influence of the number of agents

The number of agents in the ASN has an effect not only in the joint probability of detection (cf. (5.10)) but also on the final probability density function for the average position of each agent (P_v), a direct result of the data shown in Figure 3.13. One of the conclusions taken from that last Figure was that increasing the number of agents would take the obtained probability density function function of the vehicle position closer and closer to the target function, which is the probability density function function of the ignition location (P_i). Consequently, it is unexpected and improbable that increasing the number of agents decreases the individual probability of detection. However, this test is ideal to test the effect of small deviations from the target distribution in the context of the probability of detection.

To obtain the different P_v functions, as it change slightly with the number of UAVs in the simulation, the same simulation used to obtain the probability density function function shown in Figure 3.21, using the Adapted Dubin's Vehicle mode, was run but changing the number of UAVs in the simulation in a range of $N = 1$ to $N = 6$.

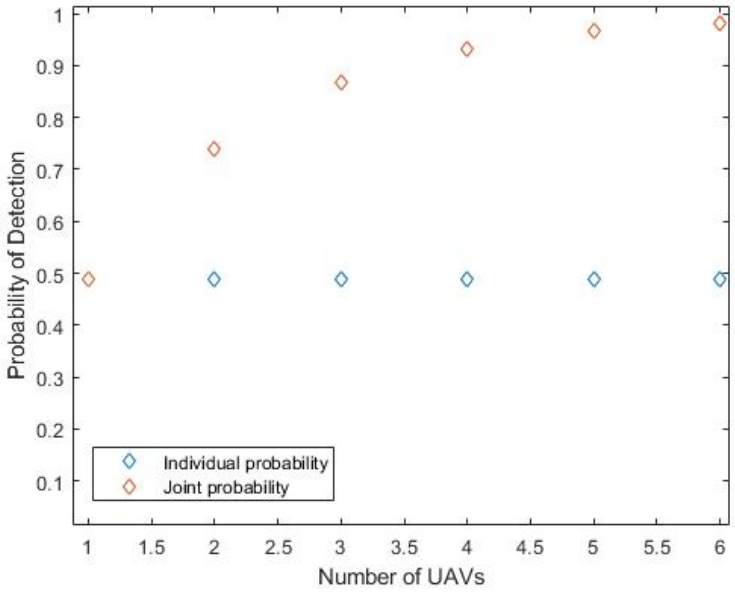


Figure 5.5: Probability of detecting an ignition in function of the number of agents.

The results obtained are shown in Figure 5.5, with the probability of detection of the ASN at a flight altitude of $h = 4500m$. The values for the individual probability of detection stayed almost the same for every simulation, with deviations smaller than 0.5% in magnitude. This is explained by the results in 3.13. The metric for uniform coverage tended to decrease faster the more agents in the network but the

end result was still in the same order of magnitude, so the performance was, in practice, very similar. In a probabilistic model where the most limiting factor is the sensor performance, the small difference in performance has little to no impact.

Even if the minute changes in the P_v function between the simulations do not affect the individual probability of detection, the joint probability of detection increases with the number of agents, but can quickly reach the saturation point of 1 due to the nature of equation 5.10.

