Swarm Robotic Consensus Strategies for Multi-Target Tracking And Feature Reconstruction

by

Aniket Ravindra Shirsat

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2022 by the
Graduate Supervisory Committee:

Spring Berman, Chair
Hyunglae Lee
Hamid Marvi
Srikanth Saripalli
Lance Gharavi

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

Technological progress in robot sensing, design, and fabrication, and the availability of open source software frameworks such as the Robot Operating System (ROS), are advancing the applications of swarm robotics from toy problems to real-world tasks such as surveillance, precision agriculture, search-and-rescue, and infrastructure inspection. These applications will require the development of robot controllers and system architectures that scale well with the number of robots and that are robust to robot errors and failures. To achieve this, one approach is to design decentralized robot control policies that require only local sensing and local, ad-hoc communication. In particular, stochastic control policies can be designed that are agnostic to individual robot identities and do not require a priori information about the environment or sophisticated computation, sensing, navigation, or communication capabilities.

This dissertation presents novel swarm control strategies with these properties for detecting and mapping static targets, which represent features of interest, in an unknown, bounded, obstacle-free environment. The robots move on a finite spatial grid according to the time-homogeneous transition probabilities of a Discrete-Time Discrete-State (DTDS) Markov chain model, and they exchange information with other robots within their communication range using a consensus (agreement) protocol. This dissertation extend theoretical guarantees on multi-robot consensus over fixed and time-varying communication networks with known connectivity properties to consensus over the networks that have Markovian switching dynamics and no presumed connectivity. This dissertation develops such swarm consensus strategies for detecting a single feature in the environment, tracking multiple features, and reconstructing a discrete distribution of features modeled as an occupancy grid map. The proposed consensus approaches are validated in numerical simulations and in 3D physics-based simulations of quadrotors in Gazebo. The scalability of the proposed

approaches is examined through extensive numerical simulation studies over different swarm populations and environment sizes.

DEDICATION

To my mom, Manasi Shirsat and my late dad, Ravindra Shirsat who have been my pillars throughout my journey.

ACKNOWLEDGMENTS

I would like to like to thank my advisor, Prof. Spring Berman, for her patience, relentless support and motivation throughout my doctorate studies. She has been a great source of inspiration and also one of the true champion of students that I had the honor of conducting my successful research. I would also like to thank my committee members, Prof.Hyunglae Lee, Prof. Hamid Marvi, Prof.Srikanth Saripalli and Prof. Lance Gharavi for providing constructive feedback on this thesis. I would also like to thank Prof. Pavan Turaga for providing me guidance and support on my research and .

I am very grateful to have great colleagues and collaborators: Dr.Karthik Elamvazhuthi, Dr.Shiba Biswal, Dr.Shatadal Mishra; who have provided me with their valuable insights. They have played a pivotal role in shaping my research in Markov chains and aerial robotic swarms. Without them my doctoral journey would have been very difficult.

My journey would be incomplete without the constant and unconditional support of friends: Anandrao Biradar, Shatadal Mishra, Shiba Biswal, Karthik Elamvazhuthi, Zahi Kakish, Varun Nalam and Alison Mirin. It was conversations with all of you that made already confusing journey of doctoral research manageable. I cannot acknowledge all by name but let that not diminish my gratitude for all my friends who started the journey with me for their Master's at ASU.

I would like to thank my late dad, Ravindra Shirsat for his support even during his time of hardship. Finally, I would like to thank my mom, Manasi Shirsat for her love and support. My parents, are the anchor that have always grounded me and without whom I truly could not have reached here. It is to them, that I dedicate this thesis. Many components of this work were supported by Global Security Initiative at Arizona State University and Office of Naval Research.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Technological advances in sensing, robot design and fabrication, and low-power computing, along with the development of open-source software ecosystems, have played a critical role in making robots ubiquitous in day-to-day life. Commercial robots are available for numerous applications, including the automation of household tasks such as vacuuming, autonomous food delivery, healthcare operations such as remote surgery, and assistance in the hospitality industry (e.g., robot concierges). The key enabling factor for their adoption is the underlying progress in the safety associated with these systems. One of the main aspects that has propelled robotics in the last few years is the devastating effects to the industry workforce and healthcare caused by the COVID-19 pandemic Yang *et al.* (2020); Tavakoli *et al.* (2020). This has necessitated a paradigm shift in the application of robots to improve supply chains by causing an increase in possible use cases for robots that were not previously needed.

With the increase in the use of robotic systems, there arises a problem of scalability The way current robotic systems are controlled in the industry is with the use of a centralized *server* or leader that is responsible for making decisions for all the robots that are connected to the network. This system architecture can have an advantage in providing optimal and efficient solutions. However, it lacks the flexibility to accommodate the loss or addition of robot nodes (end users) and is usually not robust to the effect of communication loss to the centralized server. Such a system architecture is not suitable to scale to large numbers of end users without entailing a significant infrastructure cost.

Robotic swarms are designed to be scalable and highly redundant, making them

one of the candidate architectures for large scale robotic systems for scaling up distributed robotic systems in industrial applications. However, such a scalable and redundant system architecture also makes it difficult to control the swarm as a whole. The control difficulty arises primarily from the context of modeling the swarm, formulating scalable swarm control strategies and the objective function that can account the constraints of redundancy and robustness that are inherently associated with swarms. In this dissertation, we develop swarm control approaches using models from stochastic processes and distributed estimation, with provable guarantees on the swarm performance for the application at hand. Specifically, we develop probabilistic consensus strategies under Markovian switching dynamics for search, multi-target tracking and distributed feature reconstruction in swarm robotics.

We first provide a brief overview of the literature on multi-agent systems in Section 1.1.1, followed by an overview on swarm robotics in Section 1.1.2. Then, we provide a review of related work on current swarm robots architectures and platforms in research in Section 1.1.3, and conclude by stating the contributions of this dissertation in Section 1.2.

1.1 Literature Review

1.1.1 Multi-Agent Systems

We consider an agent as any entity that is capable of intelligently and autonomously perceiving its environment through sensors and performing actions based on the sensed environment parameters to reach a predetermined goal Russell and Norvig (2010); Langley *et al.* (2009). A *multi-agent system (MAS)* is composed of many such *agents* interacting with one another under a predetermined scheme to achieve a collective goal. MAS systems are characterized by using concepts from graph theory like Godsil and Royle (2001); Mesbahi and Egerstedt (2010) to study their prop-

erties like stability, convergence guarantees etc. In MAS applications, all agents typically can sense only partial information in their sensing field of view. This can be communicated to all the agents that are direct neighbors, which depends upon the communication network topology which can be static or dynamic along with directed or undirected. There are several schemes available in both the communication topologies like P2P (peer-to-peer), O2M (one-to-many), M2O (many-to-one), M2M (many-to-many) as shown in Figure 1.1-Figure 1.3

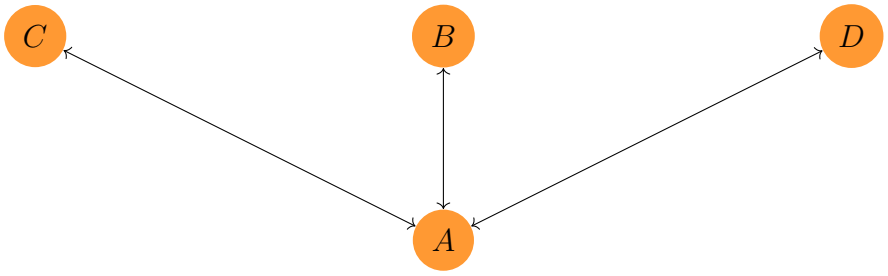


Figure 1.1: Agents Are Represented by A, B, C, D Arranged in One-to-Many and Many-to-One Communication Topology.

