

Article

Computationally-Efficient Distributed Algorithms of Navigation of Teams of Autonomous UAVs for 3D Coverage and Flocking

Taha Elmokadem and Andrey V. Savkin * 

School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney 2052, Australia; t.elmokadem@unsw.edu.au

* Correspondence: a.savkin@unsw.edu.au

Abstract: This paper proposes novel distributed control methods to address coverage and flocking problems in three-dimensional (3D) environments using multiple unmanned aerial vehicles (UAVs). Two classes of coverage problems are considered in this work, namely barrier and sweep problems. Additionally, the approach is also applied to general 3D flocking problems for advanced swarm behavior. The proposed control strategies adopt a region-based control approach based on Voronoi partitions to ensure collision-free self-deployment and coordinated movement of all vehicles within a 3D region. It provides robustness for the multi-vehicle system against vehicles' failure. It is also computationally-efficient to ensure scalability, and it handles obstacle avoidance on a higher level to avoid conflicts in control with the inter-vehicle collision avoidance objective. The problem formulation is rather general considering mobile robots navigating in 3D spaces, which makes the proposed approach applicable to different UAV types and autonomous underwater vehicles (AUVs). However, implementation details have also been shown considering quadrotor-type UAVs for an example application in precision agriculture. Validation of the proposed methods have been performed using several simulations considering different simulation platforms such as MATLAB and Gazebo. Software-in-the-loop simulations were carried out to assess the real-time computational performance of the methods showing the actual implementation with quadrotors using C++ and the Robot Operating System (ROS) framework. Good results were obtained validating the performance of the suggested methods for coverage and flocking scenarios in 3D using systems with different sizes up to 100 vehicles. Some scenarios considering obstacle avoidance and robustness against vehicles' failure were also used.



Citation: Elmokadem, T.; Savkin, A. V. Computationally-Efficient Distributed Algorithms of Navigation of Teams of Autonomous UAVs for 3D Coverage and Flocking. *Drones* **2021**, *5*, 124. <https://doi.org/10.3390/drones5040124>

Academic Editor: Abdessattar Abdelkefi

Received: 22 September 2021

Accepted: 21 October 2021

Published: 25 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: cooperative control; coverage control; flocking control; Multi-Agent Systems; Navigation of Aerial Drones; UAVs; UAV Control Systems; UAV Swarm

1. Introduction

In recent years, there has been an increasing interest in mobile wireless sensor networks (MWSNs), where a number of networked autonomous vehicles can be deployed in different environments to achieve sensing tasks. Advances in communication made MWSNs more appealing where vehicles (sensors) can share information to perform cooperative monitoring, sensing, detection and exploration.

The problem of controlling multi-vehicle systems to achieve a global objective is referred to as cooperative control. There exist different subset problems related to this area depending on the common goal to be achieved by the vehicles. Coverage control deals with problems where a certain region of interest needs to be surveyed whether using a single vehicle or multiple (i.e., using MWSNs). Tackling these problems with multi-vehicle systems has given rise to new challenges to traditional cooperative control in the field of coverage control [1–8]. Unmanned aerial vehicles (UAVs) have become a popular choice in this area to form MWSNs especially in places inaccessible by ground vehicles. In general, multi-UAV systems have been emerging in various applications such

as precision agriculture [9–13], aerial manipulation and transportation [14–18], search and rescue [14,19,20], firefighting [21], surveillance and monitoring [22,23], mapping and exploration [24–26], etc.

Coverage control problems can be classified as either static or dynamic. Another classification is based on [27] where coverage problems are categorized into *Blanket coverage*, *Barrier coverage* and *Sweeping coverage*, which are defined as follows:

- Blanket coverage is forming a static arrangement to maximize the detection rate of events through an area of interest.
- Barrier coverage is a static formation over some region (i.e., a barrier) to minimize intrusions or maximizing detection of objects going through it.
- Sweeping coverage is the formation of dynamic arrangements moving across a region of interest for maximal detection/exploration along the whole region.

Clearly, blanket and barrier coverage problems belong to the static class while sweeping is a dynamic coverage problem.

Static coverage can be applied in different applications such as surveillance, real-time monitoring of crops, live stock and pollution, intruders detection in the security domain, etc. In these scenarios, the combined sensing field of view of all sensors is sufficient to monitor the regions of interest with a static configuration that the MWSN converges to. On the other hand, dynamic coverage is based on applications where the sensors need to remain mobile in order to survey the regions of interest. Examples of these applications include [28], but not limited to, autonomous search & rescue, monitoring of dynamically-changing environments, real-time monitoring of large areas with limited number of vehicles, visual inspection using multiple vehicles with limited combined field-of-view (FOV), firefighting in forests using a group of vehicles with limited spray area, autonomous irrigation with limited spray area, etc.

Flocking is another class of problems where cooperative control is applied, which is a subset of formation control, to deal with movement of large number of vehicles (swarms). In general, controlling a large number of vehicles to achieve a common motion objective is motivated by collective behavior of animals such as bird flocks and fish schools. In this case, vehicles can make motion decisions by following a set of rules contributing towards the collective behavior (i.e., flocking). These rules are: flock centering (cohesion), collision avoidance (separation) and velocity matching (alignment) according to Reynold's model [29,30].

The main contribution of this work is to develop novel distributed control strategies addressing cooperative control problems to achieve 3D coverage and flocking behavior. Specifically, the proposed methods address the 3D barrier and sweep coverage problems motivated by some of the ideas in [1,6,31]. In a 3D environment, a barrier can be defined as a static arrangement of sensors with overlapping sensing zones [32] forming a surface or a 3D region. The suggested control strategies rely on estimated centroidal Voronoi configurations over virtual regions generated by the sensors locations in a distributed manner depending only on shared information from neighbor vehicles. Furthermore, the suggested methods are also extended to address flocking problems in 3D spaces where the movement of the 3D virtual region is decided collectively by the group to achieve the common motion objective.

The designed control laws require relative distances with neighbor vehicles to be shared over communication channels which makes the overall problem related to Networked Control Systems (NCSs) [33–39]. The current work assumes that such information is available to the control system. However, several challenges related to communication channels imperfections needs to be considered when evaluating the performance of the overall networked control system. Thus, it is important to design a reliable communication system to allow for advanced swarming behaviors of robotic systems [40].

Overall, the vehicles' collective motion becomes constrained within a specific dynamic region under the application of the suggested control methods similar to region-based shape control methods [41]. The dynamic region can be selected as a collection of non-

overlapping polygons in 3D. Furthermore, one can control the dynamics of the dynamic region to generate 3D sweeping behavior which is the key idea used in the developed 3D sweeping coverage strategy. This is also considered to handle obstacle avoidance where vehicles can collaboratively control the dynamics of the virtual barrier and even apply deformations to its shape in real-time, which is then communicated through the networked multi-vehicle system. Furthermore, bounded control laws are proposed, which is important in practice to satisfy limits on the vehicles' velocities and accelerations. The main advantages of the suggested approaches can be highlighted as follows:

- collision avoidance among vehicles and connectivity is ensured by the adopted Voronoi-based approach
- the approach is highly scalable and robust against vehicles' failure
- obstacle avoidance can be managed in a decomposed and distributed manner

A general 3D kinematic model is adopted in the design which is applicable to different UAV types and autonomous underwater vehicles (AUVs). A 6 degrees-of-freedom (DOF) dynamical model for quadrotors is further considered to show a possible way of implementation with low-level control design. Several simulations were carried out to validate the performance of the suggested methods in addition to showing its scalability and robustness. Moreover, software-in-the-loop (SITL) simulations were also performed using the Gazebo robotic simulator based on the quadrotor full dynamical model to evaluate the computational complexity of the implemented algorithms with particular interest in applications related to precision agriculture.

The organization of this paper is as follows. A summary of related works is given in Section 2 highlighting some of the drawbacks that motivated the proposed design. Section 3 introduces some essential concepts related to graph theory, locational optimization and Voronoi Partitions which are used in our control strategy, and the tackled 3D coverage problems are defined in Section 4. After that, distributed barrier and sweeping coverage control strategies are proposed in Section 5 considering a general 3D kinematic model. These approaches are validated through several simulation cases in Section 6. Furthermore, a generalization of the suggested methods is presented in Section 7 to address 3D flocking problems along with validation. Further implementation details considering quadrotors dynamics with low-level control design are presented in Section 8, which is evaluated using software-in-the-loop simulations. Finally, this work is concluded in Section 9 with a suggestion for a potential direction of future work.

2. Related Work

Coverage problems and optimal sensors deployment for both fixed and mobile wireless sensor networks have gained great interest over the years. According to [42], some of the common techniques used to address static coverage problems are resource-aware [43–45], search space-based [46,47], potential-based [48,49], Voronoi partition-based [1,5,31,50–55] and angle view [56–58].

An example algorithm of static deployment of cameras (i.e., non-mobile sensors) to achieve coverage with optimal resource allocation was proposed in [44]. In [43], modified Lloyd-like algorithms were developed to address the 2D static deployment of mobile sensors while considering optimizing the overall power consumption. The tradeoff between coverage performance and energy consumption of mobile heterogeneous wireless sensor networks was also investigated in [45].

Optimal placement of fixed and mobile networked sensors have been tackled in some works using optimization-based techniques. For example, the work in [46] adopted a particle swarm optimization-based approach to address the static deployment of cameras over 2D and 3D monitoring spaces for optimal visual monitoring. On the other hand, 2D static deployment of mobile wireless sensors was addressed in [47] using the Multi-objective Immune Algorithm (MIA). The problem formulation targeted the maximization of the coverage area while minimizing energy consumption due to movement. The authors have also considered the connectivity perseverance through limited movement within

specified communication ranges. However, collision avoidance between mobile sensors and obstacles avoidance were not considered.

Another class of approaches addressing coverage problems of MWSNs is based on artificial potential field, which makes it easier to include collision avoidance rules as potential functions. In this case, control laws are designed based on gradient descent algorithms. For example, a decentralized approach was developed in [48] addressing the problem of deploying mobile sensors with limited sensing capabilities to achieve the best possible coverage over an n -dimensional Euclidean space. The mobility of sensors was modeled as single integrator. The developed control law relies on forcing the inter-agent distances to reach a desired distance while avoiding collisions with other vehicles. Obstacle avoidance was not considered in this work. On the other hand, another method was presented in [49] using artificial potential fields considering both forces between the mobile sensors and obstacles in the environment to achieve both static deployment and collision avoidance. The method was validated considering only 2D environments.

A similar class of methods existing in the literature is based on Voronoi diagrams. The formulation of such methods relies also on gradient descent algorithms adopting potential functions that can encode optimal coverage in a distributed manner, which was well developed in [1,31,59] with rigorous mathematical proofs. In these works, Lloyd's algorithm was adopted to develop distributed control methods for motion coordination of MWSNs to achieve optimal coverage. Only 2D coverage problems were considered, and obstacle avoidance was not considered. The approaches were distributed in nature where each vehicle can compute its own Voronoi cell based on information shared from its Voronoi neighbours. Similarly, an adaptive 2D approach was proposed in [5] based on Voronoi diagrams. However, the authors of this work suggested a decentralized control law design, where each vehicle can approximate the centroid of its Voronoi cell based only on its sensory measurements rather than relying on information received from its Voronoi neighbors. This can be more practical in cases where Voronoi neighbors are at distances larger than the vehicle's communication range. This approach was also considered in [50]; however, a different weighting function, which defines the sensing performance over a planar region, was considered.

Many of the above approaches consider planar circular sensing models. Some works have considered different sensing models such as [51,52]. In these works, non-uniform unequally scaled 2D sensing footprints were considered. Additionally, Voronoi partitioning was performed only with respect to the sensed space, rather than the whole region of interest as in the previous methods.

Some research works have also considered additional requirements in the coverage problem formulation such as graph connectivity perseverance and information routing. For example, the 2D static coverage problem tackled in [53] considered that the mobile sensors are required to send the collected sensory information to a set of destinations over the network while performing the coverage task. Their solution was based on a constrained optimization problem considering Voronoi partitioning as part of the solution.

The above methods are mainly developed to address static coverage problems of MWSNs, where the sensors need to converge to a static arrangement with optimal coverage. There also exist methods addressing dynamic coverage problems such as [28,60–66]. In [28], a multi-agent system was considered with a 2D nonholonomic agent model. The approach relied on a leader-follower paradigm, where the vehicles try to maintain desired distances from a position targeted by the leader. This can be more challenging in 3D and for systems with large number of vehicles. Another approaches considering 2D nonholonomic agents were developed in [61,62] proposing control laws based on a special form of Lyapunov-like functions to achieve satisfactory coverage level and collision avoidance.

Motivated by the Voronoi-based static coverage methods, the work in [63] suggested a planar approach based on modified Voronoi partitions with time-varying density functions, which describe the coverage performance, to address the planar dynamic coverage problem. However, obstacle avoidance was not considered in this work. A similar approach was also

proposed in [63] where the authors suggested a Bayesian prediction method to estimate the unknown time-varying density function.

Overall, many of the existing static and dynamic coverage control approaches consider only two-dimensional sensing fields, and the literature lacks a proper analysis of sensor networks deployed in three-dimensional (3D) sensing fields [67]. Even those methods proposed for multi-UAV and multi-AUV systems assume that the vehicles are moving at a fixed altitude/depth with a planar sensing footprint without utilizing the full capabilities of such vehicles. It is hence more motivating to work towards addressing 3D coverage problems exploiting the rich geometric properties of 3D MWSNs [67]. Some efforts have been made in that area such as [32,67–73]. Thus, one of the main motivations of this work is to contribute towards the state-of-the-art 3D static and coverage problems using MWSNs.

On the other hand, the flocking problem has also attracted great interest. Many researchers have tackled the flocking control problem where the global group objective is to move the whole group towards some goal region or to follow a certain motion pattern (for example, see [29,74–83]). Most of state-of-the-art flocking control methods rely completely on artificial potential field to handle local interactions between vehicles, which have some disadvantages such as being prone to getting stuck at a local minimum especially due to conflicts between forces required to achieve moving towards goal, collision avoidance and obstacle avoidance control objectives. To address these limitations, the suggested control strategy separates the control components for these objectives on different layers as will be shown later.

3. Preliminaries

The proposed methods in paper relies on concepts from graph theory, locational optimization and Voronoi partitions. A summary of these concepts is provided in this section based on [1,29,31,59]. Note that when considering a multi-UAV system as a mobile wireless sensor network, UAVs may interchangeably referred to throughout the paper as sensors, nodes, agents or vehicles.

3.1. Graph Theory

A multi-UAV sensor network consisting of n UAVs can generally be characterized using a set of nodes/vertices $\mathcal{U} = \{1, 2, \dots, n\}$ and a set of edges (paired vertices) $\mathcal{E} \subseteq \{(i, j) : i, j \in \mathcal{U}, j \neq i\}$. Note that i and j will be used in the rest of the paper to refer to the index of a node/vertex within a set (i.e., index of a UAV within the group). Each vertex corresponds to a single UAV/sensor, and edges represent interaction between UAVs which are within communication or detection range from each others. The overall network topology is then described using a graph $G = (\mathcal{U}, \mathcal{E})$ which can be *directed* or *undirected*. In an *undirected graph*, an edge exists from vertex i to vertex j if and only if an edge exists from j to i (i.e., $(i, j) \in \mathcal{E} \leftrightarrow (j, i) \in \mathcal{E}$). Otherwise, the graph is called *directed*. Generally, homogeneous multi-UAV systems can be described using undirected graphs since all UAVs have same communication and sensing capabilities. Moreover, a *path* between two vertices i_0 and i_k is defined as a sequence of vertices $\{i_0, i_1, \dots, i_k\} \subset \mathcal{U}$ where an edge exists between each subsequent vertices in the sequence such that $(i_l, i_{l+1}) \in \mathcal{E}, \forall l \in \{0, 1, \dots, k-1\}$. If every pair of vertices in \mathcal{U} is connected by a path, the graph $G(\mathcal{U}, \mathcal{E})$ is then called *connected*. Clearly, a crucial part for MWSNs is to maintain network connectivity all the time.

Furthermore, define a *neighborhood* \mathcal{N}_i around a vertex i as the set of all vertices which have edges with vertex i such that:

$$\mathcal{N}_i = \{j \in \mathcal{U} \setminus \{i\} : (i, j) \in \mathcal{E}\}. \quad (1)$$

For a homogeneous system, let $r > 0$ denote the communication range for all UAVs. Hence, all UAVs within a spherical region of radius r around UAV i belong to its neighborhood such that

$$\mathcal{N}_i = \{j \in \mathcal{U} \setminus \{i\} : \|\mathbf{p}_i - \mathbf{p}_j\| \leq r\}, \quad (2)$$

where $\mathbf{p}_i \in \mathbb{R}^3$ is the position of UAV i , and $\|\cdot\|$ is the Euclidean norm in \mathbb{R}^3 . Similarly, all norms in the rest of the paper are Euclidean norms in \mathbb{R}^3 .

3.2. Locational Optimization

Deployment of mobile sensors in an environment to achieve optimal sensor coverage is regarded as a multicenter problem from locational optimization (i.e., spatial resource-allocation problem). A brief description about some of the facts related to this class of problems is summarized next based on [1,59].