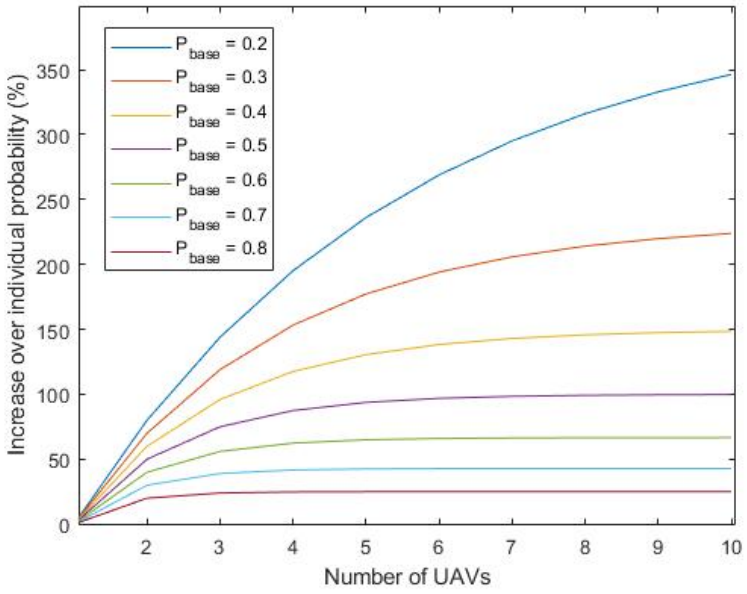


Figure 5.6: Percentage increase over the individual probability value with the number of UAVs, for a variety of base probabilities.

How quickly the value of the joint probability reaches the saturation depends on the value of the individual probability of detection. In Figure 5.6, various values for a base individual probability of detection were utilized to showcase the increase over the base value that occurs when adding more UAVs. This Figure is unrelated to Figure 5.5 and is only to serve as a showcase to the difference in the effectiveness of adding another agent for different probability values.

The effectiveness of adding more agents increases as the probability of detection by each agent decreases. For example, adding a third UAV to the network if the probability of detection of the individual UAV is 80% would barely add value to the ASN. On the other hand, if each UAV has an individual probability of detection of 20%, adding the another four increases the value of the joint probability detection by almost 250% over the base value of 20%, to a joint probability of detection of 70%.

5.3.3 Influence of the domain area

The size of the domain is something that has not yet been studied in this work. That is because the scalability of the uniform coverage algorithm when the domain scale is concerned was never in question. The equations for the optimal control law never depended on the domain size. In fact, most of the values used in the calculations were normalized.

To introduce domain scaling, a Domain Scaling Factor scalar value is used. This value is used to scale the domain covered as a proportional factor that is multiplies the risk function in Figure 3.4. For example, for the base domain of $2000 \times 2000m$, a factor of 4 means the area to be covered is then $8000 \times 8000m$ with a risk function with the exact same shape as the one in Figure 3.4.

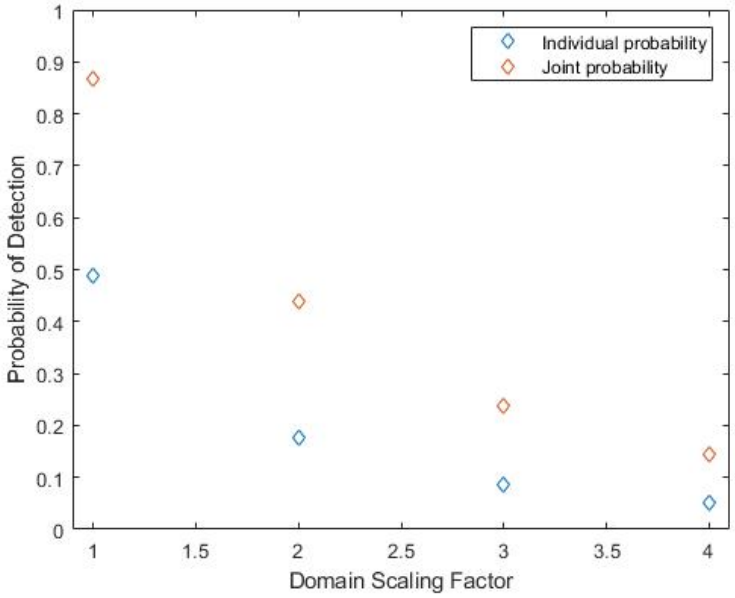


Figure 5.7: Probability of detecting an ignition in function of the domain size.

The results of varying the scaling factor between 1 and 4 are shown in Figure 5.7. The flight altitude considered is $h = 4500m$, the optimal altitude from Figure 5.4 and for an ASN of 3 UAVs.