For a MAS with *static* topology as shown in Figure 1.4, the communication scheme is fixed and does not change with time. A subset or the complete set of agents communicates with another subset or complete set of agents that comprise the MAS in a fixed and deterministic fashion. Examples of such approaches include leader follower based architectures that are of the form of many-to-one and one-to-many as shown in Figure 1.1. Liu *et al.* (2015) discusses such a scheme that is tolerant to packet dropouts.

MAS with a *dynamic* communication topology as shown in Figure 1.5-Figure 1.7, have an associated dynamics that changes with time or with respect to some parameter that governs the MAS. These MAS can also be classified as switched systems due to nature of their communication topology changing over time. Olfati-Saber and Murray (2004) discusses one application where the MAS has time delays and is modeled with a switching topology. The authors rigorously prove the robustness of such

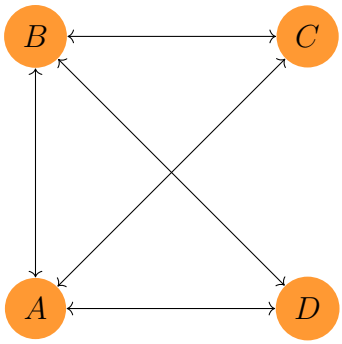


Figure 1.2: Agents Represented by A, B, C, D Arranged in a Many-to-Many Communication Topology.

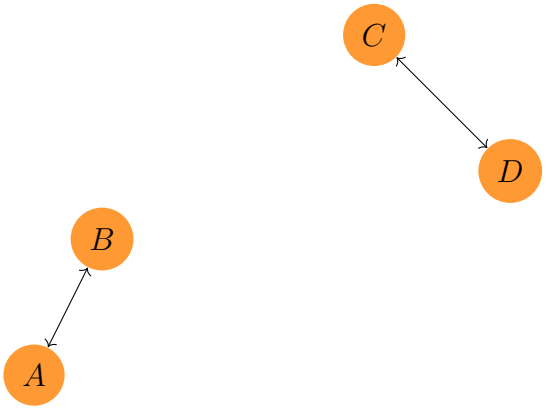


Figure 1.3: Agents Represented by A, B, C, D Arranged in a Peer-to-Peer Communication Topology. Peers Are Identified Depending on Their Proximity to Each Other.

a MAS with associated time delays formulated as a consensus scheme. Such schemes are also useful in the case of MAS where there are bound to exist multiple leader agents and this changes with time. In Liu *et al.* (2014), the authors discuss one such application of discrete time MAS for the case of multiple leaders with time delays.

Leaderless and *leader-follower* approaches are another way MAS systems are conceptualized. *Leader* agent is the one that communicate with all the other agents (followers) in the MAS. *Leader* can be chosen apriori or can be collectively chosen by the agents Song *et al.* (2016). In this configuration, the MAS also becomes a directed network from a graph theoretic perspective, as followers only communicate to the leader and not with other followers, leader communicates only to followers

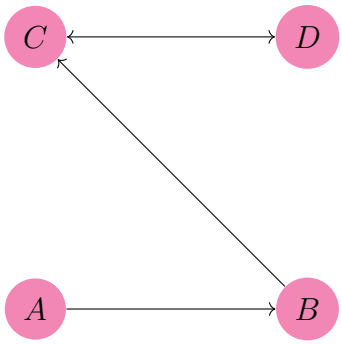
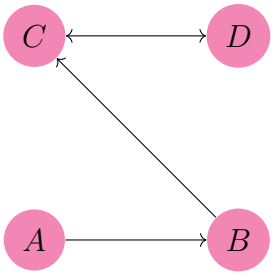


Figure 1.4: Agents A, B, C, D Connected in a Directed and Static Topology. This Communication Topology Is Fixed and Does Not Change With Time. Here A Can Communicate to B, B Can Communicate to C, but Not Vice-Versa. C, D Can Both Communicate With Each Other.



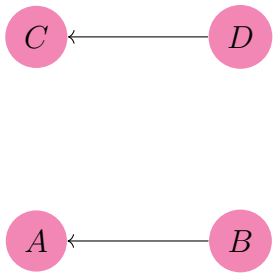
t_1

Figure 1.5: Agents A, B, C, D Change Their Communication Topology Dynamically With Time. At Time t_1, A Communicates With B, B Communicates With C and C and D Can Communicate With Each Other.

and/or other leaders. In Fu and Wang (2014), authors discuss one such application with leader-follower network for switching undirected graphs. Angeli and Bliman (2005) obtain results on the stability of a leaderless MAS with finite and bounded time delays.

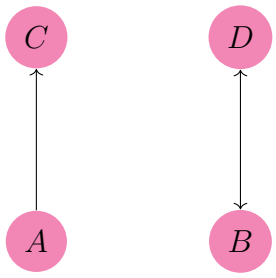
MAS systems can also be *heterogeneous* or *homogeneous*. MAS composed of agents with similar characteristics and capabilities are called *homogeneous* like Vrancken and Soares (2009). *Heterogeneous* MAS are composed of agents that have varying characteristics and capabilities like Khodaverdian (2014).

MAS have applications in several areas like smart grids Merabet *et al.* (2014), computer networks Bajo *et al.* (2016); Wang *et al.* (2018), modeling complex systems



t_2

Figure 1.6: At Time t_2, B Communicates With A, and D Communicates With C.



t_3

Figure 1.7: At Time t_3, A Communicates With C, B and D Can Communicate With Each Other.

Domínguez *et al.* (2015) and robotics Singh *et al.* (2021). These applications also address some of the challenges in MAS like task allocation Dos Santos and Bazzan (2012); Kazakova *et al.* (2020), localization Lin *et al.* (2015); Han *et al.* (2018), formation control Lin *et al.* (2014); Jing and Wang (2019), consensus Li *et al.* (2016); Nowzari *et al.* (2019), learning Hernandez-Leal *et al.* (2019); Zhang *et al.* (2021), mapping Zou *et al.* (2019); Rameau *et al.* (2022).

1.1.2 Swarm Robotics

Swarm robotics (SR) deals with addressing the challenges that arise from large scale interacting or non-interacting robots in particular and complex large systems in general. Some of the solutions are nature inspired like social insects, flocking of

birds, where a collective behavior emerges by following simple local interaction rules Sumpter (2010); Camazine *et al.* (2020). SR has its roots in computational modeling of collective self-organizing behavior. It has also led to the development of several optimization schemes like ant colony optimization Dorigo *et al.* (2006), particle swarm optimization Kennedy and Eberhart (1995). SR also investigated open questions in biological systems like *stigmergy* (a means of indirect communication between agents that leaves a trace in the environment and subsequently stimulate the performance of the same or the other agents), foraging Krieger *et al.* (2000), study mixed robot-social insect societies Halloy *et al.* (2007). For making SR *scalable* and deployable, there were a lot of hurdles from the design and hardware miniaturization perspective that needed to be overcome. Kilobots Rubenstein *et al.* (2012) (Figure 1.8) and Pheeno Wilson *et al.* (2016) as shown in Figure 1.9 are ground based swarm robotic platform and Crazyswarm Preiss *et al.* (2017)(Figure 1.10-Figure 1.11) a nano quadcopter aerial swarm are a few examples of swarm robots that exist in research and education. There are also SR deployed for maritime research. Duarte *et al.* (2016) studied collective behaviors like dispersion, clustering, homing with a fleet of 10 robots as shown in Figure 1.12-Figure 1.13.