Consider a bounded region of interest \mathcal{Q} , including its interior, defined in a space \mathbb{R}^d with dimension d . A *partition* of \mathcal{Q} consists of a group of n non-overlapping polytopes $\mathcal{W} = \{W_1, \dots, W_n\}$ such that $W_1 \cup \dots \cup W_n = \mathcal{Q}$. Furthermore, let $\phi : \mathcal{Q} \rightarrow \mathbb{R}_+$, ($\mathbb{R}_+ = \{a \in \mathbb{R} : a > 0\}$) be defined as a *distribution density function* representing a measure of information or the likelihood of an event to take place over \mathcal{Q} . The *sensing performance* of a sensor located at some position \mathbf{p}_i as seen from any point $\mathbf{q} \in \mathcal{Q}$ depends mostly on the distance $\|\mathbf{q} - \mathbf{p}_i\|$. Clearly, as this distance increases, the sensing performance degrades. Hence, one can describe the sensing performance at location \mathbf{q} of the sensor \mathbf{p}_i using a non-increasing piecewise continuously differentiable function $f(\|\mathbf{q} - \mathbf{p}_i\|) : \mathbb{R}_+ \rightarrow \mathbb{R}$. Thus, the larger the value of f , the better the sensing performance at \mathbf{q} is.

Using the above definitions, one can define a multicenter cost function characterizing the average coverage provided by a set of n sensors at $\mathbf{p}_1, \dots, \mathbf{p}_n$ over an point in \mathcal{Q} as follows:

$$\mathcal{H}(\mathbf{p}_1, \dots, \mathbf{p}_n) = \int_{\mathcal{Q}} \max_{i \in \{1, \dots, n\}} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q}. \quad (3)$$

The above function provides a measure of the sensing performance expected value provided by all sensors at any point $\mathbf{q} \in \mathcal{Q}$ [31]. Now, in order to find the optimal placement for all sensors, an optimization problem needs to be solved to maximize the value of $\mathcal{H}(\mathbf{p}_1, \dots, \mathbf{p}_n)$.

Remark 1. Note that there are slightly different definitions for f in the references [1,31,59] where it can be either considered as a representation of sensing degradation or sensing performance over \mathcal{Q} (as considered here). This does not affect the overall analysis done here except that the considered optimization problem will either be minimization (of sensing degradation) or maximization (of sensing performance).

3.3. Voronoi Partitions (Tessellation)

This subsection highlights some key points about Voronoi partitions needed for our problem formulation. A *Voronoi partition/tessellation* is the subdivision of a space into a number of regions generated by a set of points (see Figure 1 for a 2D example). Consider that we have n sensors located at fixed locations $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\} \in \mathcal{Q}^n$. A voronoi partition of \mathcal{Q} consists of a set of disjoint Voronoi regions/cells $\mathcal{V}(\mathbf{P}) = \{V_1, \dots, V_n\}$ generated by these sensors where

$$V_i = \{\mathbf{q} \in \mathcal{Q} : \|\mathbf{q} - \mathbf{p}_i\| \leq \|\mathbf{q} - \mathbf{p}_j\|, \forall j \neq i\}, \quad (4)$$

and $V_1 \cup V_2 \cup \dots \cup V_n = \mathcal{Q}$.

It has been established that this Voronoi partition is the optimal partition of \mathcal{Q} among all other partitions [59]. For any sensor located at a position \mathbf{p}_i , its *Voronoi neighbors* $\mathcal{N}_{\mathcal{V},i} \subset \mathbf{P}$ are defined as the sensors corresponding to adjacent Voronoi cells such that:

$$\mathcal{N}_{\mathcal{V},i} = \{j \in \{1, \dots, n\} : V_i \cap V_j \neq \emptyset, j \neq i\}. \quad (5)$$

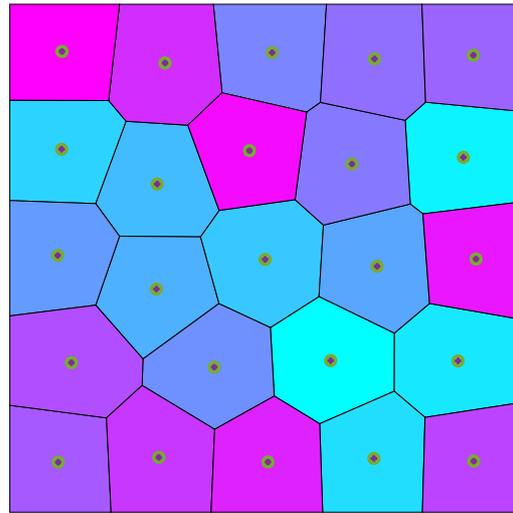


Figure 1. Example Voronoi partition of a 2D plane divided into number of regions with their associated points.

Considering the above definition, one can rewrite (3) as:

$$\mathcal{H}_{\mathcal{V}}(\mathbf{P}, \mathcal{V}(\mathbf{P})) = \sum_{i=1}^n \int_{V_i(\mathbf{P})} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q}. \quad (6)$$

By taking the partial derivative of (6) with respect to \mathbf{p}_i , the following is obtained:

$$\frac{\partial \mathcal{H}_{\mathcal{V}}}{\partial \mathbf{p}_i}(\mathbf{P}, \mathcal{V}(\mathbf{P})) = \int_{V_i} \frac{\partial}{\partial \mathbf{p}_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q}, \quad (7)$$

where it is assumed that f does not have any discontinuities. Furthermore, considering $f(x) = -x^2$, the multicenter cost function in (6) becomes:

$$\mathcal{H}_{\mathcal{V}}(\mathbf{P}, \mathcal{V}(\mathbf{P})) = - \sum_{i=1}^n \int_{V_i(\mathbf{P})} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi(\mathbf{q}) d\mathbf{q} := - \sum_{i=1}^n J_{V_i, \mathbf{p}_i}, \quad (8)$$

where J_{V_i, \mathbf{p}_i} is the polar moment of inertia of V_i about \mathbf{p}_i . Consequently, (8) reduces to:

$$\frac{\partial \mathcal{H}_{\mathcal{V}}}{\partial \mathbf{p}_i} = 2M_{V_i}(\mathbf{C}_{V_i} - \mathbf{p}_i), \quad (9)$$

where $M_{V_i} \in \mathbb{R}_+$ and $\mathbf{C}_{V_i} \in \mathbb{R}^3$ are the mass and center of mass (centroid) of the corresponding Voronoi partition V_i with respect to the density function $\phi(\mathbf{q})$. It is clear from (9) that the critical points of $\mathcal{H}_{\mathcal{V}}(\mathbf{P}, \mathcal{V}(\mathbf{P}))$ are the configurations $\mathbf{P} \in \mathcal{Q}^n$ where $\mathbf{p}_i = \mathbf{C}_{V_i} \forall i$, which are referred to as *centroidal Voronoi configurations*.

4. 3D Coverage Problems

Consider a 3D bounded region of interest $\mathcal{S} \subset \mathbb{R}^3$. A multi-vehicle system can perform coverage tasks over \mathcal{S} where coverage objective may vary according to the problem in hand. Definitions of the considered *barrier* and *sweep* coverage problems in 3D are defined next.

Problem 1. (3D Barrier Coverage) Deploy a network of vehicles/sensors to form a static arrangement over some region $\mathcal{B} \subset \mathcal{S}$ (i.e., a barrier) maximizing the sensing performance of the overall network to detect any intruder going through the barrier.

A special case of the above problem is when deploying the sensors over a planar region within \mathcal{S} defined by $\mathcal{B} = \{(x, y, z) : \sigma(x, y, z) = 0\}$ where $\sigma(x, y, z)$ is the plane's equation.

Note that for this problem to be solvable, the number of vehicles/sensors needed depends on the size of \mathcal{S} and the sensing range of all vehicles (assuming a homogeneous system).

Problem 2. (3D Sweep Coverage) Consider a group of vehicles whose overall sensing range is not large enough to achieve complete coverage over \mathcal{S} . It is required to survey the region \mathcal{S} by moving the whole group across \mathcal{S} as a dynamic formation over some region $\mathcal{F}(t) \subset \mathcal{S}$.

The coverage task can be considered completed once the whole region \mathcal{S} is scanned/surveyed. Alternatively, the coverage task can be done repetitively over a specified period of time (i.e., continuously survey \mathcal{S} for several times).

Note that $\mathcal{F}(t)$ can be of any 3D shape in which its motion along \mathcal{S} following a certain pattern can achieve complete coverage. A special case is when $\mathcal{F}(t)$ is a plane which is considered in this work. In this case, $\mathcal{F}(t)$ will be referred to as the *sweeping plane*. The dynamics of the sweeping plane $\mathcal{F}(t)$ can be determined in a way to achieve complete coverage over \mathcal{S} . It is also assumed that $\mathcal{F}(t)$ can change size and shape over time which can be utilized for other motion objectives such as obstacle avoidance as will be shown later.

Problem Formulation

The aim of this work is to develop distributed control laws for multi-UAV systems to address Problems 1 and 2. We consider a system of n homogeneous vehicles (UAVs/AUVs) with a single integrator motion model given by:

$$\dot{\mathbf{p}}_i(t) = \mathbf{u}_i(t), \quad (10)$$

where $\mathbf{p}_i \in \mathbb{R}^3$ is the i -th vehicle position defined in some inertial frame $\{\mathcal{I}\}$, and $\mathbf{u}_i \in \mathbb{R}^3$ is its control input (velocity) where

$$\|\mathbf{u}_i\| \leq u_{max}. \quad (11)$$

All vehicles can sense events in the environment within a sensing range $r_s > 0$. Furthermore, any vehicle can exchange information with nearby vehicles within some communication range $r_c > 0$. Obviously, it is assumed that $r_c > r_s$ so that it is possible to design control laws which can maintain the connectivity of the network with minimal sensors overlapping. It is assumed that the vehicles are homogeneous in terms of sensing and communication ranges. However, they can be of different types, sizes and/or weights.

5. Distributed Coverage Control Strategies

The proposed control schemes to address Problems 1 and 2 are based on a self-deployment method for the multi-vehicle system over a planar region $\mathcal{B} \subset \mathcal{S}$ which is static for barrier coverage problems and dynamic for sweep coverage problems. We consider a set of l vertices E to describe the boundary of \mathcal{B} as a polygon such that $E = \{e_1, e_2, \dots, e_l\} \subset \mathcal{S}$; clearly, $l > 2$. Lloyd's algorithm is adopted in the designed controllers to guide the vehicles to reach the instantaneous centroids of their associated Voronoi regions over \mathcal{B} (i.e., reaching the centroidal Voronoi configuration). Once this is reached, an optimal coverage over \mathcal{B} is achieved. Furthermore, for sweeping problems, the designed dynamics of \mathcal{B} will achieve coverage over \mathcal{S} .

5.1. Online Computation of Centroidal Voronoi Configurations

The developed control law requires each vehicle to be able to compute the centroid of its Voronoi region in a distributed fashion based only on information exchanged with vehicles within its neighbourhood. We extend the approach proposed in [1] to compute Voronoi cells for planar regions in 3D in a distributed fashion.

To simplify the mathematical development, a new 3D coordinate frame $\{\mathcal{B}\}$ attached to \mathcal{B} is needed. The origin of $\{\mathcal{B}\}$ can be selected to be one of the barrier vertices defined as

$\mathcal{O}_B = e_1$. Furthermore, the axes of $\{\mathcal{B}\}$ are defined using the orthonormal basis $\{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$ where:

$$\begin{aligned} \vec{a}_1 &= \frac{e_2 - e_1}{\|e_2 - e_1\|} \\ \vec{a}_2 &= \frac{h(\vec{a}_1, \vec{b})}{\|h(\vec{a}_1, \vec{b})\|}, \quad \vec{b} = \frac{e_l - e_1}{\|e_l - e_1\|} \\ \vec{a}_3 &= \vec{a}_1 \times \vec{a}_2, \end{aligned} \tag{12}$$

where $h(\vec{w}_1, \vec{w}_2) = \vec{w}_2 - (\vec{w}_1 \cdot \vec{w}_2)\vec{w}_1$ is a mapping function which gives an orthogonal vector to $\vec{w}_1 \in \mathbb{R}^3$ directed towards $\vec{w}_2 \in \mathbb{R}^3$.

All computations needed to find Voronoi centroids are carried out in the $\{\mathcal{B}\}$ frame through a transformation between $\{\mathcal{B}\}$ and the inertial frame $\{\mathcal{I}\}$. Let $v^{\mathcal{I}} \in \mathbb{R}^3$ be a vector defined in the $\{\mathcal{I}\}$ frame. This vector can be transformed to the $\{\mathcal{B}\}$ coordinate frame using a transformation matrix $T_{\mathcal{I}}^{\mathcal{B}}$ as follows:

$$\begin{bmatrix} v^{\mathcal{B}} \\ 1 \end{bmatrix} = T_{\mathcal{I}}^{\mathcal{B}} \begin{bmatrix} v^{\mathcal{I}} \\ 1 \end{bmatrix}, \tag{13}$$

where $T_{\mathcal{I}}^{\mathcal{B}}$ is a 4×4 affine transformation matrix given by:

$$T_{\mathcal{I}}^{\mathcal{B}} = \left(T_{\mathcal{B}}^{\mathcal{I}}\right)^{-1} = \begin{bmatrix} \vec{a}_1 & \vec{a}_2 & \vec{a}_3 & \mathcal{O}_B \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}.$$

Note that we represent the transformation using an augmented matrix to consider both rotation and translation in a single matrix multiplication. Furthermore, $\vec{a}_1, \vec{a}_2, \vec{a}_3$, and \mathcal{O}_B are column vectors.

From now on, vectors represented in the inertial frame will be represented without the \mathcal{I} superscript for simplicity. Given a UAV at position p_i , it is required to compute instantaneous Voronoi centroid C_{V_i} of its projection onto \mathcal{B} . First, the following assumption is made.

Assumption 1. Each Voronoi cell $V_i \subset \mathcal{B}$, generated by the projection of UAV i onto \mathcal{B} , is a convex polygon defined by m vertices $\{v_1^{\mathcal{B}}, v_2^{\mathcal{B}}, \dots, v_m^{\mathcal{B}}\}$ where $v_j^{\mathcal{B}} = (\bar{x}_j, \bar{y}_j, 0)$, $j = \{1, \dots, m\}$.

The proposed approach can now be described in these steps:

- S1: Transform the position p_i into the $\{\mathcal{B}\}$ frame to obtain $p_i^{\mathcal{B}} = [\bar{x}_i, \bar{y}_i, \bar{z}_i]^T$ by applying (13).
- S2: Compute the projection of $p_i^{\mathcal{B}}$ onto \mathcal{B} defined in the $\{\mathcal{B}\}$ frame by setting $\bar{z}_i = 0$ as $\tilde{p}_i^{\mathcal{B}} = [\bar{x}_i, \bar{y}_i, 0]^T$.
- S3: Compute the Voronoi cell centroid $\tilde{C}_{V_i}^{\mathcal{B}} = [\bar{c}_{x,V_i}, \bar{c}_{y,V_i}, 0]^T$ associated with $\tilde{p}_i^{\mathcal{B}}$ using the following [1]:

$$\begin{aligned} M_{V_i} &= \frac{1}{2} \sum_{k=1}^m (\bar{x}_k \bar{y}_{k+1} - \bar{x}_{k+1} \bar{y}_k) \\ \bar{c}_{x,V_i} &= \frac{1}{6M_{V_i}} \sum_{k=1}^m (\bar{x}_k + \bar{x}_{k+1})(\bar{x}_k \bar{y}_{k+1} - \bar{x}_{k+1} \bar{y}_k) \\ \bar{c}_{y,V_i} &= \frac{1}{6M_{V_i}} \sum_{k=1}^m (\bar{y}_k + \bar{y}_{k+1})(\bar{x}_k \bar{y}_{k+1} - \bar{x}_{k+1} \bar{y}_k), \end{aligned} \tag{14}$$

where $v_{m+1}^{\mathcal{B}} = (\bar{x}_{m+1}, \bar{y}_{m+1}, 0) = v_1^{\mathcal{B}}$. These equations are obtained considering Assumption 1 and a constant distribution density function $\phi(q) = 1$. Voronoi cell vertices $\{v_1^{\mathcal{B}}, v_2^{\mathcal{B}}, \dots, v_m^{\mathcal{B}}\}$ can be determined based on locations of Voronoi neighbours in a distributed fashion (see Remark 2).

- S4: Transform $\tilde{C}_{V_i}^{\mathcal{B}}$ to the inertial frame $\{\mathcal{I}\}$ to get $\tilde{C}_{V_i} = [c_{x,V_i}, c_{y,V_i}, c_{z,V_i}]^T$ using (13).

Remark 2. The vertices of a Voronoi cell V_i can be found as the circumcenters of triangles formed by $\tilde{p}_i^{\mathcal{B}}$ and any two of its Voronoi neighbours. A triangle made by three points p_1 , p_2 and p_3 with an area of A has a circumcenter at [1]:

$$\begin{aligned} \text{circumcenter} = & \frac{1}{4A} \left(\|\alpha_{32}\|^2 (\alpha_{21} \cdot \alpha_{13}) p_1 \right. \\ & \left. + \|\alpha_{13}\|^2 (\alpha_{32} \cdot \alpha_{21}) p_2 + \|\alpha_{21}\|^2 (\alpha_{13} \cdot \alpha_{32}) p_3 \right), \end{aligned} \quad (15)$$

where $\alpha_{ls} = p_l - p_s$.

5.2. Barrier Coverage Control Design

In order to present our control design, some technical assumptions need to be made as follows.

Assumption 2. The communication graph $G(\mathcal{U}, \mathcal{E})$ remains connected for $t \geq 0$.

Assumption 3. Each UAV is capable of estimating the Voronoi cell centroid of its projected position onto \mathcal{B} at any time using only information from UAVs within its communication range (i.e., UAVs in its neighborhood \mathcal{N}_i).

Assumption 4. The Voronoi neighbours of $\tilde{p}_i^{\mathcal{B}}$ correspond to UAVs within the neighbourhood \mathcal{N}_i .