The decrease in the probability of detection as the domain size increases is very sharp. It is important to note that because the scaling factor scales the width and length of the domain, the are increase is actually the scaling factor squared. For the domain factor of 2, the area to be covered is $2^2 = 4$ times as big. The shape of the individual probability of detection is reminiscent of half a parabola, which makes sense considering the area factor is the domain scaling factor squared. In conclusion, while not an exact relation, doubling the area covered results in about half the individual probability of detection.

In order to preserve a high joint probability of detection as the domain scale goes up, an increasing number of agents is needed to properly cover the area. For example, for the scaling factor of 4, 20 agents would be needed to obtain a joint probability of detection of 64.75%.

5.3.4 Parametric study conclusions

There are two main conclusions to take from the parametric study:

- The resource allocation problem was solved using the uniform coverage algorithm;
- The area coverage problem is entirely limited by the sensing method.

The first conclusion is proven by the results in Figure 5.5, which combined with the results from Section 3, show that the resources were allocated to higher risk zones independently of the number of

agents in the ASN. In fact, the addition of more agents did not change their own individual detection probability in any considerable way because for any of the tests, the probability density function for the agent positions obtained by the uniform coverage algorithm was already very close to the desired distribution. The bottle-neck of (5.9) for the individual probability of detection is the sensor function P_s.

For the most part, the most important graph is the one in Figure 5.4 which, when looked at in tandem with the rest of this section, can be the defining factor for the final parameterisation of the ASN. The peak probability of detection is right before the heavy decay from the sensor model, especially the threshold of 50% chance of detecting the ignition at a distance of $5000m$. Getting a better sensor, both in terms of software and hardware, can increase this threshold, which is extremely valuable as the threshold is the definitive bottle-neck of the probabilistic model. Before the peak at $h = 4500m$, the probability of detection was increasing almost linearly, so it stands to reason that should the threshold be increased, the linear increase would keep happening until the new threshold value took effect.

With a higher individual probability of detection, less UAVs need to be deployed and larger domains can be covered for a cheaper price. From the data in Figure 5.6, the increase in performance of the ASN by increasing the amount of agents can be obtained and balanced against the cost of the new agent. Given the nature of the risk distributions being associated with economical damage, each percentage point in the joint probability of detection can be associated with a monetary value. Adding another UAV becomes non-profitable when the performance increase of adding it results in a percentage increase in detection that has an inferior monetary value compared to the cost of the UAV and its maintenance.

Chapter 6

Conclusion and future work

6.1 Conclusion

Overall, the objectives outlined in Section 1.3 were all properly accomplished throughout this work. In Chapter 2, the theoretical basis for the algorithm was established, with all the functions defined for understanding how to say if the trajectories of the system can be considered uniform. In light of the movement restrictions of a fixed-wing UAV, two different movement models were introduced in Sections 2.3.1 and 2.3.2 that are easily associated with the coordinated turn movement of an aircraft. The model in Section 2.3.1 has a scalar control input while the model in Section 2.3.2 has a two dimensional control input. By the end of Chapter 2, the optimal feedback control laws for both models were obtained.

In Chapter 3, the Fire Hazard Risk function was defined in order to run Matlab/Simulink simulations for both models. In this Chapter, the issue of the agents being able to leave the domain due to the symmetry in the Fourier Transform were taken into consideration and solved by extending the Fourier Transform domain and forcing the agent back inside should it leave.

The results of the simulation for the Dubin's Vechicle Model in Section 3.3 show that the metric for uniform coverage converges to nearly zero, and the heatmaps of the probability density function function of the UAV position closely resemble the desired heatmap. There is an issue due to the one dimensional nature of the control input, which can be solved by a regularization at the cost of some performance.