For deploying swarms, SR aims at addressing key engineering challenges like *robustness*, *scalability*, *adaptability*. Some of these solutions and strategies are inspired from MAS, and are modified for applications like coordination, control task allocation, localization, etc. *Robustness* in SR refers to the continual operation of the swarm in the presence of partial failure of an agent or agents in the swarm. Swarms are designed to be highly redundant and all agents in the swarm are capable to compensate for the loss of few individual agents. *Scalability* refers to the ability to accommodate large variation in the number of individual agents without causing a significant degradation of swarm performance. It is facilitated by using rules inspired from swarms in

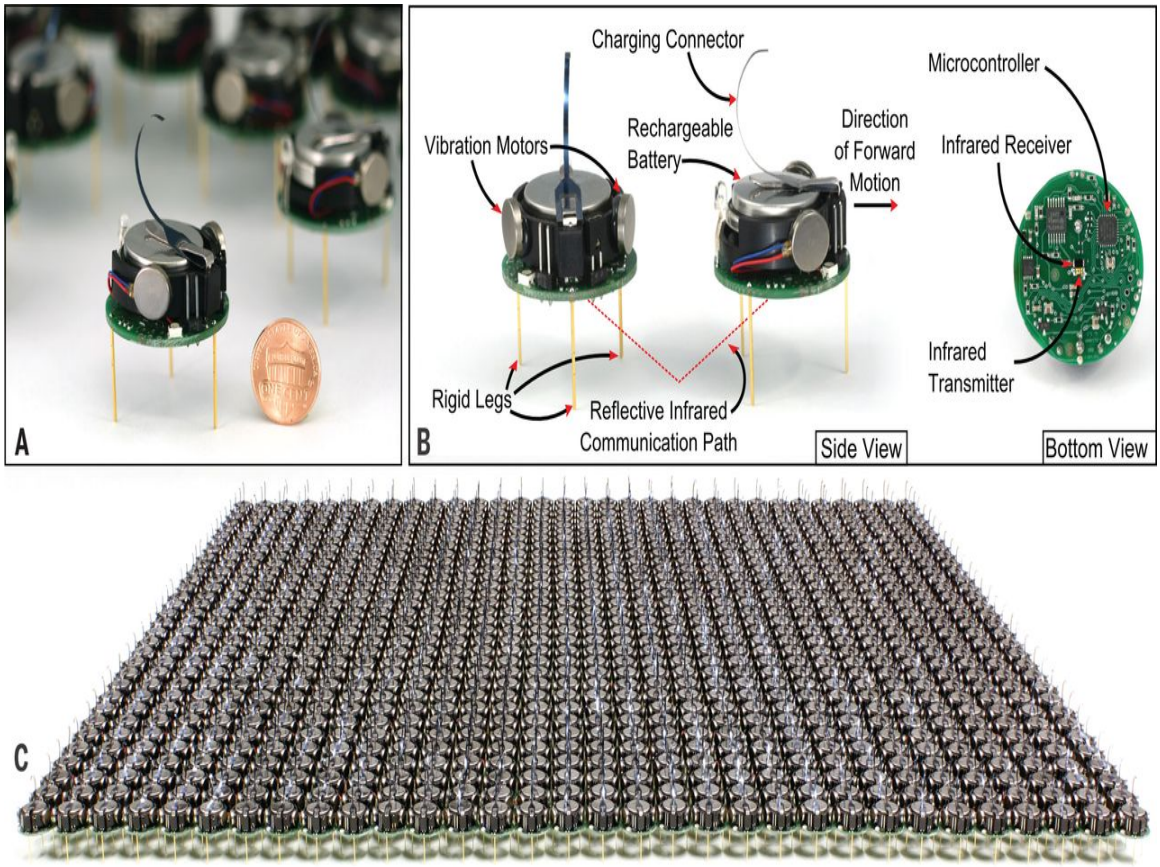


Figure 1.8: Kilobots Swarm Robotic Platform. (a) Size Comparison. (B) Sensing, Power, and Actuation Architecture. (C) 1000 Kilobots Arranged Back-to-Back.

nature like local sensing and local communication. *Adaptability* in SR is a property that allows swarms to respond to changing environment. This is partially facilitated by redundancy ingrained in the swarm by using stochastic policies Elamvazhuthi and Berman (2019) for problems like consensus, coverage.

Aerial swarms (AeS) defer from the previously discussed swarm robots in one major category. AeS must operate autonomously and safely in a complex 3D environment. The success of such AeS would depend on the capability of controlling individual and collective behavior of all robots with due consideration to the limited on board resources for sensing, computation, communication, actuation and mini-

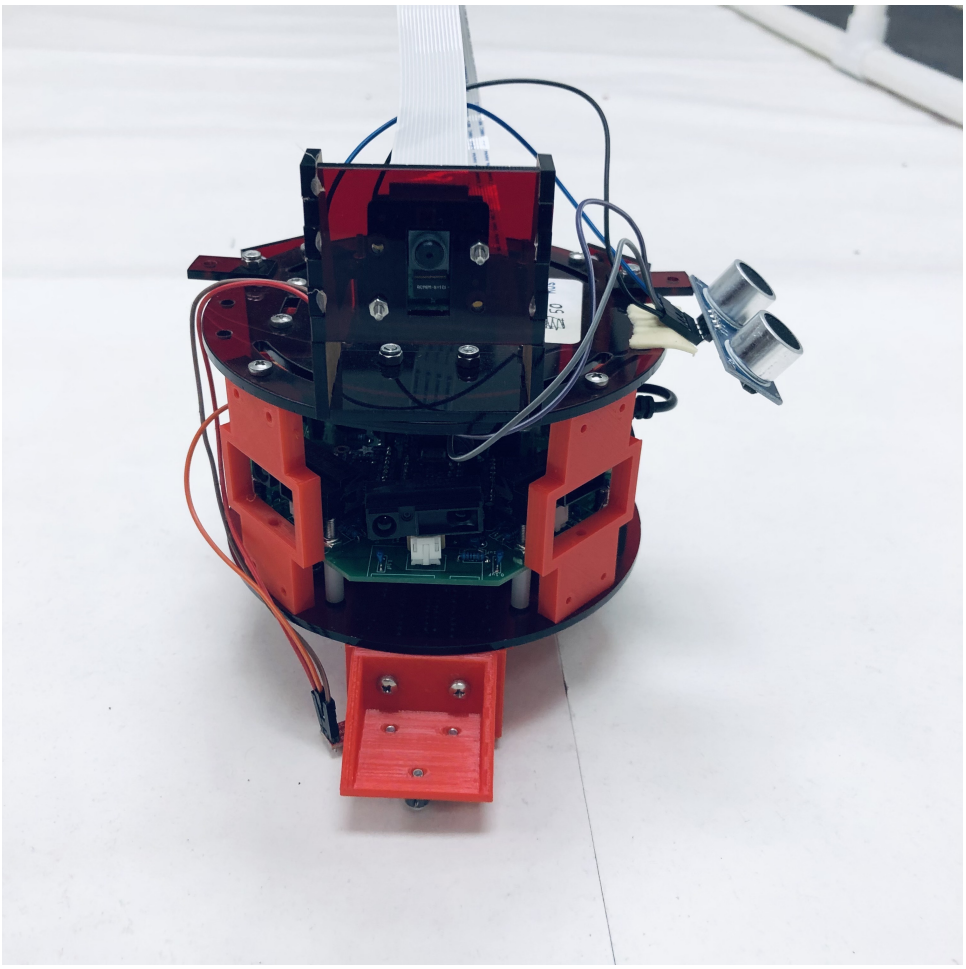


Figure 1.9: Pheeno, a Versatile Swarm Robotic Research and Education Platform.