Assumption 5. The initial configuration of the multi-UAV system satisfies the following condition: $\frac{p_i(0) - p_j(0)}{\|p_i(0) - p_j(0)\|} \cdot \mathbf{a}_3 \neq \pm 1, \forall i, j \in \{1, \dots, n\}, i \neq j$ where \mathbf{a}_3 is the barrier's normal as defined in (12).

Assumption 2 is made to make sure that updated information about the barrier \mathcal{B} is available to all vehicles at any time during the motion. This is essential in cases where \mathcal{B} is dynamic. For example, a decision could be made by one of the UAVs to apply changes to the shape of \mathcal{B} based on some detected obstacles. Such information needs to be shared among all vehicles so that they can compute their Voronoi regions accordingly. It is possible to ensure the connectivity of $G(\mathcal{U}, \mathcal{E})$ by enforcing constraints on \mathcal{B} based on the number of vehicles and communication ranges to ensure that Voronoi neighbors are within the communication range of each other. Assumptions 3 and 4 ensure that UAVs can determine Voronoi centroids in a distributed fashion. Finally, Assumption 5 ensures that all UAVs are initially located at positions with unique projections onto \mathcal{B} .

Now, the main results of this section can be presented. Consider the following control law $\mathbf{u}_i : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ based on Lloyd's algorithm:

$$\mathbf{u}_i = \bar{K}_i (\tilde{C}_{V_i} - p_i), \quad (16)$$

where the parameter \bar{K}_i is a diagonal positive definite gain matrix. We also propose a more practical bounded control law $\mathbf{u}_i : \mathbb{R}^3 \rightarrow \Gamma$ which can satisfy bounds u_{max} on the control input such that $\Gamma = \{\mathbf{u} \in \mathbb{R}^3 : \|\mathbf{u}\| \leq u_{max}\}$. It is given as follows:

$$\mathbf{u}_i = K_i \tanh \left(\gamma_i (\tilde{C}_{V_i} - p_i) \right), \quad (17)$$

where $K_i = \text{diag}\{k_{i,x}, k_{i,y}, k_{i,z}\}$ is a positive definite diagonal matrix, $\gamma_i > 0$, and $\tanh(v)$ is the hyperbolic tangent function defined element-wise for any vector $v \in \mathbb{R}^3$. Clear, the bound of this control law depends on the gain matrix K_i as follows:

$$\|\mathbf{u}_i\| \leq \sqrt{k_{i,x}^2 + k_{i,y}^2 + k_{i,z}^2} := u_{i,max}. \quad (18)$$

Theorem 1. Consider a multi-UAV system with n vehicles whose motion are modelled as (10). Under Assumptions 2–5, the distributed control law (16) (or (17) for bounded inputs) along with the algorithm in S1–S4 solves the 3D barrier coverage problem defined in Problem 1.

Proof. Let P_e be a centroidal Voronoi configuration. We define a Lyapunov candidate function as

$$V_1(P(t)) = \frac{1}{2}\bar{\mathcal{H}} - \frac{1}{2}\mathcal{H}_V(P(t), \mathcal{V}(P(t))), \quad (19)$$

where \mathcal{H}_V is defined in (8) with $\phi(q) = 1$, and $\bar{\mathcal{H}} = \mathcal{H}_V(P_e, \mathcal{V}(P_e))$ which is constant. Hence, $V_1(P_e) = 0$. Furthermore, $\bar{\mathcal{H}} \geq \mathcal{H}_V(P(t), \mathcal{V}(P(t))) \quad \forall P \in \mathcal{S}^n \setminus \{P_e\}$ since a centroidal Voronoi configuration is optimal for \mathcal{H}_V among all other configurations (see Proposition 2.14 in [59]). This indicates that $V_1(P(t)) > 0$ which makes it a valid Lyapunov function.

The time derivative of (19) can be obtained using (9) and (10) and the control law (16) as:

$$\dot{V}_1 = \frac{-1}{2} \sum_{i=1}^n \left(\frac{\partial \mathcal{H}_V}{\partial \mathbf{p}_i} \right)^T \mathbf{u}_i \quad (20)$$

$$= - \sum_{i=1}^n (\tilde{\mathcal{C}}_{V_i} - \mathbf{p}_i)^T \bar{\mathbf{L}}_i (\tilde{\mathcal{C}}_{V_i} - \mathbf{p}_i) \quad (21)$$

$$\leq - \sum_{i=1}^n \lambda_{\min}(\bar{\mathbf{L}}_i) \|\tilde{\mathcal{C}}_{V_i} - \mathbf{p}_i\|^2, \quad (22)$$

where $\bar{\mathbf{L}}_i = M_{V_i} \bar{\mathbf{K}}_i$, and $\lambda_{\min}(\bar{\mathbf{L}}_i)$ is the smallest eigenvalue of $\bar{\mathbf{L}}_i$. It is clear from (22) that $\dot{V}_1 < 0 \quad \forall P \in \mathcal{S}^n \setminus \{P_e\}$ since $\bar{\mathbf{K}}_i$ is positive definite (i.e., $\lambda_{\min}(\bar{\mathbf{L}}_i) > 0$). Hence, the set of centroidal Voronoi configurations P_e is locally asymptotically stable, and $\lim_{t \rightarrow \infty} \mathbf{p}_i = \tilde{\mathcal{C}}_{V_i}, \quad \forall i$.

Similarly, for the bounded control law, the time derivative of $V_1(P(t))$ can be obtained by substituting (17) into (20) as follows:

$$\dot{V}_1 = - \sum_{i=1}^n (\tilde{\mathcal{C}}_{V_i} - \mathbf{p}_i)^T \mathbf{L}_i \tanh(\gamma_i (\tilde{\mathcal{C}}_{V_i} - \mathbf{p}_i)) \quad (23)$$

$$:= - \sum_{i=1}^n \mathbf{e}_i^T \mathbf{L}_i \tanh(\gamma_i \mathbf{e}_i). \quad (24)$$

It is evident from (24) that $\dot{V}_1 < 0 \quad \forall P \in \mathcal{S}^n \setminus \{P_e\}$ since the hyperbolic tangent function is an odd function and $\gamma_i > 0$. This implies that P_e is also locally asymptotically stable under the application of the bounded control law (17).

Therefore, the control law (16) (or (17)) guarantees that the vehicles will converge to centroidal Voronoi configurations which maximizes the sensing performance over the barrier \mathcal{B} according to Proposition 2.13 in [59]. Furthermore, Assumptions 2–5 ensures that all vehicles can compute the centroids of their Voronoi cells in a distributed fashion at all times with no overlapping following the algorithm in S1–S4. This completes the proof. \square

Note that the proposed control law ensures that the vehicles will converge to centroidal Voronoi configurations generated by their projections onto \mathcal{B} . Thus, all the vehicles will eventually reach \mathcal{B} such that $\lim_{t \rightarrow \infty} \mathbf{p}_i = \tilde{\mathcal{C}}_{V_i} \in \mathcal{B}$ even if they are initially deployed at some positions $\mathbf{p}_i(0) \notin \mathcal{B}$. Additionally, the trajectory of each vehicle remains within its Voronoi region which does not intersect with any other regions by definition. This guarantees that vehicles motions are collision-free using the proposed control laws.

5.3. Sweep Coverage Control Design

Theorem 1 shows that the proposed control laws can force the vehicles to reach a specified region within the 3D space (i.e., the barrier) and constrain their motion within

that region. This is extended in this section to address the sweep coverage control problems. It can be achieved by enforcing vehicles to deploy over some dynamical "virtual" region whose motion is determined by the group.

In general, motion coordination control laws for coverage problems needs to satisfy the following objectives:

- O1** Avoid collisions with other vehicles while maintaining a certain formation as a group
- O2** Avoid collisions with obstacles within the environment
- O3** Achieve optimal coverage of the targeted environment \mathcal{S} collaboratively

The proposed sweeping algorithm targets these objective on two levels. At a lower level, the vehicles motions are constrained within a dynamic "sweeping" region $\mathcal{F}(t) \subset \mathcal{S}$, and they maintain an optimal formation over that region for maximal sensing. This achieves objective **O1**. At a higher level, decisions can be made in real-time collaboratively by the vehicles to decide the dynamics of $\mathcal{F}(t)$ (i.e., $\dot{\mathcal{F}}(t)$). Note that it is also possible to apply deformations to $\mathcal{F}(t)$ (will be shown in simulations) as long as the deformed region is large enough for the vehicles to distribute over with safe spacing. This provides a good way in addressing objectives **O2** and **O3**. In particular, the trajectory of $\mathcal{F}(t)$ will result in sweeping the whole environment \mathcal{S} providing optimal coverage. Moreover, obstacle avoidance can be achieved by only changing the dynamics of $\mathcal{F}(t)$ rather than having each vehicle reacting independently to obstacles.

For $\mathbf{r} \in \mathcal{F}(t)$, the motion dynamics of the sweeping region can be described as follows:

$$\dot{\mathbf{r}} = \mathbf{g}(t, \mathbf{r}), \quad \|\mathbf{g}(t, \mathbf{r})\| \leq \bar{g}, \quad \|\dot{\mathbf{g}}(t, \mathbf{r})\| \leq \bar{\dot{g}}, \quad (25)$$

where $\mathbf{g}(t, \mathbf{r}) \in \mathbb{R}^3$ is a desired velocity profile, $\bar{g}, \bar{\dot{g}} > 0$ are upper bound design parameters to account for dynamic limitations where the following condition must hold:

$$\bar{g} \leq u_{max}. \quad (26)$$

In other words, the speed of $\mathcal{F}(t)$ should not be larger than the maximum physical speed that can be achieved by the vehicles. Note that $\mathbf{g}(t, \mathbf{r}) \in \mathbb{R}^3$ can have any direction. However, for simplicity in achieving sweeping coverage, it assumed that $\mathcal{F}(t)$ is a planar region, and it is defined similar to \mathcal{B} (see Section 5.1) at different time instants. A simple example is moving $\mathcal{F}(t)$ in the direction of its normal (i.e., $\bar{\mathbf{a}}_3$) with a constant *sweeping speed* $g_0 > 0$ such that:

$$\mathbf{g}(t, \mathbf{r}) = g_0 \bar{\mathbf{a}}_3. \quad (27)$$

More complex movements can be achieved depending on the considered environment shape and nearby obstacles. One can also adopt a 3D holonomic or non-holonomic model for (25) utilizing the available literature in obstacle for these models. Generally, obstacle avoidance can be achieved either by varying the dynamics in (25) or by dynamically deforming $\mathcal{F}(t)$. Simulation cases showing both approaches will be shown later. Note that the distributed behavior of the proposed algorithm is maintained in all these cases since information about $\mathbf{g}(t, \mathbf{r})$ are exchanged over the connected network following Assumption 2. At this point, we will leave out the design of (25) to a high-level controller shared among the vehicles while assuming the following:

Assumption 6. *The sweeping plane $\mathcal{F}(t)$ remains all the time within the sensing environment (i.e., $\mathcal{F}(t) \subset \mathcal{S}$, $\forall t > 0$), and its movement governed by (25) will completely span the volume of the sensing environment \mathcal{S} .*

The main results of this section can now be presented.

Theorem 2. *Consider a multi-UAV system of size n where each vehicle's motion model is represented by (10). The control law (16) along with the algorithm in S1–S4 defined for a sweeping*

region $\mathcal{F}(t)$ whose dynamics is governed by (25) solves the 3D sweeping coverage problem defined in Problem 2 under Assumptions 2–6 and the condition (26).

Proof. Let

$$e_{C_i} = \tilde{C}_{V_i} - p_i \rightarrow \dot{e}_{C_i} = \dot{\tilde{C}}_{V_i} - u_i, \tag{28}$$

where $u_i(t)$ is given by (16). Now, define a Lyapunov candidate function as follows:

$$\begin{aligned} V_2(t) &:= V_2(e_{C_1}(t), \dots, e_{C_n}(t), \dot{e}_{C_1}(t), \dots, \dot{e}_{C_n}(t)) \\ &= V_1 + \frac{1}{2} \sum_{i=1}^n \left(k_1 e_{C_i}^T e_{C_i} + k_2 \dot{e}_{C_i}^T \dot{e}_{C_i} \right), \end{aligned} \tag{29}$$

where V_1 is defined in (19), and k_1 and k_2 are positive kinematic constants with units of m^2 and $m^2 \cdot s^2$ to match the physical units of V_1 which is m^4 . The choice (29) guarantees that $V_2(t) > 0$ for

$$(e_{C_1}(t), \dots, e_{C_n}(t), \dot{e}_{C_1}(t), \dots, \dot{e}_{C_n}(t)) \neq (0, \dots, 0),$$

and $V_2(0, \dots, 0) = 0$ is also true.

The time derivative of (29) is obtained using (10) and (16) as follows:

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + \sum_{i=1}^n \left(k_1 e_{C_i}^T \dot{e}_{C_i} + k_2 \dot{e}_{C_i}^T \ddot{e}_{C_i} \right) \\ &= \dot{V}_1 + \sum_{i=1}^n \left(k_1 e_{C_i}^T (\dot{\tilde{C}}_{V_i} - \bar{K}_i e_{C_i}) + k_2 \dot{e}_{C_i}^T (\dot{\tilde{C}}_{V_i} - \bar{K}_i \dot{e}_{C_i}) \right) \\ &\leq \dot{V}_1 + \sum_{i=1}^n \left(k_1 \|\dot{\tilde{C}}_{V_i}\| \|e_{C_i}\| - \lambda_{\min}(\bar{K}_i) k_1 \|e_{C_i}\|^2 + \bar{g} k_2 \|\dot{e}_{C_i}\| - \lambda_{\min}(\bar{K}_i) k_2 \|\dot{e}_{C_i}\|^2 \right) \end{aligned} \tag{30}$$

Recall that $\dot{V}_1 < 0$ which was established in (24). Therefore, the time derivative of V_2 is negative outside the compact set $B_\Gamma = B_{\Gamma_1} \cup \dots \cup B_{\Gamma_n}$ where $B_{\Gamma_i} = \{e_{C_i} \in \mathbb{R}^3, \dot{e}_{C_i} \in \mathbb{R}^3 : \|e_{C_i}\| > \Gamma_{i,1}, \|\dot{e}_{C_i}\| > \Gamma_{i,2}\}$ and $\Gamma_{i,1}$ and $\Gamma_{i,2}$ are given by:

$$\Gamma_{i,1} = \frac{\|\dot{\tilde{C}}_{V_i}\|}{\lambda_{\min}(\bar{K}_i)}, \quad \Gamma_{i,2} = \frac{\bar{g}}{\lambda_{\min}(\bar{K}_i)}, \tag{31}$$

which represent closed balls with radius $\Gamma_{i,1}$ and $\Gamma_{i,2}$ respectively. Thus, starting from any initial condition outside the set B_Γ , the errors will converge to the closed set B_Γ in finite time and stay there forever. That is,

$$\limsup_{t \rightarrow \infty} \|\tilde{C}_{V_i} - p_i\| \leq \Gamma_{i,1}, \quad \limsup_{t \rightarrow \infty} \|\dot{\tilde{C}}_{V_i} - u_i\| \leq \Gamma_{i,2},$$

which means that the errors will be uniformly ultimately bounded with respect to B_{Γ_i} . Moreover, the radius of B_{Γ_i} can be made arbitrary small by increasing \bar{K}_i such that $\lambda_{\min}(\bar{K}_i) \gg \bar{g}$ and $\lambda_{\min}(\bar{K}_i) \gg \|\dot{\tilde{C}}_{V_i}\|$. Furthermore, since $\tilde{C}_{V_i} \in \mathcal{F}(t)$, its derivative follows (25) (i.e., $\limsup_{t \rightarrow \infty} \|g(t, \tilde{C}_{V_i}) - u_i\| \leq \Gamma_{i,2}$). Hence, all the vehicles will converge to their centroidal Voronoi configurations over $\mathcal{F}(t)$ and follow its trajectory associated with (25). According to Assumption 6, the movement of the multi-UAV along the trajectory of $\mathcal{F}(t)$ solves the sweeping coverage problem. This completes the proof. \square

Theorem 3. Consider a multi-UAV system of size n where each vehicle’s motion model is represented by (10). Furthermore, consider using the algorithm in S1–S4 defined for a sweeping region $\mathcal{F}(t)$ whose dynamics is governed by (25) so that the vehicles can compute their centroidal Voronoi configurations. The bounded control law (17) solves the 3D sweeping coverage problem defined in Problem 2 under Assumptions 2–6 and the condition (26).