The results of the simulation for the Adapted Dubin's Vehicle Model in Section 3.4 show that the performance is equivalent to the performance in of the Dubin's Vehicle Model before regularization without having issues in the control input. However, the model is very sensitive to bad parameterisation, which can cause the model to not function properly if the parameters are not picked carefuly.

The real time component of the trajectory generation proved to be possible to implement within the limitations of a real life environment by running an unoptimised ROS simulation using the Gazebo physics simulator and a PX4 vehicle model for simulation in Section 4. The vehicle could keep up with the trajectory waypoints sent at a rate of near 10Hz in simulation and it is assumed that barring some communication issues due to bandwith and signal latency, this could work just as well in a real scenario.

In Section 5, the concept of ignition was defined in order for a probabilistic model for the ignition

detection probability to be proposed. A sensor model for a Infra-Red sensor was proposed and is the basis for the probabilistic model.

Lastly, the workflow to decide how to size the sensing network was proposed, studying the effect of different parameters such as flight altitude and the number of UAVs. It was concluded that the sensor model was the bottleneck of the probability of detection, namely the conic angle that defined the FOV and the distance from the ignition at which the sensor starts losing performance. Additionally, due to FOV restrictions, there is an optimal altitude for the ASN that maximizes the joint probability of detection.

6.2 Future work

During the development of this work, many subjects were not developed to their full potential, and further research would increase the effectiveness of a network using the ergodicity based algorithm.

6.2.1 Integrating the sensor model into the optimal control law

As concluded in Section 5.3.4, the sensor is one of the biggest bottlenecks of the sensing network. In order to decrease the effect of this bottleneck, it might be possible to increase the sensing effectiveness if the optimal control law doesn't measure just the UAV's trajectory in the time domain but instead the area which the UAV senses, integrating the sensor model in order to obtain an equivalent sensing area at ground level. This could be done by modifying equation 2.5, for example.

6.2.2 Adding the flight altitude to the optimal control law

The current model basically functions in a 2D plane. This reduces the movement possibilities by one whole dimension. For example, it could be possible that the optimal solution in a 3D environment would be to decrease the flight altitude near zones of higher risk and spend extra time there while increasing the altitude in lower risk zones in order to cover more terrain at once and spend less overall time in lower risk zones.

Developing this idea would most most likely come hand in hand with the integration of the sensor model, as it contains all the relevant information in regards to the effect of the flight altitude.

6.2.3 Developing an inner control loop for UAV control using the direct model outputs

The ROS implementation during this work lacked the ability to control the agent using the Adapted Dubin's Vehicle model. Additionally, the Dubin's Vehicle model needed to be controlled simulating way-points instead of using the direct output from the model, or rather the optimal control law. This is far from ideal.

To avoid as many non-optimal layers as possible, the development of a controller that is able to take the direct control inputs of the model is of extreme importance. This would reduce the need to use an

integration model for waypoint simulation which can introduce errors especially when smaller time steps are considered due to floating point precision.

6.2.4 Optimizing the target vehicle position distribution based on the probability of detection instead of the risk probability

Equation 5.9 for the probability of detection of an ignition by one agent shows potential for an increase in the detection probability. Over the course of this work, it was assumed the optimal solution for the vehicle's position distribution would be that is follows the risk distribution. However, it might be possible that giving even more importance to higher risk zones than the risk itself provides better results.

Obtaining an optimal solution to an equation as complex as equation 5.9 is very complicated due to the nature of the 4 Dimension integral and would be enough for a standalone work due to the variety of possible optimization methods. One example would be optimization using a genetic algorithm with the first guess being the risk function itself.

The optimization would prove to be even more difficult due to the few constraints the risk function has, mainly the fact that there is no need for it to be a smooth analytical function over the entirety of the domain.