mal human intervention. The potential commercial applications include *Urban Air Mobility (UAM)*Thipphavong *et al.* (2018), drone deliveryKellermann *et al.* (2020) to name a few. AeS research also aims at addressing problems considered in MAS like coordination, control, task allocation, target search and tracking under the SR paradigm. Models for aerial swarms range from discrete models like Reynolds' model Reynolds (1987) to continuous models utilizing PDE's Ferrari-Trecate *et al.* (2006). Probabilistic strategies have also been used for modeling swarm behaviors like Bandyopadhyay *et al.* (2017); Elamvazhuthi *et al.* (2020); Biswal *et al.* (2019) which used Markov process to model the robot motion as transition probabilities between states

Figure 1.10: Crazyflie 2.0 Nano Quadcopter.

or nodes. Hereinafter, the terms *agents* and *robots* are used interchangeably.

1.1.3 Related Work on Multi Robot Systems

Teams of mobile robots have proved to be useful for exploring and mapping environments in disaster response scenarios Michael *et al.* (2014); Burgard *et al.* (2005); Nagatani *et al.* (2013). However, such robots are subject to constraints on the payloads that they can carry, including power sources, sensors, embedded processors, actuators, and communication devices for transmitting information to other agents and/or to a command center. In addition, many multi-robot control strategies rely on a communication network for coordination. Mobile ground robots Hu *et al.* (2020)

Figure 1.11: Light Trails of CrazySwarm in Action.

and aerial robots Sampedro *et al.* (2019) have often been used for exploration and mapping tasks. Heterogeneous teams of ground and aerial robots have been employed for applications that involve mapping an environment, such as disaster response Murphy *et al.* (2016) and surveillance Dang *et al.* (2019). Such tasks require the robots to track features of interest that are present in the environment. Mobile robots, especially quadrotors, are subject to limitations on their operation due to constraints on the payloads that they can carry, including power, sensing and communication devices for transmitting information to other robots and/or to a command center. Many multi-robot control strategies rely on a centralized communication network for coordination. For example, some multi-robot exploration strategies, e.g. Mahdoui *et al.* (2020), rely on constant two-way communication between the robots and a cen-

Figure 1.12: Surface Boat Equipped With a Compass, GPS, IMU for Studying the Performance of a Over Surface Robots in a Real Aquatic Environment for Achieving a Collective Goal Such As Clustering, Homing.

tral node. Since a centralized communication architecture is required, these strategies do not scale well with the robot population size, as the communication bandwidth becomes a bottleneck with increasing numbers of robots. Communication among robots can become unreliable as the number of robots increases Howard *et al.* (2006); Rizk *et al.* (2019), and the communication network connectivity may be disrupted by the environment Husain *et al.* (2013); Vaquero *et al.* (2019) or by the movement of robots beyond communication range.

Centralized *Multi-Robot System (MRS)* strategies for exploration and mapping, e.g. Rocha *et al.* (2005); Sun *et al.* (2018), rely on constant communication between all the robots and a central node. For these strategies to scale with the number of

Figure 1.13: Swarm of 10 Boats Used to Study the Performance of Swarming Strategies in a Real World Aquatic Environment.

robots, the communication infrastructure needs to be expanded to accommodate the additional bandwidth required for more robots to communicate, which limits their applicability. Moreover, the failure of the central node disrupts communication for all robots, resulting in failure of the entire system.

Decentralized multi-agent control strategies that employ communication networks often require the agents to reach *consensus* on a particular variable. Achieving consensus is the problem of arriving at a common output variable or global property from measurements by distributed agents with local communication, without the need for a supervisory agent (leader or central processor) Spanos *et al.* (2005). Consensus problems have been studied in the cases of static or fixed network topologies Ren *et al.* (2007) and dynamic or switching network topologies George and Freeman (2019),

directed and undirected communication graphs Ren and Beard (2004), random networks Mesbahi and Egerstedt (2010), and mobile networks with communication delays Olfati-Saber and Murray (2004). Consensus algorithms for multi-robot rendezvous, e.g. Parasuraman *et al.* (2018); Yu and Hsieh (2020); Fox *et al.* (2006); Vincent *et al.* (2008), are an example of such a strategy on a dynamic network. The robot controllers drive the robots to meet at a common location in order to enable their information exchange via local communication. However, such strategies restrict exploration since the robots must aggregate at a common location. Distributed consensus for merging individual agents' information has been previously used for multi-agent search, e.g. Hu *et al.* (2012); however, it requires a connected communication network. Although random mobility models are commonly used in multi-robot exploration, e.g. Martinez *et al.* (2012); Wagner *et al.* (1998); Winfield (2000); Kegeleirs *et al.* (2019); Dimidov *et al.* (2016), few works consider consensus problems for communication networks that exhibit Markovian switching dynamics. Consensus on *opinion pools* have been developed in Stone (1961); Degroot (1974); Bailey *et al.* (2012) and guarantee the convergence to a common shared opinion.

1.2 Contributions

The main contributions of this dissertation are outlined as follows:

1. A probabilistic consensus framework for multi-robot system robust to communication link failures is formulated. The robots (or agents) are capable of local sensing in a limited field of view. Each robot explores a bounded environment according to a Discrete-Time Discrete-Space (DTDS) Markov model. We have theoretically proven convergence of the algorithm under these Markovian switching robot dynamics.

2. Decentralized multi-target tracking with Gaussian Mixture (GM) approximation of the Probability Hypothesis Density (PHD) filter to track multiple stationary targets is implemented. Markov Renewal Reward process is used to validate the consensus framework using this GM-PHD filter.

3. A fusion algorithm for generating a shared occupancy grid map by robotic swarms is proposed. We have generalized the results for opinion pools over connected networks and proven theoretically the convergence results under a Markovian switched network using a modification of the distributed Chernoff fusion rule.

4. Rigorous validation of the framework through extensive numerical simulations and 3D physics simulation with simulated quadrotors in Gazebo with ROS. We also show the effects of robot population and environment size on the convergence time under this framework.

Chapter 2

SINGLE-TARGET SEARCH USING CONSENSUS ON MARKOV CHAINS

This chapter presents research and results from Shirsat *et al.* (2020)[**A. Shirsat**, K. Elamvazhuthi, and S. Berman, "Multi-Robot Target Search Using Probabilistic Consensus on Discrete Markov Chains," in 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR).IEEE, 2020, pp. 108–115]

2.1 Overview

Disaster areas, such as regions affected by earthquakes and floods, experience great disruption to communication and power infrastructure. This presents challenges in coordinating searches for survivors and dispersing relief teams to those locations. To address this problem, we present and analyze a probabilistic multi-agent search strategy that is based on a distributed consensus protocol. The proposed strategy is decentralized and asynchronous and relies on only limited communication among agents. Thus, it can be employed in applications, such as disaster response scenarios, where it is infeasible to maintain a connected communication network, rendezvous, or communicate with a central node. The agents move according to a discrete-time discrete-state (DTDS) Markov chain model on a finite spatial grid, as illustrated in Figure 2.1. We consider only static features here, which represent persistent characteristics of the target(s) that the agents are searching for in the environment.