Proof. Again, consider the errors definitions in (28). We now define a Lyapunov candidate function as follows:

$$\begin{aligned}
 V_3(t) &:= V_3(e_{C_1}(t), \dots, e_{C_n}(t), \dot{e}_{C_1}(t), \dots, \dot{e}_{C_n}(t)) \\
 &= V_1 + \sum_{i=1}^n \left(\frac{k_3}{\gamma_i} [1 \ 1 \ 1]^T \log(\cosh(\gamma_i e_{C_i})) + \frac{1}{2} k_4 \dot{e}_{C_i}^T \dot{e}_{C_i} \right), \tag{32}
 \end{aligned}$$

where $\log(v) \in \mathbb{R}^3$ and $\cosh(v) \in \mathbb{R}^3$ are defined element-wise for any vector $v \in \mathbb{R}^3$, k_3 and k_4 are positive kinematic constants, and u_i is defined using (17). Furthermore, let $Sech^2(v) : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ be a mapping function, based on the hyperbolic secant function, which maps the vector $v = [v_1, v_2, v_3]^T$ into a diagonal matrix as follows:

$$Sech^2(v) = \begin{bmatrix} sech^2(v_1) & 0 & 0 \\ 0 & sech^2(v_2) & 0 \\ 0 & 0 & sech^2(v_3) \end{bmatrix}.$$

The time derivative of V_3 can then be obtained as follows:

$$\begin{aligned}
 \dot{V}_3 &= \dot{V}_1 + \sum_{i=1}^n \left(k_3 \tanh(\gamma_i e_{C_i})^T (\dot{\check{C}}_{V_i} - K_i \tanh(\gamma_i e_{C_i})) + k_4 \dot{e}_{C_i}^T (\ddot{\check{C}}_{V_i} - \gamma_i K_i Sech^2(\gamma_i e_{C_i}) \dot{e}_{C_i}) \right) \\
 &\leq \dot{V}_1 + \sum_{i=1}^n \left(k_3 \|\dot{\check{C}}_{V_i}\| \|\tanh(\gamma_i e_{C_i})\| - \lambda_{\min}(K_i) k_3 \|\tanh(\gamma_i e_{C_i})\|^2 \right. \\
 &\quad \left. + k_4 \|\ddot{\check{C}}_{V_i}\| \|\dot{e}_{C_i}\| - \gamma_i \lambda_{\min}(\Lambda_i) k_4 \|\dot{e}_{C_i}\|^2 \right), \tag{33}
 \end{aligned}$$

where $\Lambda_i = K_i Sech^2(\gamma_i e_{C_i})$.

Similar to the previous analysis, \dot{V}_3 is negative outside the compact set $D_Y = D_{Y_1} \cup \dots \cup D_{Y_n}$ where $D_{Y_i} = \{e_{C_i} \in \mathbb{R}^3, \dot{e}_{C_i} \in \mathbb{R}^3 : \|\tanh(\gamma_i e_{C_i})\| > Y_{i,1}, \|\dot{e}_{C_i}\| > Y_{i,2}\}$ where $Y_{i,1}$ and $Y_{i,2}$ are given by:

$$Y_{i,1} = \frac{\|\dot{\check{C}}_{V_i}\|}{\lambda_{\min}(K_i)}, \quad Y_{i,2} = \frac{\bar{g}}{\gamma_i \lambda_{\min}(K_i)}. \tag{34}$$

Thus, the system trajectories will converge to the set D_Y starting from any initial condition, and stay there forever leading to the following:

$$\limsup_{t \rightarrow \infty} \left\| \tanh(\gamma_i (\check{C}_{V_i} - p_i)) \right\| \leq Y_{i,1}, \quad \limsup_{t \rightarrow \infty} \|\dot{\check{C}}_{V_i} - u_i\| \leq Y_{i,2}.$$

Therefore, the tracking errors are uniformly ultimately bounded with respect to D_{Y_i} . By choosing, $\lambda_{\min}(K_i) \gg \bar{g}$ and $\gamma_i \lambda_{\min}(K_i) \gg \bar{g}$, the errors can be made arbitrarily small.

Moreover, $\check{C}_{V_i} \in \mathcal{F}(t) \rightarrow \dot{\check{C}}_{V_i} = g(t, \check{C}_{V_i})$ according to (25) and S1–S4. By Assumption 6, the multi-UAV system solves the sweeping coverage problem which completes the proof. \square

6. Validation & Discussion

Simulations were carried out to validate the performance of the proposed 3D coverage control laws in (16) and (17) using the algorithm in S1–S4 to compute the centroidal Voronoi configurations. Additionally, more simulation cases were performed to demonstrate the robustness of the proposed method and how obstacle avoidance can be incorporated within the overall framework. The following subsections provide details of these simulations and the obtained results.

6.1. Simulation Cases 1–4: Performance Validation

In the first set of simulations, a multi-UAV system of size $n = 20$ was used. All vehicles have been initially deployed to random locations in some predefined region

$\mathcal{I} = \{(x_i, y_i, z_i) : 0 \leq x_i, y_i \leq 4, 0 \leq z_i \leq 2, \forall i = \{1, \dots, 20\}\}$. The control design parameters were chosen to be $K_i = \text{diag}\{2.5, 0.5, 0.5\}$ and $\bar{K}_i = \text{diag}\{0.3, 0.3, 0.3\}$ for all vehicles.

The 3D barrier coverage problem was considered in the first two simulation cases where the unbounded control law (16) and the bounded control law (17) are used. The goal was to achieve optimal coverage over a barrier region defined according to $\mathcal{B} = \{(x, y, z) : x = 20, 0 \leq (y, z) \leq 10\}$. The obtained results for this case are presented in Figures 2–5. Figures 2 and 4 show the complete trajectories taken by all UAVs and their final locations which are optimally distributed over the barrier \mathcal{B} (rectangular area highlighted in yellow). The multi-UAV system reaches the centroidal Voronoi configurations and all vehicles form a static arrangement over \mathcal{B} . Note that the shape of \mathcal{B} and the number of UAVs were chosen arbitrarily just as a proof of concept. However, in practical applications, the number of UAVs and the design of the barrier can be considered as a design problem which depends on the UAVs sensing and communications capabilities. Having a larger number of vehicles may result in some overlapping between sensors field-of-view (FOV). On the other hand, It may not be possible to completely cover \mathcal{B} using lower number of UAVs than what is needed (i.e., the combined FOV of all sensors is less than the size (area/volume) of \mathcal{B}). Overall, the sensing performance will be maximized using the developed strategy. It can also be seen that the resultant trajectories are collision-free.

The time evolution of control inputs for all vehicles are shown in Figures 3 and 5 for both the bounded and unbounded control laws, respectively. Using (16) would require choosing the controller gain matrix \bar{K}_i properly in order to satisfy the constraint (11). This depends mostly on how far the vehicle is from its centroidal Voronoi configuration initially which indicates that tuning \bar{K}_i could not be ideal in practice. Alternatively, using (17) provides an easier way of choosing K_i to ensure that the physical limit on the vehicles velocity is respected (i.e., (11) is satisfied). This can be clearly seen in Figure 5 where the vehicles speed remains constant for the first 8 s until the barrier is reached at which the velocities drop down to zero, and the vehicles become statically distributed over \mathcal{B} . The upper bounds on $\|u_i\|$ in this case was $u_{i,max} = 2.6 \text{ m/s } \forall i$ which is in accordance with (18).

In the next two simulation cases, the sweeping coverage problem was considered. The task was to completely scan a 3D sensing region \mathcal{S} which was defined as $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3$ where

$$\begin{aligned}\mathcal{S}_1 &= \{(x, y, z) : 10 \leq x \leq 60, 0 \leq y \leq 10, 0 \leq z \leq 5\} \\ \mathcal{S}_2 &= \{(x, y, z) : 10 \leq x \leq 60, 0 \leq y \leq 2.5, 5 \leq z \leq 10\} \\ \mathcal{S}_3 &= \{(x, y, z) : 10 \leq x \leq 60, 7.5 \leq y \leq 10, 5 \leq z \leq 10\}.\end{aligned}$$

Based on the environment shape, an initial sweeping plane $\mathcal{F}(t = 0)$ was determined by the multi-UAV system as $\mathcal{F}(0) = \{(x, y, z) \in \mathcal{S} : x = 10\}$. The dynamics of $\mathcal{F}(t)$ was also considered to be the simplest case as in (27) where the sweeping plane is moving with a constant speed of $g_0 = 1.5 \text{ m/s}$ in a progressive direction that can result in a sweeping behavior, as will be shown. In more complex cases, some other patterns could be adopted for moving $\mathcal{F}(t)$ as a higher level decision making which can still be done in a distributed manner as the vehicles can exchange information over the connected network. For example, the literature on coverage path planning for single-vehicle systems can be utilized in this case.

Figures 6 and 7 shows the results for the sweeping coverage problem when using (16), and the results obtained when applying (17) are shown in Figures 8 and 9. For both cases, the vehicles move from their initial positions to quickly deploy over $\mathcal{F}(t)$ reaching their centroidal Voronoi configurations. As the position of $\mathcal{F}(t)$ evolves over time according to (27), the vehicles corresponding centroidal Voronoi configurations evolve accordingly since $\tilde{C}_{V_i} \in \mathcal{F}(t)$. Hence, the vehicles will start to move with the same velocity as of $\mathcal{F}(t)$. This can be clearly seen from Figures 7 and 9. You can see that the vehicles starts with a higher speed to reach the moving sweeping plane and achieve optimal distribution. Once

this is achieved (around $t = 10$ s), the vehicles are no longer moving within $\mathcal{F}(t)$ at which all velocities converge to $g_0 = 1.5$ m/s in the direction of the sweeping plane's movement. It is important to notice that the speed of $\mathcal{F}(t)$ (i.e., g_0) should be slower than the maximum velocity achievable by any vehicle ($u_{max} \geq g_0$). After scanning the desired region \mathcal{S} , the sweeping plane $\mathcal{F}(t)$ becomes static which reflects on all the vehicles as can be seen from the results where all velocities converge to 0. The complete trajectories of the vehicles along with the scanned 3D volume (highlighted in yellow) are shown in Figures 6 and 8 which confirms that the sweeping coverage problem is achieved over \mathcal{S} . These results clearly validate the performance of the proposed 3D coverage control strategy.

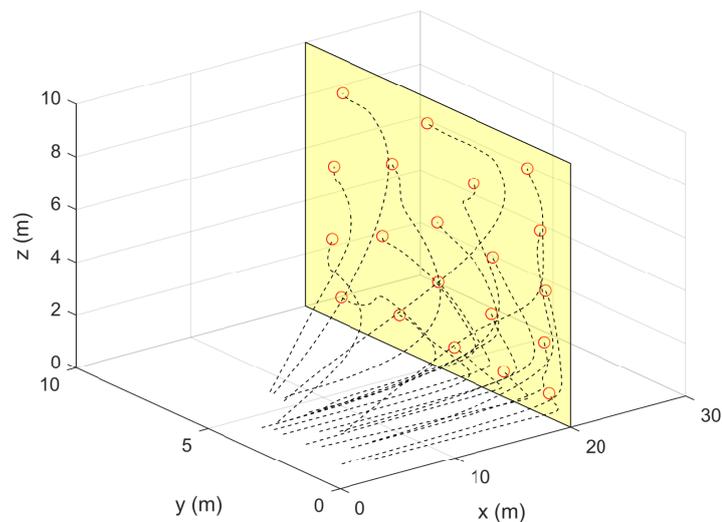


Figure 2. UAVs trajectories for the barrier coverage case using (16) (Simulation Case 1).

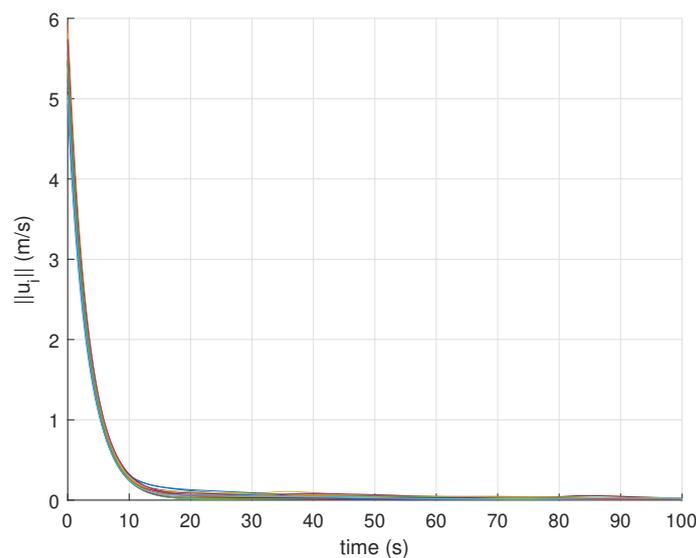


Figure 3. Norms of the control inputs applied to the UAVs for the barrier coverage case using (16) (Simulation Case 1).

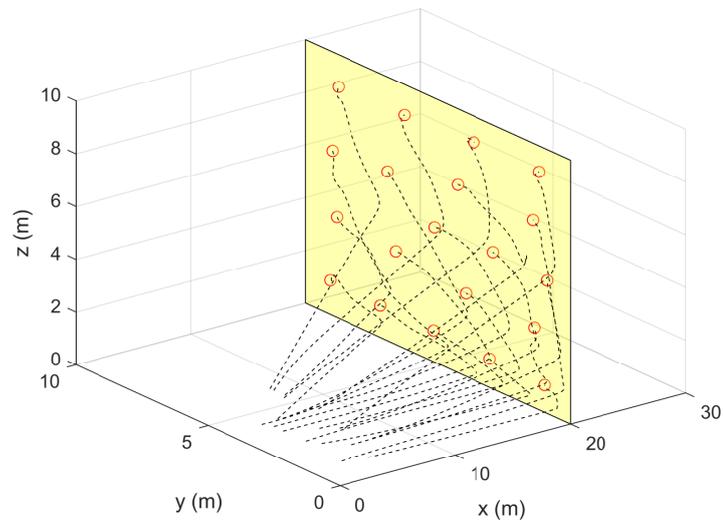


Figure 4. Sensors trajectories for a barrier coverage case using bounded control laws (17) (Simulation Case 2).

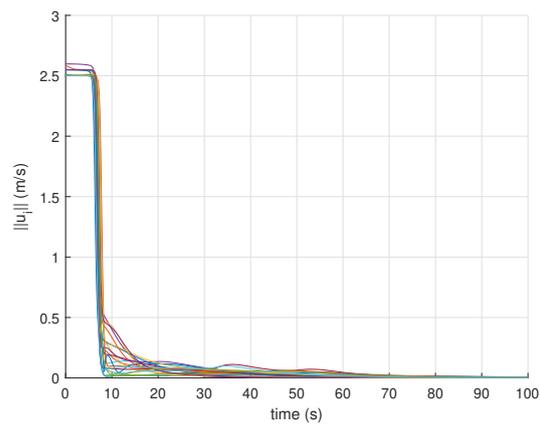


Figure 5. Sensors control inputs norms for a barrier coverage case using bounded control laws (17) (Simulation Case 2).

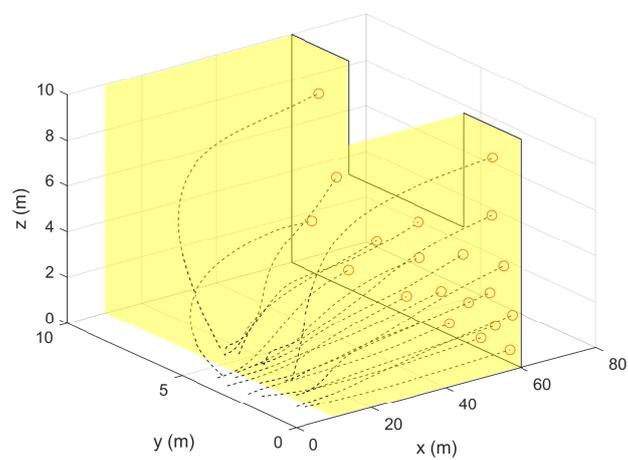


Figure 6. Sensors trajectories for a sweeping coverage case (Simulation Case 3).

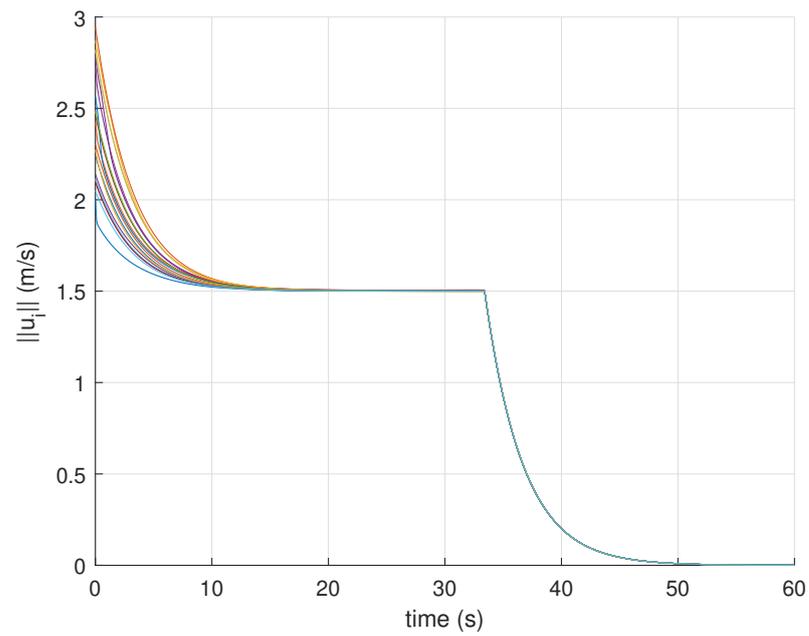


Figure 7. Sensors control inputs norms for a sweeping coverage case using (16) (Simulation Case 3).