6.2.5 Staggering the agents over multiple altitudes and obtaining a different position distribution for each agent

The further optimization to the previous methods is that not every single agent needs to have the same target position distribution. In fact, staggering the various agents over different altitudes could prove to be a better solution. For example, an agent could be responsible for the coverage of the lower risk areas using a higher altitude while other agents could be covering the domain at a lower altitude and completely disregarding the lower risk zones in order to maximize the probability of detection. The addition of this step would make the optimization of the target vehicle position distribution even more complex and time consuming, but could prove to increase the detection probability considerably using fewer resources.

6.2.6 Implementing a risk assessment algorithm based on surveillance data

One advantage of having a real time sensing network is that human activity can be easily detected, and the risk distribution of a fire hazard can be dynamically changed based on the current human behaviour, allocating more resources for covering areas where humans are currently in. This was not discussed during this work, but the model is only effective if the risk function is trustworthy, as it is the base upon which the vehicle probability density function is chosen from. All the probabilistic models developed throughout this work are useless if the conditions of the covered area are constantly changing and the changes are not reflected on the fire hazard risk function.

Bibliography

[1] D. de Gestão de Áreas Públicas e de Protecção Florestal. 10º relatório provisório de incêndios florestais, 2017.

[2] L. Pinto. Pedrógão grande: Governo estima prejuízos de 500 milhões de euros, 2017. URL https://www.publico.pt/2017/07/03/sociedade/noticia/pedrogao-grande-governo-estima-prejuizos-de-500-milhoes-de-euros-1777751. Last Visited: 10/14/2021.

[3] C. A. San-Miguel-Ayanz J. Forest fires at a glance: Facts, figures and trends in the eu. *European Forest Institute.*, pages 11–18, 2009.

[4] D. de Gestão de Áreas Públicas e de Protecção Florestal. 6º relatório provisório de incêndios florestais, 2021.

[5] A. C. Meira Castro, A. Nunes, A. Sousa, and L. Lourenço. Mapping the causes of forest fires in portugal by clustering analysis. *Geosciences*, 10(2), 2020. doi: 10.3390/geosciences10020053.

[6] B. E and A. C. *Introduction to Satellite Remote Sensing*. Elsevier, 2017. ISBN 9780128092545.

[7] Y. J. Kaufman, C. O. Justice, L. P. Flynn, J. D. Kendall, E. M. Prins, L. Giglio, D. E. Ward, W. P. Menzel, and A. W. Setzer. Potential global fire monitoring from eos-modis. *Journal of Geophysical Research: Atmospheres*, 103(D24):32215–32238, 1998. doi: https://doi.org/10.1029/98JD01644.

[8] A. Alkhatib. A review on forest fire detection techniques. *International Journal of Distributed Sensor Networks*, 2014, 03 2013. doi: 10.1155/2014/597368.

[9] G. Avellar, G. Pereira, L. Pimenta, and P. Iscold. Multi-uav routing for area coverage and remote sensing with minimum time. *Sensors*, 15:27783–27803, 11 2015. doi: 10.3390/s151127783.

[10] D. Thomas, D. Butry, S. Gilbert, D. Webb, and J. Fung. The costs and losses of wildfires: a literature survey. Technical report, nov 2017.

[11] R. F. B. F. M. F. Catry, F. X. Modeling and mapping wildfire ignition risk in portugal. *International Journal of Wildland Fire.*, 18:921–931, 2009. https://doi.org/10.1071/WF07123.

[12] C. E. Van Wagner. Development and structure of the canadian forest fire weather index system. forestry technical report 35. Technical report, National Capital Region, 1987.

[13] IPMA. Cálculo do Índice de risco de incêndio rural risco conjuntural e meteorológico – rcm. Technical report, Instituto Português do Mar e da Atmosfera, I.P., 2020.

[14] J. C. Verde and J. L. Zêzere. Assessment and validation of wildfire susceptibility and hazard in portugal. *Natural Hazards and Earth System Sciences*, 10(3):485–497, 2010. doi: 10.5194/nhess-10-485-2010.