The organization of this chapter is as follows. Section 2.2 presents the problem formulation, and Section 2.3 describes the probabilistic motion model of the agents. Section 2.4 proves that all agents will reach consensus on the presence of the feature under our stochastic search strategy. Section 2.5 presents example implementations

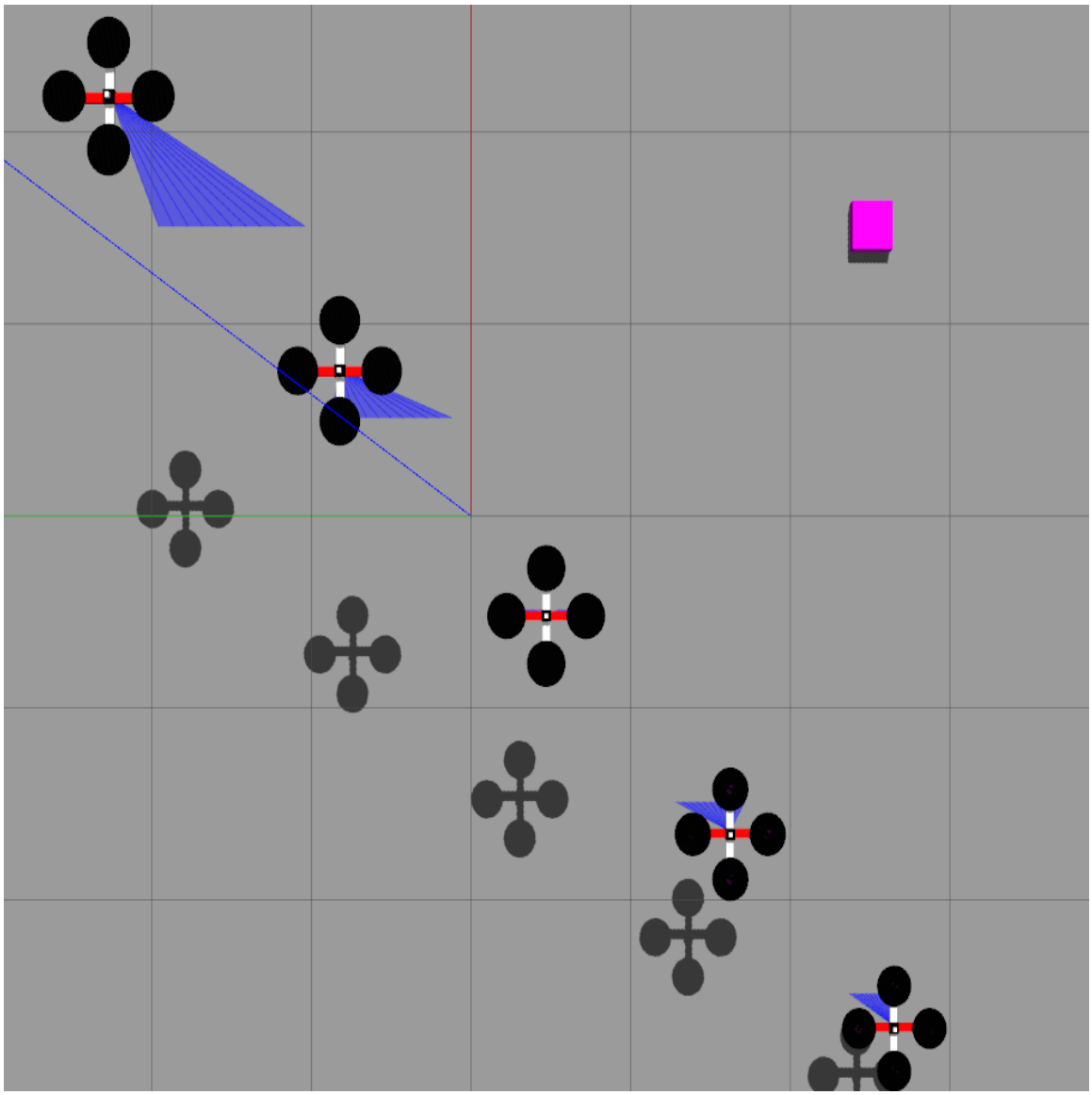


Figure 2.1: Overhead view of problem scenario, simulated in Gazebo 9 (Koenig and Howard, 2004). Multiple aerial robots, flying at different heights, search for a target represented by the magenta box using a Markov chain motion model.

of our strategy in numerical and 3D physics simulations and discusses the results. Section 2.6 concludes the chapter.

2.2 Problem Formulation

We consider an unknown, bounded environment that contains a finite, non-zero number of static features of interest, indexed by the set $\mathcal{I} \subset \mathbb{Z}_+$, where $\mathbb{Z}_+$ is the set of positive integers. A set of N agents, indexed by the set $\mathcal{N} = \{1, 2, ..., N\}$, explore the environment using a random walk strategy. We assume that each agent can localize itself in the environment and can detect a feature within its sensing range. When an agent $a \in \mathcal{N}$ detects a feature at discrete time k, it associates a scalar *information state* $\xi_a[k] \in \mathbb{R}_{\geq 0}$ with its current position. The vector of information states for all agents at time k is denoted by $\boldsymbol{\xi}[k]$. Defining $\mathcal{U}(0, 1)$ as the uniform probability distribution on the interval $[0, 1]$, the initial information state of each agent a is specified *a priori* as $\xi_a[0] \sim \mathcal{U}(0, 1)$. The agent can communicate its information state $\xi_a[k]$ at time k to all agents within a disc of radius $r_{comm} \in (0, \delta]$, where δ is the maximum communication radius. We define these agents as the set of *neighbors* of agent a at time k, denoted by $\mathcal{N}_k^a$. Additionally, we assume that the agents can avoid obstacles during their exploration. Since the agents are constantly moving, the set of agents with which they can communicate changes over time. The time evolution of this communication network is determined by the random walks of the agents throughout the bounded environment. This approach uses low communication bandwidth, since each agent only transmits a scalar value associated with each feature that it detects, to only the current set of neighboring agents.

We discretize the environment, as shown in Figure 2.2, into a square grid of nodes spaced at a distance d apart. The set of nodes is denoted by $\mathcal{S} \subset \mathbb{Z}_+$. We define $S = |\mathcal{S}|$. Let $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ be an undirected graph associated with this finite spatial

grid, where $\mathcal{V}_s$ is the set of nodes and $\mathcal{E}_s$ is the set of edges (i, j) that signify pairs of nodes $i, j \in \mathcal{V}_s$ between which agents can travel. We refer to these pairs of nodes as *neighboring nodes*. Each agent performs a random walk on this grid, moving from its current node i to a neighboring node j at the next time step with transition probability p_{ij}. Let $Z_k^a \in \mathcal{S}$ be a random variable that represents the index of the node that an agent $a \in \mathcal{N}$ occupies at the discrete time k. For each agent a, the probability mass function $\pi_k \in \mathbb{R}^{1\times S}$ of Z_k^a evolves according to a DTDS Markov chain:

$$\pi_{k+1} = \pi_k \mathbf{P}, \tag{2.1}$$

where the *state transition matrix* $\mathbf{P} \in \mathbb{R}^{S\times S}$ has elements $p_{ij} \in [0, 1]$ at row $i \in \mathcal{S}$ and column $j \in \mathcal{S}$.