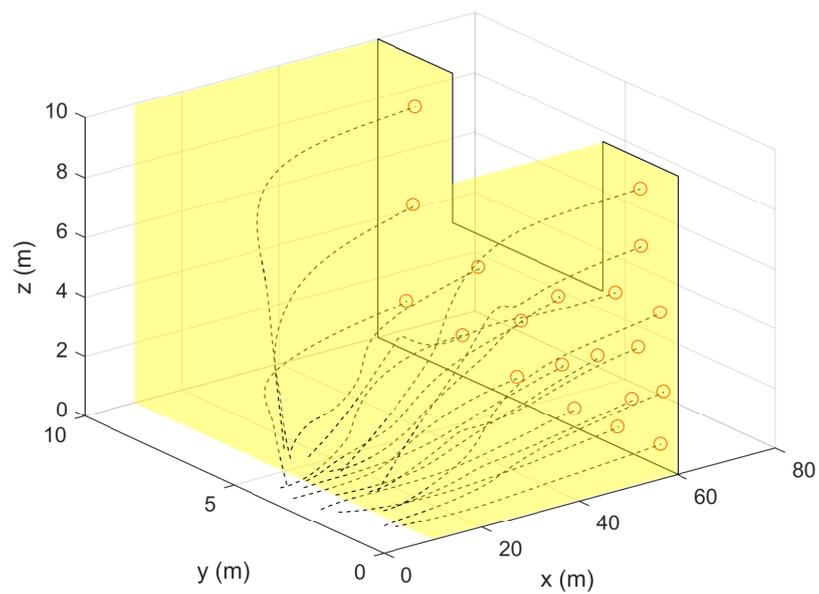


Figure 8. Sensors trajectories for a sweeping coverage case using bounded control laws (17) (Simulation Case 4).

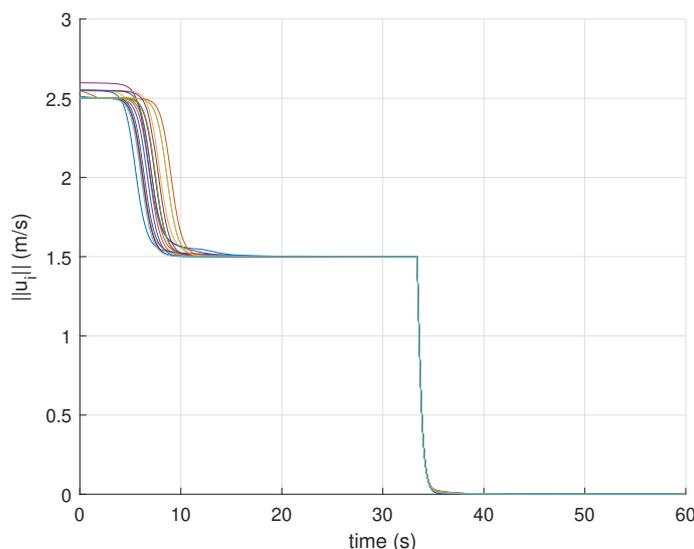


Figure 9. Sensors control inputs norms for a sweeping coverage case using bounded control laws (17) (Simulation Case 4).

6.2. Simulation Case 5: Robustness

Another simulation case was considered to show how the proposed method perform very well in situations where the number of active UAVs within the multi-UAV system change over time during a sweeping coverage mission. For example, when a number of UAVs fail, the whole group should still be able to continue their coverage task as long as there is enough number of active UAVs to finish the mission. This goes the same way when adding new vehicles to the system during the mission; however, this case was not considered here for brevity.

We considered a similar environment \mathcal{S} and sweeping plane choice $\mathcal{F}(t)$ as in simulations cases 3 and 4. Different time instants of the simulation are shown in Figure 10. The Voronoi regions generated by the UAVs over $\mathcal{F}(t)$ with their centroids are clearly highlighted to show how they change over time as the vehicles move. Once centroidal Voronoi configurations are reached, the vehicles move according to the plane's movement as was discussed earlier. However, some vehicles fail and become inactive at certain time instants as in Figure 10e,g,i,k. Whenever this occurs, the remaining active vehicles quickly adjust their distribution over $\mathcal{F}(t)$ while still moving in accordance with $\mathcal{F}(t)$. For example, at $t = 21$ s, one vehicle fail as shown in Figure 10e which directly indicates a change of the Voronoi partition of $\mathcal{F}(t)$. This results in a change of the centroidal Voronoi configurations which causes the vehicles to quickly adapt to the situation in a robust way by moving to the new centroidal Voronoi configurations. It can be noticed that once a vehicle fail, only the vehicles in the neighborhood of that vehicle will be affected (i.e., their Voronoi regions will be extended).

Initially, the multi-UAV system had a size of $n = 9$. The complete collision-free trajectories of all vehicles are shown in Figure 11 where 4 vehicles have failed during the mission (inactive UAVs), and the coverage task was completed efficiently by the remaining 5 vehicles (active UAVs). This clearly shows how robust and scalable our method can be. It is also worth mentioning that changes to $\mathcal{F}(t)$ can be applied in real-time if its size becomes larger/smaller than what the remaining vehicles can cover based on their combined sensing FOV. Such a decision can be autonomously made by the vehicles and shared among the connected network. The next simulation case shows how the proposed control laws work when such changes to $\mathcal{F}(t)$ are applied which is really important when considering obstacle avoidance.

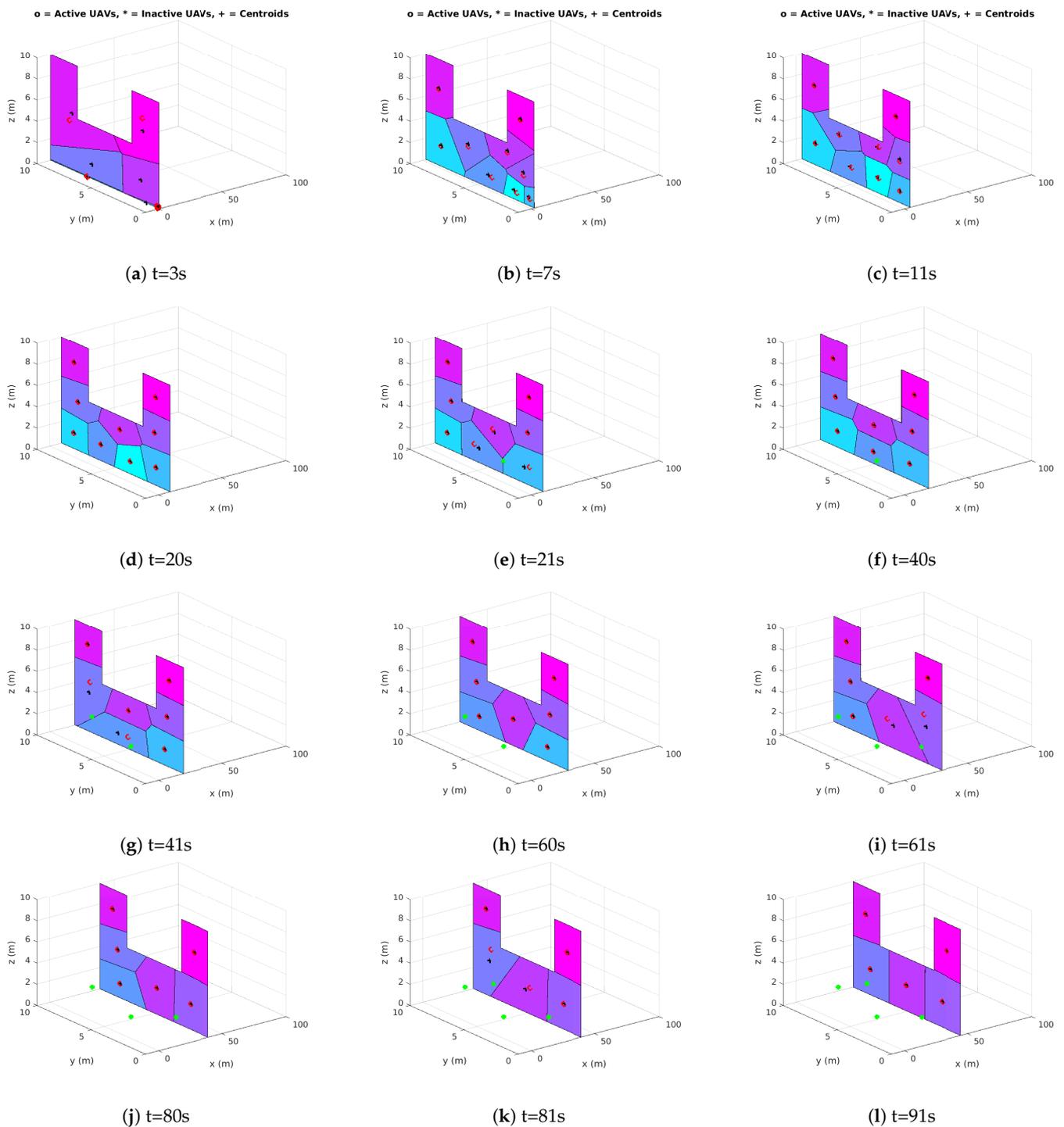


Figure 10. UAV locations at different time instants showing the robustness of the proposed approach against UAVs failures when they get removed from the group (Simulation Case 5).

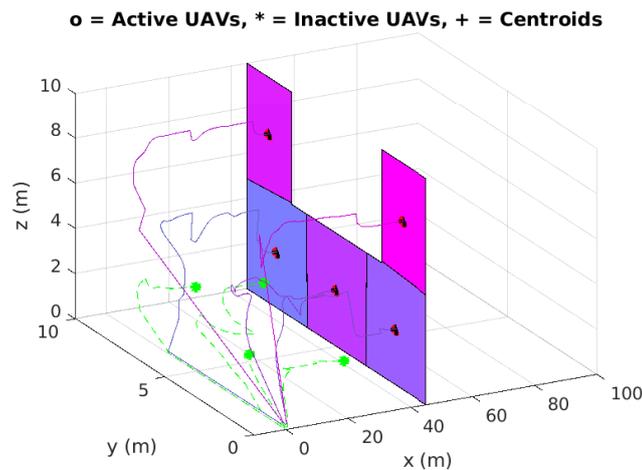


Figure 11. Complete trajectories of the Multi-UAV system showing both active and inactive UAVs (Simulation Case 5).

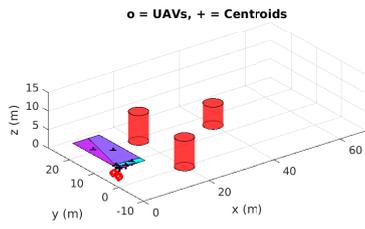
6.3. Simulation Case 6: Obstacle Avoidance

An additional simulation case was performed to show one way of incorporating obstacle avoidance capabilities within the proposed method. The simulation results for this case are given in Figures 12–14. In this case, six vehicles were initially deployed, and a rectangular region was considered to be the sweeping plane $\mathcal{F}(t)$. This particular choice of $\mathcal{F}(t)$ can be used in scenarios where the vehicles are equipped with some downward-facing sensors (ex. cameras), and their sensing FOV are simply a footprint on the ground. Even though using six vehicles may be redundant in this case as there will be overlapping in the vehicles' sensing FOV, the aim here is only to show a particular way for the multi-UAV system to avoid obstacles.

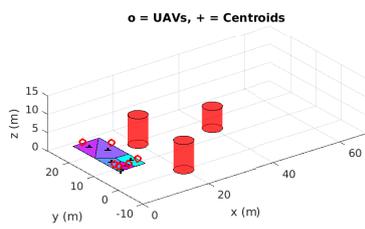
The adopted approach is simply to manipulate the shape, size, orientation and/or velocity of the sweeping plane $\mathcal{F}(t)$ to safely avoid detected obstacles. Note that in this case, only vehicles within a sensing range from obstacles can determine such required action which then sent to the remaining vehicles over the connected network. Obstacle avoidance is ensured in this case because vehicles are guaranteed to maintain their motion within $\mathcal{F}(t)$ once its reached using the proposed control laws (i.e., $p_i(t) \in \mathcal{F}(t)$, $\forall t > t_*$ where t_* is the time at which the vehicles reach $\mathcal{F}(t)$).

It can be seen from Figure 12a,b that the vehicles are approaching $\mathcal{F}(t)$ since they were initially deployed away from it. Simultaneously, the vehicles move along $\mathcal{F}(t)$ to reach centroidal Voronoi configurations. At $t = 15$ s, two obstacles are detected along the way of the sweeping plane's movement by two nearby vehicles. The vehicles decide to dynamically change the size of $\mathcal{F}(t)$ to avoid both obstacles as shown in Figure 12g–j. A different decision is also made by the multi-UAV system when detecting another obstacle at $t = 49.5$ s (Figure 12k). At this time, the sweeping plane is tilted in the z-direction as can be seen from Figure 12l–n.

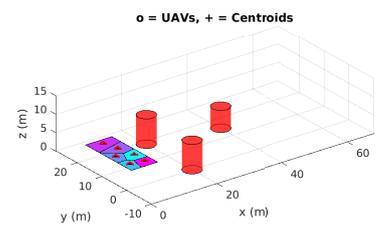
Overall, this approach has good potential for motion coordination of multi-vehicle systems especially when considering obstacle avoidance compared to artificial potential based approaches where sometimes there will be conflicts between achieving obstacle avoidance and avoiding collision with other vehicles.



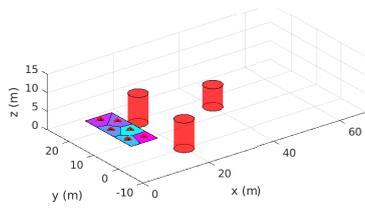
(a) t=1s



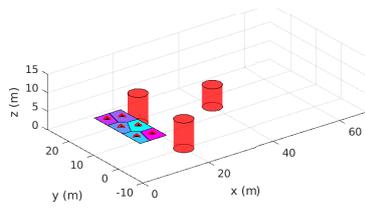
(b) t=1.5s



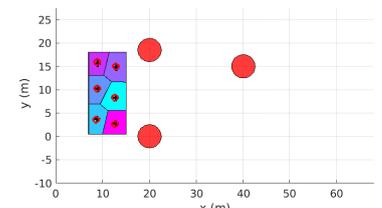
(c) t=4s



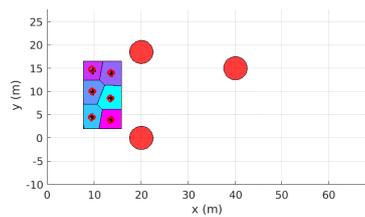
(d) t=9s



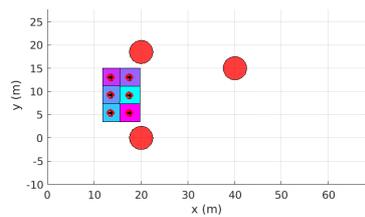
(e) t=14.5s



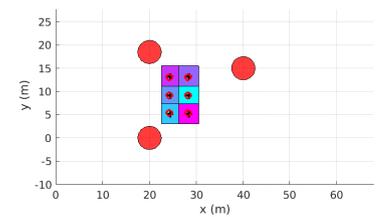
(f) t=15s



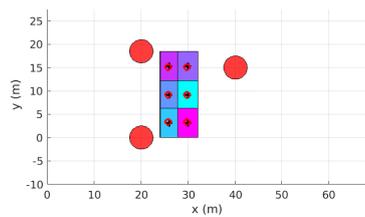
(g) t=16.5s



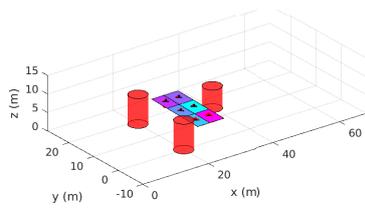
(h) t=24.5s



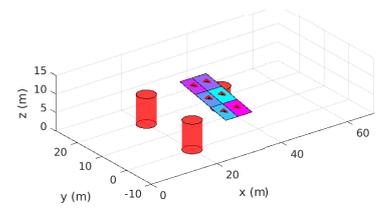
(i) t=46s



(j) t=49s



(k) t=49.5s



(l) t=61.5s

Figure 12. Cont.

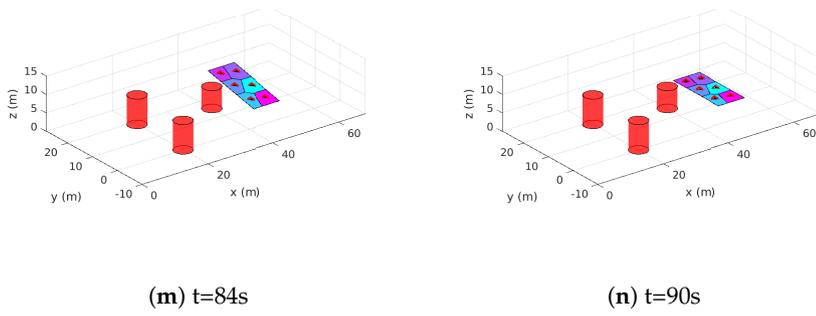


Figure 12. UAV locations at different time instants showing the obstacle avoidance capability of the proposed approach by dynamically manipulating the sweeping plane $\mathcal{F}(t)$ (Simulation Case 6).