[15] L. Lourenço and M. Mira. Grandes incêndios florestais de 17 de junho de 2017 em portugal e exemplos da determinação das respetivas causas. *Territorium*, 26(II):49–60, 2019. doi: https://doi.org/10.14195/1647-7723_26-2_4.

[16] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2013.09.004.

[17] A. Xu, C. Viriyasuthee, and I. Rekleitis. Optimal complete terrain coverage using an unmanned aerial vehicle. In *2011 IEEE International Conference on Robotics and Automation*, pages 2513–2519, 2011. doi: 10.1109/ICRA.2011.5979707.

[18] G. Mathew and I. Mezi. Metrics for ergodicity and design of ergodic dynamics for multi-agent systems. *Physica D: Nonlinear Phenomena*, 240(4-5):432–442, Feb. 2011. 10.1016/j.physd.2010.10.010.

[19] N. Michael, E. Stump, and K. Mohta. Persistent surveillance with a team of mavs. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2708–2714, 2011. doi: 10.1109/IROS.2011.6095174.

[20] P. Tokekar, A. Budhiraja, and V. Kumar. Algorithms for visibility-based monitoring with robot teams. 12 2016.

[21] K. E. Petersen. *Ergodic theory*. Cambridge University Press, 1989.

[22] G. Mathew, I. Mezi, and L. Petzold. A multiscale measure for mixing. *Physica D: Nonlinear Phenomena*, 211(1-2):23–46, Nov. 2005. 10.1016/j.physd.2005.07.017.

[23] R. W. Beard and T. W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012. ISBN 0691149216.

[24] R. W. B. J. R. T. Lawton and B. J. Young. A decentralized approach to formation maneuvers. *IEEE Transactions on Robotics*, 19(6):933–941, 2003. doi: 10.1109/TRA.2003.819598.

[25] D. E. Kirk. *Optimal control theory: an introduction*. Dover Publications, 2004.

[26] Px4 controller, . URL `https://docs.px4.io/v1.12/en/`. Last Visited: 10/14/2021.

[27] Albatross. URL `https://www.unmannedsystemstechnology.com/company/applied-aeronautics/`. Last Visited: 10/14/2021.

[28] Helvetis – jabali. URL https://helvetis.com/vtol-uav-isr/. Last Visited: 10/14/2021.

[29] Pd-2 uas vtol. URL https://www.unmannedsystemstechnology.com/company/ukrspecsystems/pd-2-uas-vtol-fixed-wing-uas/. Last Visited: 10/14/2021.

[30] Stream c vtol. URL https://threod.com/products/stream-c-vtol/. Last Visited: 10/14/2021.

[31] Bramor ppx. URL https://www.c-astral.com/en/unmanned-systems/bramor-ppx. Last Visited: 10/14/2021.

[32] Px4 offboard mode, . URL https://docs.px4.io/master/en/flight_modes/offboard.html. Last Visited: 10/14/2021.

[33] W. J. B. S. J. P. M. E. A. Andrew C. Scott, David M. J. S. Bowman. *Fire on Earth: An Introduction*. Wiley-Blackwell, 2014.

[34] R. S. Allison, J. M. Johnston, G. Craig, and S. Jennings. Airborne optical and thermal remote sensing for wildfire detection and monitoring. *Sensors*, 16(8), 2016. doi: 10.3390/s16081310.

[35] J.-L. Rossi, K. Chetehouna, A. Collin, B. Moretti, and J.-H. Balbi. Simplified flame models and prediction of the thermal radiation emitted by a flame front in an outdoor fire. *Combustion Science and Technology*, 182(10):1457–1477, 2010. doi: 10.1080/00102202.2010.489914.

[36] W. Li and C. Cassandras. Distributed cooperative coverage control of sensor networks. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2542–2547, 2005. doi: 10.1109/CDC.2005.1582545.

[37] Flir boson. URL https://www.flir.eu/support/products/boson/#Documents. Last Visited: 10/14/2021.