We assume that no prior information about possible search locations is available. To cover the search area uniformly, each agent is deployed from a random node on the spatial grid. These initial agent positions are chosen independently of one another and are identically distributed according to the probability mass function π_0, defined as a discrete uniform distribution over the set of nodes. We define $\xi^r \in \mathbb{R}_{\geq 0}$ as a scalar *reference information state* that is associated with the set of nodes $\mathcal{Z}^r \subset \mathcal{S}$ from which an agent can detect a feature. In this work, we only consider environments with a single feature of interest.

We now define another graph that models the time-varying communication topology of the agents as they move along the spatial grid. Let $\mathcal{G}_c[k] = (\mathcal{V}_c, \mathcal{E}_c[k])$ be an undirected graph in which $\mathcal{V}_c = \mathcal{N}$, the set of agents, and $\mathcal{E}_c[k]$ is the set of all pairs of agents $(a, b) \in \mathcal{N} \times \mathcal{N}$ that can communicate with each other at time k. Let $\mathbf{M}[k] \in \mathbb{R}^{N\times N}$ be the *adjacency matrix* with elements $m_{ab}[k] = 1$ if $(a, b) \in \mathcal{E}_c[k]$ and $m_{ab}[k] = 0$ otherwise. We define $\mathbf{L}[k] \in \mathbb{R}^{N\times N}$ as the graph Laplacian, whose elements are $l_{ab}[k] = \sum_{b=1}^{N} m_{ab}[k] = deg(v_a)$ if $a = b$ and $l_{ab}[k] = -m_{ab}[k]$ if $a \neq b$.

Given the agent dynamics Equation (2.1) on the spatial grid, each agent a updates its information state at each time k according to a consensus protocol similar to one developed in (Ren and Beard, 2008). This update is based on the agent's current information; the information from all its neighboring agents, of which there are at most $d_{max} = N - 1$; and the reference information state:

$$\begin{aligned}\xi_a[k+1] = \xi_a[k] - \alpha \sum_{b \in \mathcal{N}_k^a} l_{ab}[k](\xi_a[k] - \xi_b[k]) \\ - \ g_a(\xi_a[k] - \xi^r),\end{aligned} \tag{2.2}$$

where $a, b \in \mathcal{N}$; α is a constant, chosen such that $\alpha \in (0, \frac{1}{d_{max}})$ (Olfati-Saber and Murray, 2004); and g_a is defined as:

$$g_a = \begin{cases} 1, & Z_k^a \in \mathcal{Z}^r \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

In the following two sections, we will show that when agents move on the spatial grid according to Equation (2.1) and exchange information with their neighbors according to Equation (2.2), they achieve *average consensus* on their information states, defined as follows:

Definition 2.2.1. *We say that the vector $\boldsymbol{\xi}[k]$ converges almost surely to average consensus if*

$$\boldsymbol{\xi}[k] \ \overset{a.s}{\rightarrow} \ \xi^r \mathbf{1}, \tag{2.4}$$

where $\mathbf{1} \in \mathbb{R}^{N \times 1}$ is a vector of ones.

This implies that the agents' individual information states will eventually converge to a common information state that indicates the presence of the object being searched. We define $\mathcal{T}_c$ as the time k at which every agent's information state $\xi_a[k]$

Algorithm 1: Control strategy for agent $a \in \mathcal{N}$

Input: $\alpha, g_a, \epsilon, \xi^r$; $\xi_a[0] \sim \mathcal{U}(0,1)$; $Z_0^a \leftarrow i \in \mathcal{S}$

Output: $k, \xi_a[k]$ for which $|\xi_a[k] - \xi^r| \leq \epsilon$

$k \leftarrow 0$

while $|\xi_a[k] - \xi^r| > \epsilon$ **do**

  sum1 $\leftarrow 0$

  sum2 $\leftarrow 0$

  forall $b \in \mathcal{N}_k^a$ **do**

    `/* agents` a`,` b `communicate */`

    sum1 $\leftarrow$ sum1 $- \alpha l_{ab}[k](\xi_a[k] - \xi_b[k])$

  end

  if $i \in \mathcal{Z}^r$ **then**

    `/* agent` a `detects feature */`

    sum2 $\leftarrow -g_a(\xi_a[k] - \xi^r)$

  end

  $\xi_a[k+1] \leftarrow \xi_a[k]$ + sum1 + sum2

  $Z_{k+1}^a \leftarrow j$, $(i,j) \in \mathcal{E}_s$, with probability p_{ij}

  $i \leftarrow j$

  $k \leftarrow k + 1$

end

reaches ξ^r within a small tolerance ϵ, where $0 \leq \epsilon \ll 1$; i.e., $|\xi_a[\mathcal{T}_c] - \xi^r| < \epsilon$ for all agents $a \in \mathcal{N}$. We consider $\mathcal{T}_c$ to be the time at which the agents reach consensus.

The implementation of this probabilistic search strategy on each agent is described in the pseudo code shown in Algorithm 1. We illustrate the strategy for a scenario with two quadrotors in Figure 2.2. The quadrotors start at the spatial grid nodes indexed by i and j and move on the grid according to the DTDS Markov chain

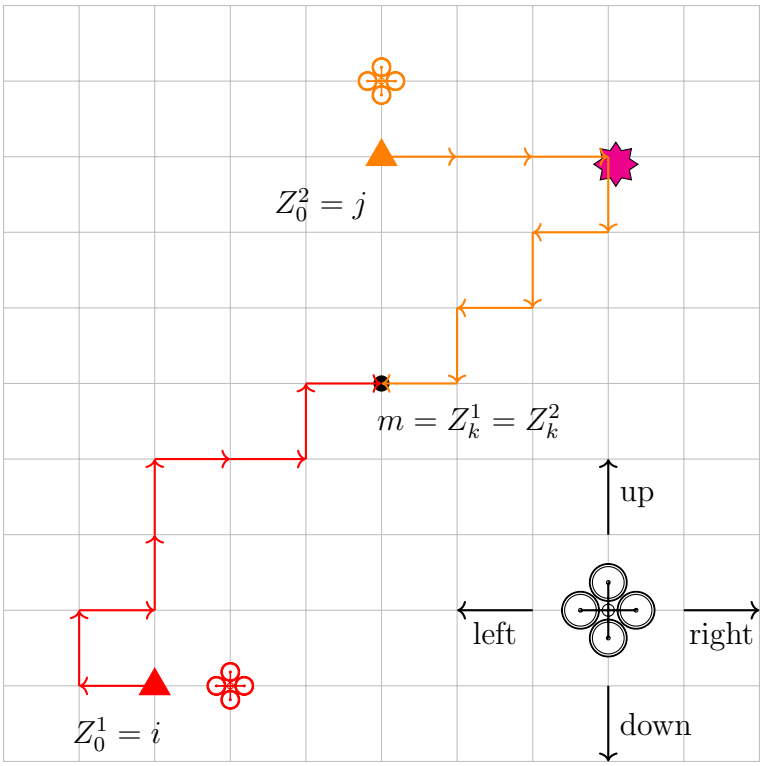


Figure 2.2: Illustration of our multi-agent search strategy, showing sample paths for two quadrotors (orange and red) on a square grid. The quadrotors search the environment for a static target (the magenta star) as they perform a random walk on the grid.

dynamics in Equation (2.1). The figure shows sample paths of the quadrotors. The orange quadrotor detects the feature, indicated by a magenta star, when it moves to a node in the set $\mathcal{Z}^r$ (at these nodes, the feature is within the quadrotor's sensing range). The quadrotors meet at grid node m after $k = 9$ time steps and exchange information according to Equation (2.2). They stop the search if their information states are within ϵ of ξ^r; otherwise, they continue to random-walk on the grid.