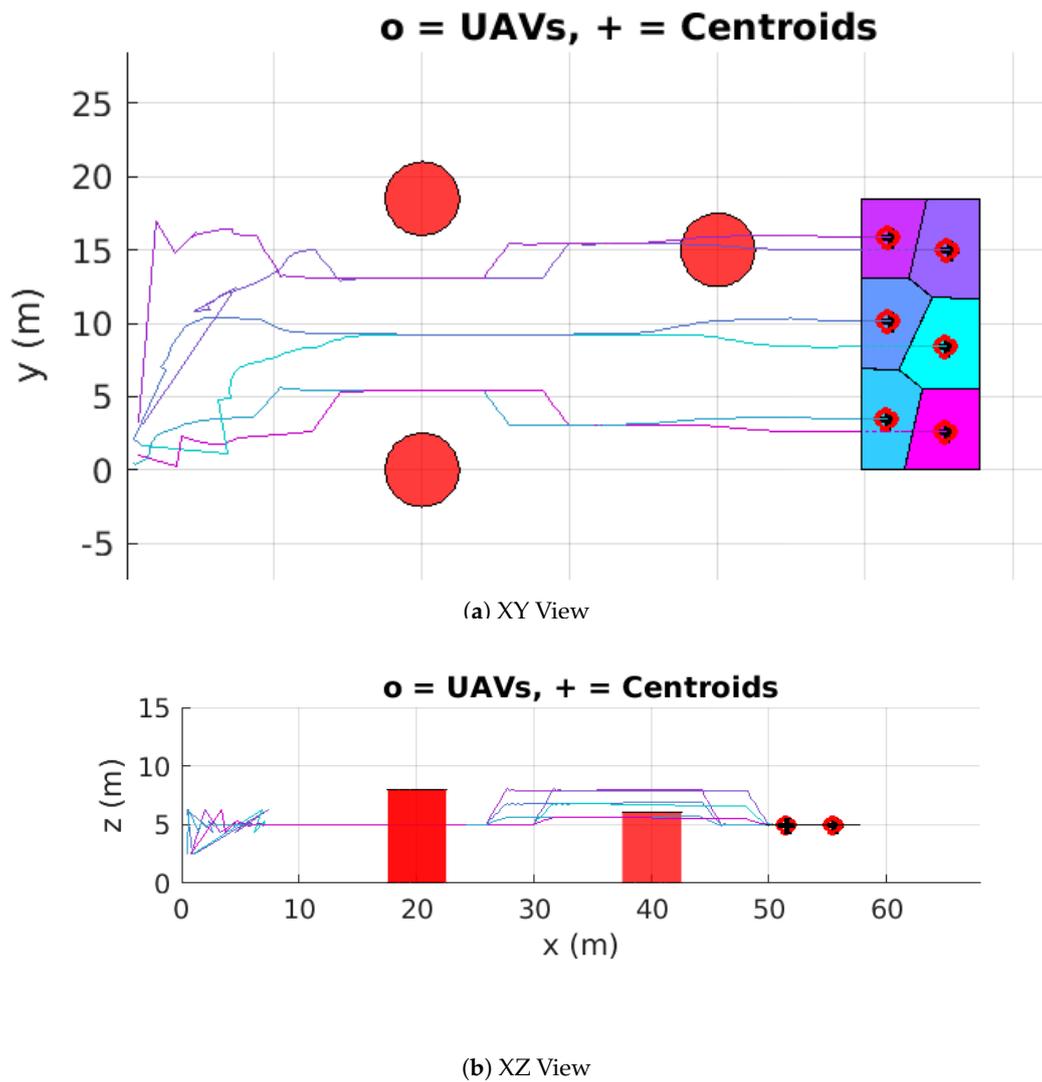


Figure 13. Complete trajectories of the Multi-UAV system showing the obstacle avoidance capability of the proposed approach [Planar Views] (Simulation Case 6).

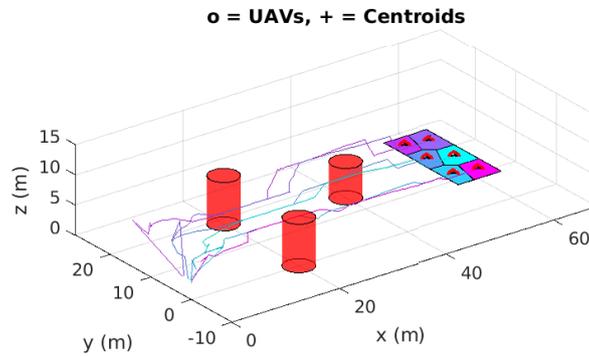


Figure 14. Complete trajectories of the Multi-UAV system showing the obstacle avoidance capability of the proposed approach [3D View] (Simulation Case 6).

7. Generalized Multi-Region Approach & 3D Flocking

7.1. Approach

The proposed strategy constraints the vehicles' movement within a desired plane. This can be sufficient for sweep coverage problems as the region's movement will ensure that the 3D environment is covered. However, when considering flocking problems and swarms with large number of vehicles, it becomes limiting to constrain the movement of all vehicles within a planar region. Thus, this section generalizes the proposed method by introducing the use of multi-regions.

Let $F(t) = \{\mathcal{F}_1(t), \dots, \mathcal{F}_m(t)\}$ be a list of m 3D polygonal regions defined by their vertices such that the following condition is satisfied:

$$\mathcal{F}_i(t) \cap \mathcal{F}_j(t) = \emptyset, \quad i, j \in \{1, \dots, m\}, \quad i \neq j, \quad t \geq 0. \quad (35)$$

Each region has dynamics governed by:

$$\dot{r}_i(t) = g_i(t, r_i), \quad i = \{1, \dots, m\}, \quad (36)$$

where r_i and $g_i(t, r_i)$ are as defined in (25), and $g_i(t, r_i)$ is subject to the same constraint in (26). Furthermore, let $Q_{max} \in \mathbb{R}^m$ be a vector representing the maximum capacity for a region (i.e., the maximum number of vehicles it can include without violating the safety condition). Additionally, define $Q(T) \in \mathbb{R}^m$ as a vector of the instantaneous number of vehicles allocated for each region.

The proposed approach using Centroidal Voronoi tessellations along with the control law (16) (or (17)) can guarantee collision avoidance among vehicles moving towards the same region. However, the control law (16) (or (17)) need to be modified to ensure that vehicles can avoid collisions when moving towards centroidal Voronoi tessellations of different regions. Thus, the following control law is proposed (based on (17)):

$$\mathbf{u}_i = \mathbf{K}_i \tanh\left(\gamma_i(\tilde{\mathbf{C}}_{V_i} - \mathbf{p}_i)\right) + \sum_{j \in \mathcal{N}_i} \mathbf{K}_{ij} \tanh\left(\gamma_i \max(0, d_c^2 - \|\mathbf{p}_i - \mathbf{p}_j\|^2)\right) \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|}, \quad (37)$$

where \mathbf{K}_{ij} is a positive semi-definite diagonal matrix, and $d_c > 0$ is some threshold distance at which the repelling force (the second component of (37)) is applied. Once all vehicles converge to their associated regions, this component vanishes given that the regions design is feasible. Note that the control law (37) allows us to tune the gains \mathbf{K}_i and \mathbf{K}_{ij} while ensuring that the condition (11) is satisfied.

To maintain the distributed nature of the overall strategy, it is assumed that $F(t)$ is shared among the connected networked vehicles along with the regions' dynamical model, the maximum capacity vector Q_{max} and the vector $Q(t)$. Thus, whenever a vehicle decides to pick a region $\mathcal{F}_i(t)$, it can choose the closest one where $Q_i(t) < Q_{max}$.

7.2. Simulations

Several simulations have been carried to verify the generalized approach presented in this section. The simulation cases show how the swarm converges to the desired structure and how the flocking objective can be achieved. Different swarm sizes, desired structures and movement patterns have been considered.

The obtained results for four cases are shown in Figures 15–18. In all cases, the vehicles are initially deployed to random locations. Figures 15–18a show the trajectories taken by all vehicles, which verify that the swarm can converge to and hold the desired structure while following the desired movement pattern. Figures 15–18b show the multi-regions considered and the final locations of the vehicles. Notice that there are no constraints on the shapes of the regions except that they should not intersect according to (35).

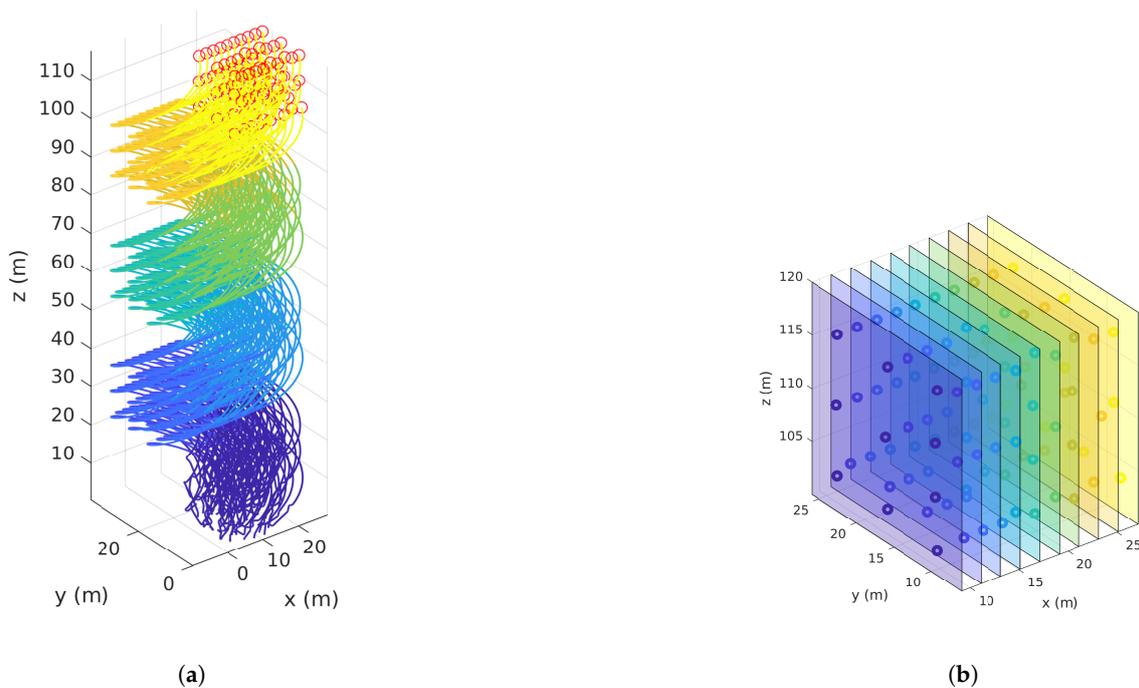
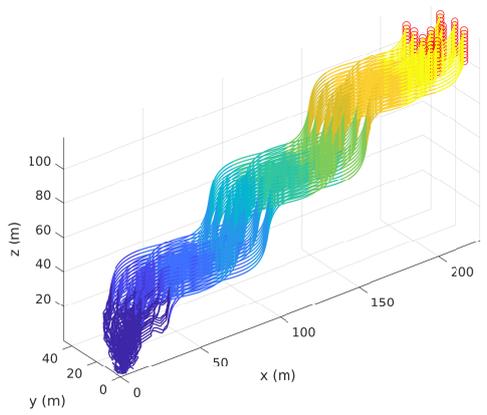
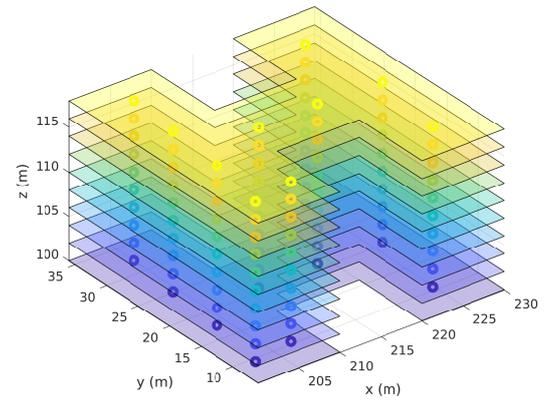


Figure 15. Multi-Region simulation case 1 ($n = 100$ and $m = 10$).

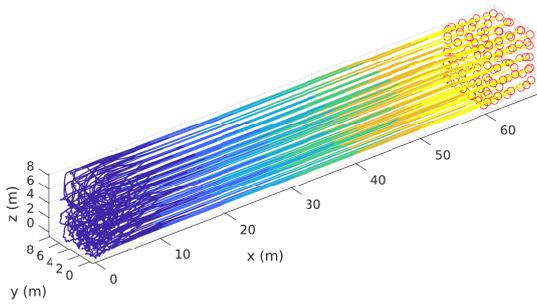


(a)

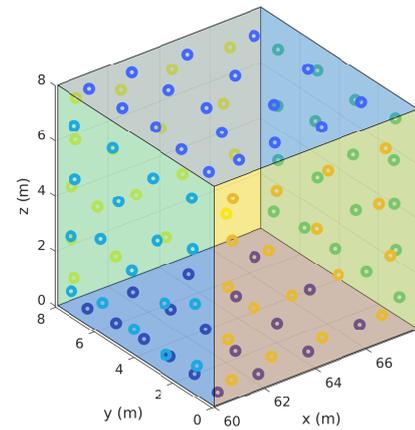


(b)

Figure 16. Multi-Region simulation case 2 ($n = 100$ and $m = 10$).

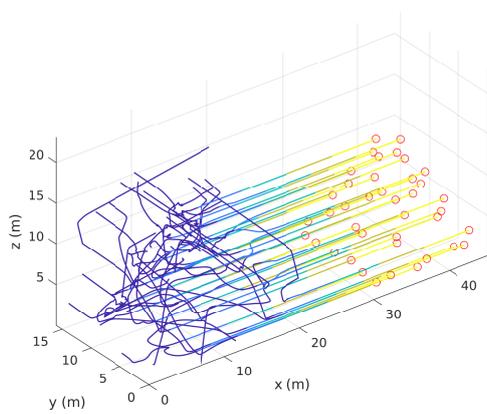


(a)

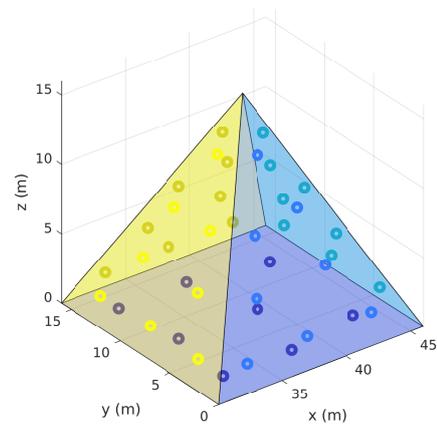


(b)

Figure 17. Multi-Region simulation case 3 ($n = 108$ and $m = 6$).



(a)



(b)

Figure 18. Multi-Region simulation case 1 ($n = 40$ and $m = 5$).

8. Implementation Using a Multi-Quadrotor System

The proposed control methods are developed based on the general kinematic model in (10) which is applicable to different UAV types and AUVs. An example way in implementing the sweep coverage control method is presented in this section for a particular application in precision agriculture using a group of quadrotor UAVs.

8.1. Quadrotor Dynamics

Now, we will extend the model in (10) to include the dynamics of quadrotor-type UAVs according to the following model:

$$\dot{\boldsymbol{p}}_i = \boldsymbol{v}_i \quad (38)$$

$$\dot{\boldsymbol{v}}_i = -g\boldsymbol{e}_3 + \frac{1}{m_i} T_i \boldsymbol{R}_i \boldsymbol{e}_3 \quad (39)$$

$$\dot{\boldsymbol{R}}_i = \boldsymbol{R}_i \text{skew}(\boldsymbol{\Omega}_i) \quad (40)$$

$$\boldsymbol{I}_i \dot{\boldsymbol{\Omega}}_i = -\boldsymbol{\Omega}_i \times \boldsymbol{I}_i \boldsymbol{\Omega}_i + \boldsymbol{\tau}_i \quad (41)$$

where the velocity vector \boldsymbol{u}_i is replaced by \boldsymbol{v}_i for convenience, g is the gravitational acceleration, $\boldsymbol{e}_3 = [1 \ 1 \ 1]^T$, m_i is the UAV mass, and \boldsymbol{I}_i is the inertia matrix. Recall that the vehicle's position and velocities are expressed in the inertial coordinate frame $\{\mathcal{I}\}$. Consider also another reference frame attached to the UAV body with its origin at the center of mass. This frame is referred to as the body-fixed frame $\{B^i\}$. The orientation of the UAV is represented using a rotation matrix \boldsymbol{R}_i between $\{B^i\}$ and $\{\mathcal{I}\}$. The rate of change in the vehicle's orientation is denoted by $\boldsymbol{\Omega}_i$ (i.e., angular velocity) which is expressed in the $\{B^i\}$ frame. The control inputs for this model are the collective thrust $T_i \in \mathbb{R}$ and the body-torques vector $\boldsymbol{\tau}_i \in \mathbb{R}^3$. For more details about quadrotors modelling, the reader is referred to [84,85].

8.2. Tracking Control

There are different ways to implement the proposed sweeping control strategy. One possible approach is to directly couple the velocity commands in (16) (or (17)) into attitude and thrust control inputs. Another possible direction is to use the determined centroidal Voronoi configurations by the algorithm S1-S4 as goal positions with a position tracking controller. The latter approach is considered here for a control design based on the differential-flatness property of the quadrotor dynamics and the sliding mode control technique.

Let us redefine the position and velocity tracking errors as follows:

$$\boldsymbol{e}_{\boldsymbol{p},i} = \boldsymbol{p}_i - \boldsymbol{c}_{V_i} \quad (42)$$

$$\boldsymbol{e}_{\boldsymbol{v},i} = \boldsymbol{v}_i - \dot{\boldsymbol{c}}_{V_i} \quad (43)$$

consider sliding surfaces $\boldsymbol{\sigma}_i$ such that:

$$\boldsymbol{\sigma}_i = \boldsymbol{e}_{\boldsymbol{v},i} + \boldsymbol{L}_{1,i} \tanh(\mu_1 \boldsymbol{e}_{\boldsymbol{p},i}), \quad (44)$$

where $\mu_1 > 0$, and $\boldsymbol{L}_{1,i}$ is a positive definite diagonal matrix. Note that this choice will ensure that the velocities can remain bounded by $\lambda_{\max}(\boldsymbol{L}_{1,i})$ to account for physical limits.

A desired acceleration command can then be computed using:

$$\begin{aligned} \frac{1}{m_i} \boldsymbol{a}_{des,i} = & g\boldsymbol{e}_3 + \boldsymbol{L}_{2,i} \tanh(\mu_2 \boldsymbol{\sigma}_i) \\ & + \sum_{j \in \mathcal{N}_i} \boldsymbol{L}_{ij} \max(0, d_c^2 - \|\boldsymbol{p}_i - \boldsymbol{p}_j\|^2) \frac{\boldsymbol{p}_i - \boldsymbol{p}_j}{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|}, \end{aligned} \quad (45)$$

where $L_{2,i}$ and L_{ij} are positive definite diagonal matrices. The additional term in (45) provides more safety as it represents a repelling force from vehicles closer than some critical distance $d_c > 0$ in the neighbourhood of vehicle i .

The differential-flatness property is then utilized to achieve the acceleration command in (45) while maintaining some desired yaw angle $\psi_{ref,i}$ using the following equations:

$$T_i = \mathbf{a}_{des,i}^T \mathbf{R}_i \mathbf{e}_3 \quad (46)$$

$$\mathbf{R}_{des,i} = [\mathbf{x}_{\mathcal{B}^i,des}^i, \mathbf{y}_{\mathcal{B}^i,des}^i, \mathbf{z}_{\mathcal{B}^i,des}^i] \quad (47)$$

$$\mathbf{z}_{\mathcal{B}^i,des}^i = \frac{\mathbf{a}_{des,i}}{\|\mathbf{a}_{des,i}\|} \quad (48)$$

$$\mathbf{x}_{\mathcal{B}^i,des}^i = \frac{\mathbf{y}_{\mathcal{C}^i}^i \times \mathbf{z}_{\mathcal{B}^i,des}^i}{\|\mathbf{y}_{\mathcal{C}^i}^i \times \mathbf{z}_{\mathcal{B}^i,des}^i\|} \quad (49)$$

$$\mathbf{y}_{\mathcal{B}^i,des}^i = \mathbf{z}_{\mathcal{B}^i,des}^i \times \mathbf{x}_{\mathcal{B}^i,des}^i \quad (50)$$

$$\mathbf{y}_{\mathcal{C}^i}^i = [-\sin(\psi_{ref,i}) \cos(\psi_{ref,i}) 0]^T. \quad (51)$$

A low-level control is then used to compute τ_i to ensure that the vehicle can achieve the desired attitude $\mathbf{R}_{des,i}$. Further details about the differential-flatness property of quadrotor dynamics can be found in [85,86].

8.3. Software-in-the-Loop Simulations

The performance of the suggested sweeping coverage control for multi-quadrotor systems was evaluated using Software-in-the-Loop (SITL) simulations using *Gazebo* and the *Robot Operating System (ROS)* framework. This allows us to test the computational performance of the production code which can be used directly on the real vehicle's onboard computer to implement our algorithms and control strategies in real-time. A simple scenario was considered where a group of three quadrotors were needed to survey a crops field $\mathcal{S} = \{x, y, z \in \mathbb{R} : 0 \leq x \leq 32, 0 \leq y \leq 40, z = 0\}$. Each vehicle is assumed to have an onboard camera providing a sensing footprint $\mathcal{A}_i \subset \mathcal{S}$ dependent on its characteristics and the UAV's altitude. For simplicity, an obstacle-free environment was used as shown in Figure 19.

We used the open-source PX4 flight stack as an implementation for the low-level controller and an extended Kalman filter for states estimation considering noisy measurements. Our tracking control logic was implemented using C++ providing thrust and attitude inputs at a rate of 100Hz. Furthermore, centroidal Voronoi configurations computations were implemented using Python utilizing some of the available tools from the "scipy.spatial" Python module to determine Voronoi cells as a convex polygon in accordance with Assumption 1. Furthermore, the algorithm in S1–S4 was used to obtain the centroids. This turned out to be very computationally efficient since the computations rely on closed-form expressions.

In order to achieve the coverage task in hand, an initial rectangular sweeping plane was selected as: $\mathcal{F}(0) = \{x, y, z \in \mathbb{R} : 0 \leq x \leq 8, 0 \leq y \leq 2, z = 2\}$ where it was required for the UAVs to fly at a fixed altitude of 2m. The classical lawn-mower pattern was considered to generate the trajectory of $\mathcal{F}(t)$ to completely scan \mathcal{S} while keeping the yaw angle at $\psi_{ref} = 0$ without loss of generality. The following parameters were used in the simulation: $L_{1,i} = \text{diag}\{2, 2.5, 3\}$, $L_{2,i} = \text{diag}\{2.5, 3, 4\}$, $L_{ij} = \text{diag}\{1, 1, 1\}$, $d_c = 0.5$ m, $\mu_1 = 2$ and $\mu_2 = 1.5$ (where $\text{diag}\{\cdot\}$ represents a diagonal matrix).

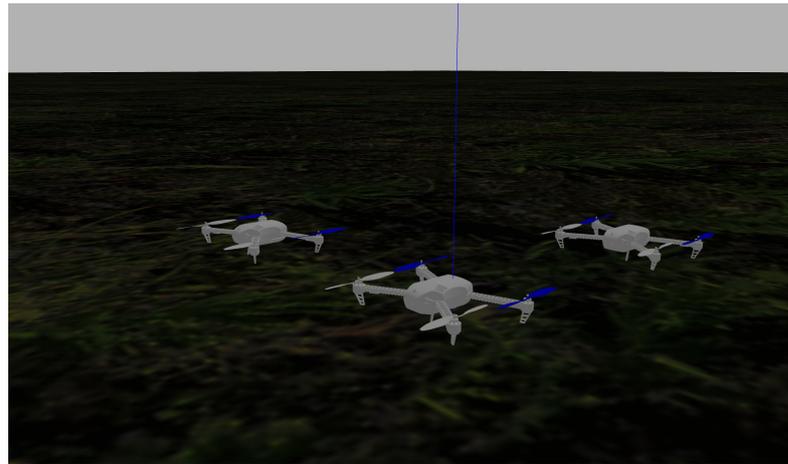


Figure 19. UAVs and environment used in simulations.

During the simulation, all measurements and data were captured as a bag file using available ROS recording tools. The results are plotted using MATLAB which are shown in Figures 20–25. In Figure 20, coordinates of all vehicles are plotted with respect to time. The overall paths taken by the vehicles are shown in Figures 21 where the highlighted rectangular area represents the crops field to be surveyed (i.e., \mathcal{S}). Note that in this time, the vehicles were moving along some other region \mathcal{S} whose projection on the ground floor is \mathcal{S} . Thus, the trajectory of $\mathcal{F}(t)$ was designed to cover \mathcal{S} given that the equipped sensors have a collective footprint which ensures the optimal coverage of \mathcal{S} . It is clear from these results that the overall motion is collision-free, and the lawn-mower pattern decided for $\mathcal{F}(t)$ was followed by the whole group. Velocity norms are shown in Figure 23. Overall the vehicles are moving with a constant speed $g_0 = 1$ m/s except for some changes in velocity at times where the vehicles are turning ($t \approx 60$ s, 110 s, 160 s) in order to avoid collisions with each others. Overall, velocities remain within the desired limits of $\|v\| \leq 3.5$ m/s except at the start when vehicles were taking off which is based on a different control law. The high-level control commands, namely roll, pitch, yaw and thrust, are shown in Figures 24 and 25 where the thrust is normalized in the range $[0, 1]$. This simulation case shows that our control can be successfully implemented using multi-quadrotor systems.

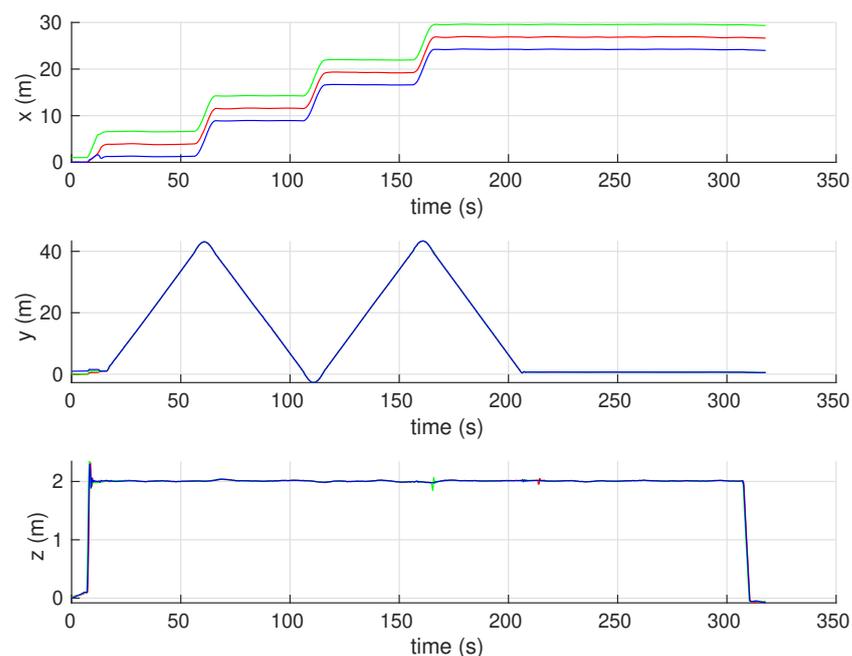


Figure 20. UAVs coordinates with respect to time.

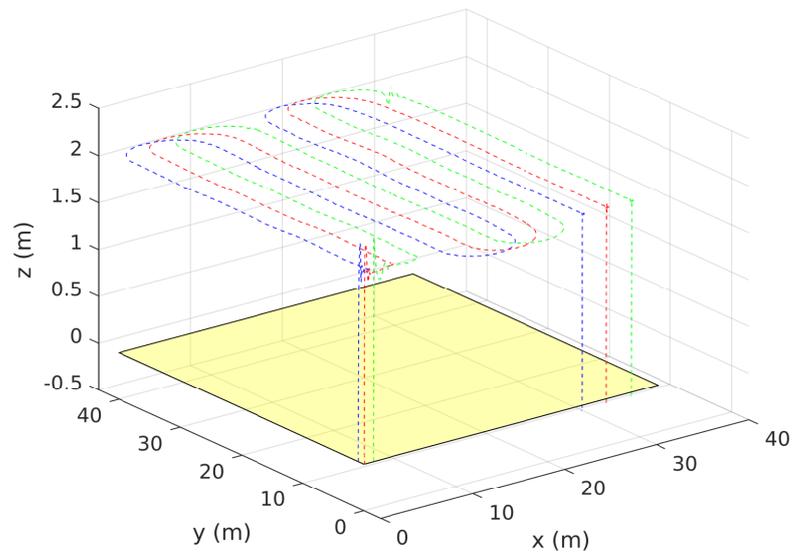


Figure 21. UAVs actual trajectories (3D prospective).

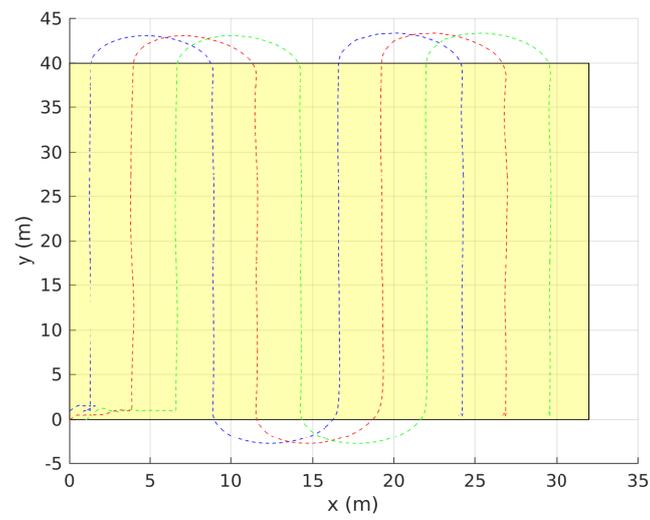


Figure 22. UAVs actual trajectories (2D prospective).

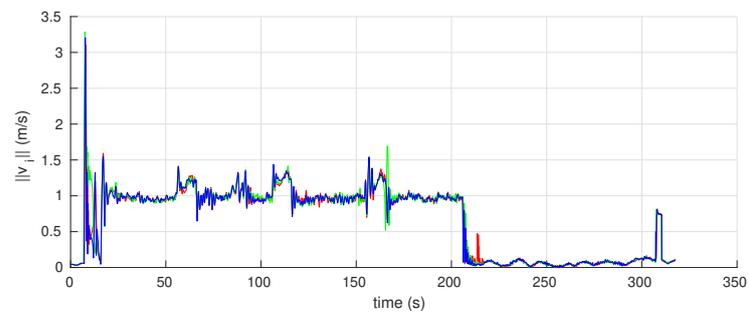


Figure 23. UAVs linear velocities with respect to time.

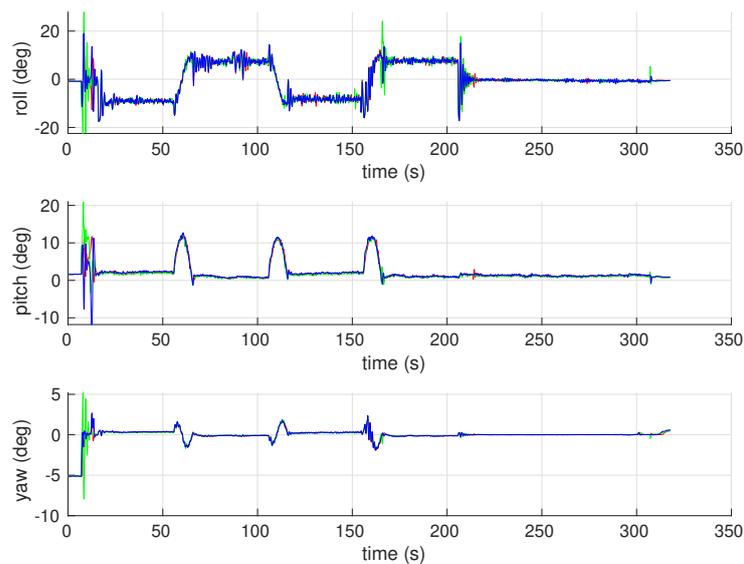


Figure 24. UAVs attitude (roll, pitch and yaw) with respect to time.

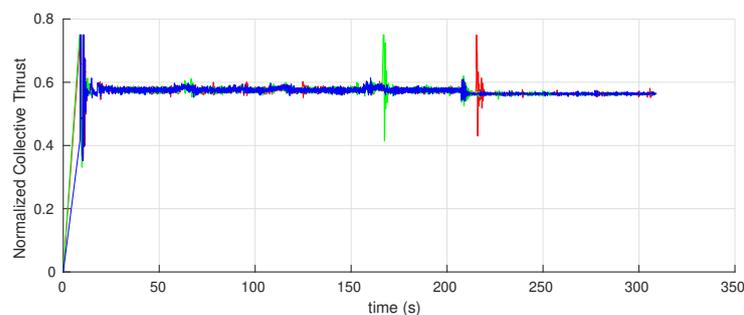


Figure 25. UAVs normalized collective thrust input with respect to time.

9. Conclusions & Future Work

Distributed cooperative control methods were proposed in this paper for multi-UAV systems to address coverage problems in 3D spaces including barrier and sweeping problems. Additionally, a generalization of these approaches was given to address flocking problems. The development of control laws was based on general kinematic model which make them applicable to different UAV types and autonomous underwater vehicles as well. A region-based control mechanism was adopted to constrain the movement of the vehicles within some desired region providing a computationally efficient solution that is both robust and scalable. Mathematical analysis was performed to ensure the stability of the developed control laws. Several simulations have been performed confirming the performance of the developed methods for 3D static and dynamic coverage problems. Additional simulation cases were also presented to show how well the suggested approach can handle obstacle avoidance as well as being robust against vehicles' failure. To evaluate the computational performance, software-in-the-loop simulations, using Gazebo and ROS, were also carried out considering a special case of using a multi-quadrotor system in precision agriculture. To that end, specific implementation details were also proposed based on quadrotor dynamics. Future work can consider practical implementation and developing proper ways where the vehicles can collaboratively decide the shape of the dynamic region based on the multi-UAV system characteristics and the targeted sensing region size.