2.3 Random Mobility Model Analysis

Consider the DTDS Markov chain that governs the probability mass function of the state Z_k^a, defined as the location of agent a at time k on the spatial grid that represents the environment. Then, the time evolution of the agent a's movement in

this finite state space can be expressed by using the Markov property as follows:

$$\begin{aligned} Pr(Z^a_{k+1} = j \mid Z^a_k = i, Z^a_{k-1} = m, \ldots, Z^a_0 = l) \\ = Pr(Z^a_{k+1} = j \mid Z^a_k = i), \end{aligned} \tag{2.5}$$

where the second expression is the probability with which an agent at node j transitions to node i at time $k+1$, and $m, l \in \mathbb{Z}_+$.

2.3.1 State Transition Matrix

The Markov chain Equation (2.1) is expressed in terms of the *state transition matrix* $\mathbf{P}$. The time invariant matrix $\mathbf{P}$ is defined by the state space of the spatial grid representing the discretized environment. Hence, the Markov chain is *time* homogeneous, which implies that $Pr(Z^a_{k+1} = j \mid Z^a_k = i)$ is the same for all agents at all times k. The entries of $\mathbf{P}$, which are the state transition probabilities, can therefore be defined as

$$p_{ij} = Pr(Z^a_{k+1} = j \mid Z^a_k = i),\ \forall i, j \in \mathcal{S},\ k \in \mathbb{Z}_+,\ \forall a \in \mathcal{N}. \tag{2.6}$$

Since each agent chooses its next node from a uniform distribution, these entries can be computed as

$$p_{ij} = \begin{cases} \frac{1}{d_i+1}, & (i,j) \in \mathcal{E}_s, \\ 0, & \text{otherwise}, \end{cases} \tag{2.7}$$

where d_i is the degree of the node $i \in \mathcal{S}$, defined as $d_i = 2$ if i is a corner of the spatial grid, $d_i = 3$ if it is on an edge between two corners, and $d_i = 4$ otherwise. Since each entry $p_{ij} \geq 0$, we use the notation $\mathbf{P} \geq 0$. We see that $\mathbf{P}^m \geq 0$ for $m \geq 1$. Hence, $\mathbf{P}$ is a *non-negative matrix.* Using Theorem 5 in (Grimmett and Stirzaker, 2001), we can conclude that the state transition matrix $\mathbf{P}$ is a stochastic matrix.

2.3.2 Stationary Distribution

A stationary distribution of a Markov chain is defined as follows.

Definition 2.3.1. *(Page 227 in (Grimmett and Stirzaker, 2001)) The vector $\pi \in \mathbb{R}^S$ is called a stationary distribution of a Markov chain if π has entries such that:*

1. $\pi_j \geq 0 \quad \forall j \in \mathcal{S}$ *and* $\sum_{j=1}^{S} \pi_j = 1$

2. $\pi \mathbf{P} = \pi$

Thus, if π is a stationary distribution, we can say that $\forall k \in \mathbb{Z}_+$,

$$\pi \mathbf{P}^k = \pi. \tag{2.8}$$

From the construction of the Markov chain Equation (2.1), each agent has a positive probability of moving from any node $i \in \mathcal{S}$ to any other node $j \in \mathcal{S}$ of the spatial grid in a finite number of time steps. As a result, the Markov chain Z_k^a is an *irreducible* Markov chain, and therefore $\mathbf{P}$ is an irreducible matrix.

From Lemma 8.4.4 (Perron-Frobenius) in (Horn and Johnson, 1990), we know that there exists a real unique positive left eigenvector of $\mathbf{P}$. Moreover, since $\mathbf{P}$ is a stochastic matrix, its spectral radius $\rho(\mathbf{P})$ is equal to 1. Therefore, we can conclude that this left eigenvector is the stationary distribution of the corresponding DTDS Markov chain. We will next apply the following theorem.

Theorem 2.3.1. *(Theorem 21.12 in (Levin and Peres, 2017)) An irreducible Markov chain with transition matrix $\mathbf{P}$ is positive recurrent if and only if there exists a probability distribution π such that $\pi \mathbf{P} = \pi$.*

Since we have shown that the Markov chain is *irreducible* and has a stationary distribution π, which satisfies $\pi \mathbf{P} = \pi$, we can conclude from Theorem Theorem 2.3.1

that the Markov chain is *positive recurrent*. Thus, all states in the Markov chain are positive recurrent, which implies that each agent will keep visiting every state on the finite spatial grid infinitely often.

2.4 Information Consensus Analysis

The dynamics of all agents' movements on the spatial grid can be modeled by a composite Markov chain with states defined as $\mathbf{Z}_k = (Z_k^1, Z_k^2, ..., Z_k^N) \in \mathcal{M}$, where $\mathcal{M} = \mathcal{S}^{\mathcal{N}}$. Note that $S = |\mathcal{S}|$ and $|\mathcal{M}| = S^N$. We define an undirected graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ that is associated with the composite Markov chain. The vertex set $\hat{\mathcal{V}}$ is the set of all possible realizations $\hat{\imath} \in \mathcal{M}$ of $\mathbf{Z}_k$. The notation $\hat{\imath}(a)$ represents the a^{th} entry of $\hat{\imath}$, which is the spatial node $i \in \mathcal{S}$ occupied by agent a. We define the edge set $\hat{\mathcal{E}}$ of the graph $\hat{\mathcal{G}}$ as follows: $(\hat{\imath}, \hat{\jmath}) \in \hat{\mathcal{E}}$ if and only if $(\hat{\imath}(a), \hat{\jmath}(a)) \in \mathcal{E}_s$ for all agents $a \in \mathcal{N}$. Let $\mathbf{Q} \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{M}|}$ be the state transition matrix associated with the composite Markov chain. The elements of $\mathbf{Q}$, denoted by $q_{\hat{\imath}\hat{\jmath}}$, are computed from the transition probabilities defined by Equation (2.7) as follows:

$$q_{\hat{\imath}\hat{\jmath}} = \prod_{a=1}^{N} p_{\hat{\imath}(a)\hat{\jmath}(a)}, \qquad \forall \hat{\imath}, \hat{\jmath} \in \mathcal{M}. \tag{2.9}$$

In the above expression, $q_{\hat{\imath}\hat{\jmath}}$ is the probability that in the next time step, each agent a will move from spatial node $\hat{\imath}(a)$ to node $\hat{\jmath}(a)$.

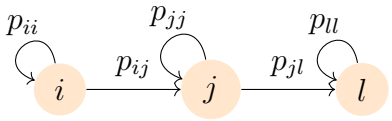


Figure 2.3: A graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ defined on the set of spatial nodes $\mathcal{V}_s = \{i, j, l\}$. The arrows signify directed edges between pairs of distinct nodes or self-edges. The edge set of the graph is $\mathcal{E}_s = \{(i,i), (j,j), (l,l), (i,j), (j,l)\}$.