Author Contributions: Conceptualization, T.E. and A.V.S.; methodology, T.E.; software, T.E.; validation, T.E.; formal analysis, T.E.; resources, A.V.S.; writing—original draft preparation, T.E.; writing—

review and editing, A.V.S.; visualization, T.E.; supervision, A.V.S.; project administration, A.V.S.; funding acquisition, A.V.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Australian Research Council. Furthermore, this work received funding from the Australian Government, via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cortes, J.; Martinez, S.; Karatas, T.; Bullo, F. Coverage control for mobile sensing networks. *IEEE Trans. Robot. Autom.* **2004**, *20*, 243–255. [[CrossRef](#)]
2. Hussein, I.I.; Stipanovic, D.M. Effective coverage control for mobile sensor networks with guaranteed collision avoidance. *IEEE Trans. Control. Syst. Technol.* **2007**, *15*, 642–657. [[CrossRef](#)]
3. Pimenta, L.C.A.; Kumar, V.; Mesquita, R.C.; Pereira, G.A.S. Sensing and coverage for a network of heterogeneous robots. In Proceedings of the 2008 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; IEEE: Manhattan, NY, USA, 2008; pp. 3947–3952.
4. Cheng, T.M.; Savkin, A.V. A distributed self-deployment algorithm for the coverage of mobile wireless sensor networks. *IEEE Commun. Lett.* **2009**, *13*, 877–879. [[CrossRef](#)]
5. Schwager, M.; Rus, D.; Slotine, J.J. Decentralized, adaptive coverage control for networked robots. *Int. J. Robot. Res.* **2009**, *28*, 357–375. [[CrossRef](#)]
6. Cheng, T.M.; Savkin, A.V. Decentralized control for mobile robotic sensor network self-deployment: Barrier and sweep coverage problems. *Robotica* **2011**, *29*, 283–294. [[CrossRef](#)]
7. Stergiopoulos, Y.; Thanou, M.; Tzes, A. Distributed collaborative coverage-control schemes for non-convex domains. *IEEE Trans. Autom. Control.* **2015**, *60*, 2422–2427. [[CrossRef](#)]
8. Savkin, A.V.; Cheng, T.M.; Xi, Z.; Javed, F.; Matveev, A.S.; Nguyen, H. *Decentralized Coverage Control Problems for Mobile Robotic Sensor and Actuator Networks*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
9. Chao, H.; Baumann, M.; Jensen, A.; Chen, Y.; Cao, Y.; Ren, W.; McKee, M. Band-reconfigurable multi-UAV-based cooperative remote sensing for real-time water management and distributed irrigation control. *IFAC Proc. Vol.* **2008**, *41*, 11744–11749. [[CrossRef](#)]
10. Hu, J.; Yang, J. Application of distributed auction to multi-UAV task assignment in agriculture. *Int. J. Precis. Agric. Aviat.* **2018**, *1*. [[CrossRef](#)]
11. Ju, C.; Son, H. Multiple UAV systems for agricultural applications: Control, implementation, and evaluation. *Electronics* **2018**, *7*, 162. [[CrossRef](#)]
12. Maes, W.H.; Steppe, K. Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture. *Trends Plant Sci.* **2019**, *24*, 152–164. [[CrossRef](#)] [[PubMed](#)]
13. Hegde, A.; Ghose, D. Multi-UAV Distributed Control for Load Transportation in Precision Agriculture. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; p. 2068.
14. Bernard, M.; Kondak, K.; Maza, I.; Ollero, A. Autonomous transportation and deployment with aerial robots for search and rescue missions. *J. Field Robot.* **2011**, *28*, 914–931. [[CrossRef](#)]
15. Michael, N.; Fink, J.; Kumar, V. Cooperative manipulation and transportation with aerial robots. *Auton. Robot.* **2011**, *30*, 73–86. [[CrossRef](#)]
16. Fink, J.; Michael, N.; Kim, S.; Kumar, V. Planning and control for cooperative manipulation and transportation with aerial robots. *Int. J. Robot. Res.* **2011**, *30*, 324–334. [[CrossRef](#)]
17. Sreenath, K.; Kumar, V. Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots. In Proceedings of the Robotics: Science and Systems IX, Berlin, Germany, 24–28 June 2013.
18. Ruggiero, F.; Lippiello, V.; Ollero, A. Aerial manipulation: A literature review. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1957–1964. [[CrossRef](#)]
19. Arnold, R.D.; Yamaguchi, H.; Tanaka, T. Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *J. Int. Humanit. Action* **2018**, *3*, 1–18. [[CrossRef](#)]
20. Hayat, S.; Yanmaz, E.; Bettstetter, C.; Brown, T.X. Multi-objective drone path planning for search and rescue with quality-of-service requirements. *Auton. Robot.* **2020**, *44*, 1183–1198. [[CrossRef](#)]
21. Ausonio, E.; Bagnnerini, P.; Ghio, M. Drone Swarms in Fire Suppression Activities: A Conceptual Framework. *Drones* **2021**, *5*, 17. [[CrossRef](#)]
22. Li, X.; Savkin, A.V. Networked Unmanned Aerial Vehicles for Surveillance and Monitoring: A Survey. *Future Internet* **2021**, *13*, 174. [[CrossRef](#)]
23. Xu, C.; Zhang, K.; Jiang, Y.; Niu, S.; Yang, T.; Song, H. Communication Aware UAV Swarm Surveillance Based on Hierarchical Architecture. *Drones* **2021**, *5*, 33. [[CrossRef](#)]
24. Cole, D.T.; Thompson, P.; Göktoğan, A.H.; Sukkarieh, S. System development and demonstration of a cooperative UAV team for mapping and tracking. *Int. J. Robot. Res.* **2010**, *29*, 1371–1399. [[CrossRef](#)]

25. Hu, J.; Xu, J.; Xie, L. Cooperative search and exploration in robotic networks. *Unmanned Syst.* **2013**, *1*, 121–142. [[CrossRef](#)]
26. Mahdoui, N.; Frémont, V.; Natalizio, E. Communicating Multi-UAV System for cooperative SLAM-based exploration. *J. Intell. Robot. Syst.* **2020**, *98*, 325–343. [[CrossRef](#)]
27. Gage, D.W. Command control for many-robot systems. *Unmanned Syst.* **1992**, *10*, 28–34.
Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*; ACM: New York, NY, USA, 1987; Volume 21, pp. 25–34.
28. Atınc, G.M.; Stipanović, D.M.; Voulgaris, P.G. A swarm-based approach to dynamic coverage control of multi-agent systems. *Automatica* **2020**, *112*, 108637. [[CrossRef](#)]
29. Olfati-Saber, R. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control.* **2006**, *51*, 401–420. [[CrossRef](#)]
30. Reynolds, C.W. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH Computer Graphics*; ACM: New York, NY, USA, 1987; Volume 21, pp. 25–34.
31. Cortes, J.; Martinez, S.; Bullo, F. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM Control. Optim. Calc. Vars.* **2005**, *11*, 691–719. [[CrossRef](#)]
32. Barr, S.J.; Wang, J.; Liu, B. An Efficient Method for Constructing Underwater Sensor Barriers. *J. Commun.* **2011**, *6*, 370–383. [[CrossRef](#)]
33. Petersen, I.R.; Savkin, A.V. *Robust Kalman Filtering for Signals and Systems with Large Uncertainties*; Birkhauser: Boston, MA, USA, 1999.
34. Tipsuwan, Y.; Chow, M.Y. Control methodologies in networked control systems. *Control. Eng. Pract.* **2003**, *11*, 1099–1111. [[CrossRef](#)]
35. Hespanha, J.P.; Naghshtabrizi, P.; Xu, Y. A survey of recent results in networked control systems. *Proc. IEEE* **2007**, *95*, 138–162. [[CrossRef](#)]
36. Wang, F.Y.; Liu, D. *Networked control systems: Theory and Applications*; Springer: London, UK, 2008.
37. Matveev, A.S.; Savkin, A.V. *Estimation and Control over Communication Networks*; Birkhauser Boston: Cambridge, MA, USA, 2009.
38. Bemporad, A.; Heemels, M.; Johansson, M.; others. *Networked Control Systems*; Springer: New York, NY, USA, 2010; Volume 406.
39. Ge, X.; Yang, F.; Han, Q.L. Distributed networked control systems: A brief overview. *Inf. Sci.* **2017**, *380*, 117–131. [[CrossRef](#)]
40. Davoli, L.; Pagliari, E.; Ferrari, G. Hybrid LoRa-IEEE 802.11s Opportunistic Mesh Networking for Flexible UAV Swarming. *Drones* **2021**, *5*, 26. [[CrossRef](#)]
41. Cheah, C.C.; Hou, S.P.; Slotine, J.J.E. Region-based shape control for a swarm of robots. *Automatica* **2009**, *45*, 2406–2411. [[CrossRef](#)]
42. Huang, S.; Teo, R.S.H.; Leong, W.W.L.; Martinel, N.; Forest, G.L.; Micheloni, C. Coverage control of multiple unmanned aerial vehicles: A short review. *Unmanned Syst.* **2018**, *6*, 131–144. [[CrossRef](#)]
43. Kwok, A.; Martinez, S. Energy-balancing cooperative strategies for sensor deployment. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; IEEE: Manhattan, NY, USA, 2007; pp. 6136–6141.
44. Dieber, B.; Micheloni, C.; Rinner, B. Resource-aware coverage and task assignment in visual sensor networks. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 1424–1437. [[CrossRef](#)]
45. Wang, X.; Han, S.; Wu, Y.; Wang, X. Coverage and energy consumption control in mobile heterogeneous wireless sensor networks. *IEEE Trans. Autom. Control.* **2012**, *58*, 975–988. [[CrossRef](#)]
46. Morsly, Y.; Aouf, N.; Djouadi, M.S.; Richardson, M. Particle swarm optimization inspired probability algorithm for optimal camera network placement. *IEEE Sens. J.* **2011**, *12*, 1402–1412. [[CrossRef](#)]
47. Abo-Zahhad, M.; Ahmed, S.M.; Sabor, N.; Sasaki, S. Rearrangement of mobile wireless sensor nodes for coverage maximization based on immune node deployment algorithm. *Comput. Electr. Eng.* **2015**, *43*, 76–89. [[CrossRef](#)]
48. Wang, H.; Guo, Y. A decentralized control for mobile sensor network effective coverage. In Proceedings of the 2008 7th World Congress on Intelligent Control and Automation, Chongqing, China, 25–27 June 2008; IEEE: Manhattan, NY, USA, 2008; pp. 473–478.
49. Howard, A.; Matarić, M.J.; Sukhatme, G.S. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Distributed Autonomous Robotic Systems 5*; Springer: Tokyo, Japan, 2002; pp. 299–308.
50. Schwager, M.; Vitus, M.P.; Rus, D.; Tomlin, C.J. Robust adaptive coverage for robotic sensor networks. In *Robotics Research*; Springer: New York, NY, USA, 2017; pp. 437–454.
51. Stergiopoulos, Y.; Tzes, A. Autonomous deployment of heterogeneous mobile agents with arbitrarily anisotropic sensing patterns. In Proceedings of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, 3–6 July 2012; IEEE: Manhattan, NY, USA, 2012; pp. 1585–1590.
52. Stergiopoulos, Y.; Tzes, A. Cooperative positioning/orientation control of mobile heterogeneous anisotropic sensor networks for area coverage. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; IEEE: Manhattan, NY, USA, 2014; pp. 1106–1111.
53. Kantaros, Y.; Zavlanos, M.M. Distributed communication-aware coverage control by mobile sensor networks. *Automatica* **2016**, *63*, 209–220. [[CrossRef](#)]
54. Papatheodorou, S.; Tzes, A.; Stergiopoulos, Y. Collaborative visual area coverage. *Robot. Auton. Syst.* **2017**, *92*, 126–138. [[CrossRef](#)]

55. Thanou, M.; Tzes, A. Distributed visibility-based coverage using a swarm of UAVs in known 3D-terrains. In Proceedings of the 2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP), Athens, Greece, 21–23 May 2014; IEEE: Manhattan, NY, USA, 2014; pp. 425–428.
56. Hexsel, B.; Chakraborty, N.; Sycara, K. Coverage control for mobile anisotropic sensor networks. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: Manhattan, NY, USA, 2011; pp. 2878–2885.
57. Mohapatra, S.K.; Sahoo, P.K.; Wu, S.L. Big data analytic architecture for intruder detection in heterogeneous wireless sensor networks. *J. Netw. Comput. Appl.* **2016**, *66*, 236–249. [[CrossRef](#)]
58. Saeed, A.; Abdelkader, A.; Khan, M.; Neishaboori, A.; Harras, K.A.; Mohamed, A. Argus: Realistic target coverage by drones. In Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, Pittsburgh, PA, USA, 18–21 April 2017; IEEE: Manhattan, NY, USA, 2017; pp. 155–166.
59. Bullo, F.; Cortés, J.; Martinez, S. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*; Princeton University Press: Princeton, NJ, USA, 2009.
60. Bhattacharya, S.; Ghrist, R.; Kumar, V. Multi-robot coverage and exploration on Riemannian manifolds with boundaries. *Int. J. Robot. Res.* **2014**, *33*, 113–137. [[CrossRef](#)]
61. Panagou, D.; Stipanović, D.M.; Voulgaris, P.G. Vision-based dynamic coverage control for nonholonomic agents. In Proceedings of the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December 2014; IEEE: Manhattan, NY, USA, 2014; pp. 2198–2203.
62. Panagou, D.; Stipanović, D.M.; Voulgaris, P.G. Distributed dynamic coverage and avoidance control under anisotropic sensing. *IEEE Trans. Control. Netw. Syst.* **2016**, *4*, 850–862. [[CrossRef](#)]
63. Li, W.T.; Liu, Y.C. Dynamic coverage control for mobile robot network with limited and nonidentical sensory ranges. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; IEEE: Manhattan, NY, USA, 2017; pp. 775–780.
64. Zuo, L.; Shi, Y.; Yan, W. Dynamic coverage control in a time-varying environment using Bayesian prediction. *IEEE Trans. Cybern.* **2017**, *49*, 354–362. [[CrossRef](#)] [[PubMed](#)]
65. Song, C.; Liu, L.; Feng, G.; Wang, Y.; Gao, Q. Persistent awareness coverage control for mobile sensor networks. *Automatica* **2013**, *49*, 1867–1873. [[CrossRef](#)]
66. Bhattacharya, S.; Michael, N.; Kumar, V. Distributed coverage and exploration in unknown non-convex environments. In *Distributed Autonomous Robotic Systems*; Springer: New York, NY, USA, 2013; pp. 61–75.
67. Wang, Y.; Liu, Y.; Guo, Z. Three-dimensional ocean sensor networks: A survey. *J. Ocean. Univ. China* **2012**, *11*, 436–450. [[CrossRef](#)]
68. Bentz, W.; Panagou, D. 3D dynamic coverage and avoidance control in power-constrained UAV surveillance networks. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; IEEE: Manhattan, NY, USA, 2017; pp. 1–10.
69. Bentz, W.; Hoang, T.; Bayasgalan, E.; Panagou, D. Complete 3-D dynamic coverage in energy-constrained multi-UAV sensor networks. *Auton. Robot.* **2018**, *42*, 825–851. [[CrossRef](#)]
70. Pompili, D.; Melodia, T.; Akyildiz, I.F. Deployment analysis in underwater acoustic wireless sensor networks. In Proceedings of the 1st ACM international workshop on Underwater networks, Los Angeles, CA, USA, 25 September 2006; ACM: New York, NY, USA, 2006; pp. 48–55.
71. Stirling, T.; Wischmann, S.; Floreano, D. Energy-efficient indoor search by swarms of simulated flying robots without global information. *Swarm Intell.* **2010**, *4*, 117–143. [[CrossRef](#)]
72. Boufares, N.; Khoufi, I.; Minet, P.; Saidane, L.; Saied, Y.B. Three dimensional mobile wireless sensor networks redeployment based on virtual forces. In Proceedings of the 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), Dubrovnik, Croatia, 24–28 August 2015; IEEE: Manhattan, NY, USA, 2015; pp. 563–568.
73. Nazarzehi, V.; Savkin, A.V. Distributed self-deployment of mobile wireless 3D robotic sensor networks for complete sensing coverage and forming specific shapes. *Robotica* **2018**, *36*, 1–18. [[CrossRef](#)]
74. Tanner, H.G.; Jadbabaie, A.; Pappas, G.J. Flocking in fixed and switching networks. *IEEE Trans. Autom. Control.* **2007**, *52*, 863–868. [[CrossRef](#)]
75. Dimarogonas, D.V.; Kyriakopoulos, K.J. A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems. *Automatica* **2008**, *44*, 2648–2654. [[CrossRef](#)]
76. Savkin, A.V.; Teimoori, H. Decentralized navigation of groups of wheeled mobile robots with limited communication. *IEEE Trans. Robot.* **2010**, *26*, 1099–1104. [[CrossRef](#)]
77. Reyes, L.A.V.; Tanner, H.G. Flocking, formation control, and path following for a group of mobile robots. *IEEE Trans. Control. Syst. Technol.* **2015**, *23*, 1268–1282. [[CrossRef](#)]
78. Khaledyan, M.; Liu, T.; Fernandez-Kim, V.; de Queiroz, M. Flocking and target interception control for formations of nonholonomic kinematic agents. *IEEE Trans. Control. Syst. Technol.* **2019**, *28*, 1603–1610. [[CrossRef](#)]
79. Do, K.D. Flocking for multiple elliptical agents with limited communication ranges. *IEEE Trans. Robot.* **2011**, *27*, 931–942. [[CrossRef](#)]
80. Antonelli, G.; Arrichiello, F.; Chiaverini, S. Flocking for multi-robot systems via the null-space-based behavioral control. *Swarm Intell.* **2010**, *4*, 37. [[CrossRef](#)]

81. Virágh, C.; Vásárhelyi, G.; Tarcai, N.; Szörényi, T.; Somorjai, G.; Nepusz, T.; Vicsek, T. Flocking algorithm for autonomous flying robots. *Bioinspiration Biomim.* **2014**, *9*, 025012. [[CrossRef](#)]
82. Ghapani, S.; Mei, J.; Ren, W.; Song, Y. Fully distributed flocking with a moving leader for Lagrange networks with parametric uncertainties. *Automatica* **2016**, *67*, 67–76. [[CrossRef](#)]
83. Jafari, M.; Xu, H. A biologically-inspired distributed fault tolerant flocking control for multi-agent system in presence of uncertain dynamics and unknown disturbance. *Eng. Appl. Artif. Intell.* **2019**, *79*, 1–12. [[CrossRef](#)]
84. Hamel, T.; Mahony, R.; Lozano, R.; Ostrowski, J. Dynamic modelling and configuration stabilization for an X4-flyer. *IFAC Proc. Vol.* **2002**, *35*, 217–222. [[CrossRef](#)]
85. Mellinger, D.; Kumar, V. Minimum snap trajectory generation and control for quadrotors. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: Manhattan, NY, USA, 2011; pp. 2520–2525.
86. Faessler, M.; Franchi, A.; Scaramuzza, D. Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. *IEEE Robot. Autom. Lett.* **2017**, *3*, 620–626. [[CrossRef](#)]