For example, consider a set of two agents, $\mathcal{N} = \{1, 2\}$, that move on the graph $\mathcal{G}_s$ as shown in Figure 2.3. The agents can stay at their current node in the next time

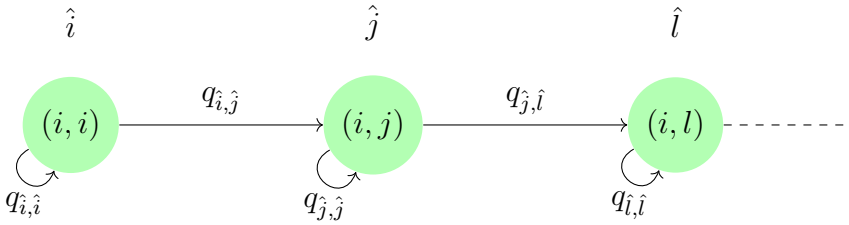


Figure 2.4: A Subset of the Composite Graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ For 2 Agents That Move on the Graph $\mathcal{G}_s$ Shown In Figure 2.3.

step or travel between nodes i and j and between nodes j and l, but they cannot travel between nodes i and l. Figure 2.4 shows a subset of the resulting composite graph $\hat{\mathcal{G}}$. The set of nodes in the graph $\hat{\mathcal{G}}$ is $\hat{\mathcal{V}}$

$$\hat{\mathcal{V}} == \{(i,i),(i,j),(i,l),(j,i),(j,j),(j,l),(l,i),(l,j),(l,l)\}$$

Each node in $\hat{\mathcal{V}}$ is labeled by a single index $\hat{\imath}$, e.g., $\hat{\imath} = (i,j)$, with $\hat{\imath}(1) = i$ and $\hat{\imath}(2) = j$. Due to the connectivity of the spatial grid defined by $\mathcal{E}_s$, we can for example identify $((i,j),(i,l))$ as an edge in $\hat{\mathcal{E}}$, but not $((i,j),(l,l))$.

Since $N = 2$ and $S = 3$, we have that $|\mathcal{M}| = 3^2 = 9$. For each $\hat{\imath}, \hat{\jmath} \in \hat{\mathcal{V}}$, we can compute the transition probabilities in $\mathbf{Q} \in \mathbb{R}^{9\times 9}$ from Equation (2.9) as follows:

$$q_{\hat{\imath}\hat{\jmath}} = Pr\left(\mathbf{Z}_{k+1} = \hat{\jmath} \mid \mathbf{Z}_k = \hat{\imath}\right) = p_{\hat{\imath}(1)\hat{\jmath}(1)} p_{\hat{\imath}(2)\hat{\jmath}(2)}, \;\; k \in \mathbb{Z}_+. \tag{2.10}$$

We now define $\hat{\boldsymbol{\xi}}[k] = [\xi_1[k] \;\; \xi_2[k] \;\; \ldots \;\; \xi_N[k] \;\; \xi^r]^T \in \mathbb{R}^{N+1}$ as an augmented information state vector. The dynamics of information exchange among the agents modeled by Equation (2.2) can then be represented in matrix form as follows:

$$\hat{\boldsymbol{\xi}}[k+1] = \mathbf{H}[k]\hat{\boldsymbol{\xi}}[k], \tag{2.11}$$

where $\mathbf{H}[k] \in \mathbb{R}^{(N+1)\times(N+1)}$ is defined as

$$\mathbf{H}[k] = \begin{bmatrix} \mathbf{I} - \alpha\mathbf{L}[k] + diag(\mathbf{d}) & -\mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.12}$$

in which $\mathbf{d} = [g_1 \ g_2 \ \dots \ g_N]^T$, $\mathbf{0} \in \mathbb{R}^{1\times N}$ is a vector of zeros, and $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix.

We associate Equation (2.11) with a graph $\mathcal{G}_r[k]$, an expansion of the graph $\mathcal{G}_c[k]$ that includes information flow from the feature nodes $\mathcal{Z}^r$ to agents that occupy these nodes. Here we consider the feature as an additional agent $a_f = N+1$, which remains fixed. Let $\mathcal{G}_r[k] = (\mathcal{V}_r, \mathcal{E}_r[k])$ be a directed graph in which $\mathcal{V}_r = \mathcal{N} \cup a_f$, the set of agents and the feature, and $\mathcal{E}_r[k] = \mathcal{E}_c[k] \ \cup \ \mathcal{E}_f[k]$, where $\mathcal{E}_f[k]$ is the set of agent-feature pairs (a, a_f) for which $Z_k^a \in \mathcal{Z}^r$ at time k. In this graph, information flows in one direction from the feature nodes to all agents that occupy a feature node on the finite spatial grid at time k. In addition, information flows bidirectionally between agents that are neighbors at time k. We now prove the main result of this paper in the following theorem, which shows that all agents will track the reference feature in the environment almost surely and in a distributed fashion.

Theorem 2.4.1. *Consider a group of N agents whose information states evolve according to Equation* (2.11)*. The information states of all agents will converge to the reference information state ξ^r almost surely.*

Proof. Suppose that at an initial time k_0, the locations of the N agents on the spatial grid are represented by the node $\hat{\imath} \in \hat{\mathcal{V}}$. Consider another set of agent locations at a future time $k_0 + k$, represented by the node $\hat{\jmath} \in \hat{\mathcal{V}}$. The transition of the agents from configuration $\hat{\imath}$ to configuration $\hat{\jmath}$ in k time steps corresponds to a random walk of length k on the composite Markov chain $\mathbf{Z}_k$ from node $\hat{\imath}$ to node $\hat{\jmath}$. It also corresponds to a random walk by each agent a on the spatial grid from node $\hat{\imath}(a)$ to node $\hat{\jmath}(a)$ in k time steps. By construction, the graph $\mathcal{G}_s$ is strongly connected and each of its nodes has a self-edge. Thus, there exists a discrete time $n > 0$ such that, for each agent a, there exists a random walk on the spatial grid from node $\hat{\imath}(a)$ to node $\hat{\jmath}(a)$ in n time

steps. Consequently, there always exists a random walk of length n on the composite Markov chain $\mathbf{Z}_k$ from node $\hat{\imath}$ to node $\hat{\jmath}$. Therefore, $\mathbf{Z}_k$ is an irreducible Markov chain. All states of an irreducible Markov chain belong to a single communication class. In this case, all states are *positive recurrent*. As a result, each state of $\mathbf{Z}_k$ is visited infinitely often by the group of agents. Moreover, because the composite Markov chain is irreducible, we can conclude that $\cup_{k\in\mathbb{Z}_+}\mathcal{G}_c[k] = \mathcal{G}_0$, where $\mathcal{G}_0$ is the complete graph on the set of agents $\mathcal{N}$, and therefore that $\cup_{k\in\mathbb{Z}_+}\mathcal{G}_r[k]$ contains a directed spanning tree with ξ^r as the fixed root. Since this union of graphs has a spanning tree, we can apply Theorem 3.1 in (Matei *et al.*, 2009) to conclude that the information state of each agent will converge to ξ^r almost surely. The notation $\theta(k)$ and $F_{\theta(k)}$ in (Matei *et al.*, 2009) corresponds to our definitions of $\mathbf{Z}_k$ and $\mathbf{H}[k]$, respectively. □

2.5 Simulation Results

We validate the result on average information consensus in Theorem Theorem 2.4.1 with numerical simulations in MATLAB and 3D physics-based software-in-the-loop (SITL) simulations developed in ROS-Melodic and Gazebo 9 (Koenig and Howard, 2004). In the simulations, multiple agents perform random walks on a finite spatial grid according to the dynamics in Equation (2.1). Each grid is defined as a square lattice with $c = \sqrt{S}$ nodes on each side, where the distance between neighboring nodes is $d = 1$ m. s. The state transition probabilities p_{ij} of the corresponding graph $\mathcal{G}_s$ are defined according to Equation (2.7). Since our largest simulated agent population is $N = 14$, and the parameter α must be less than $\frac{1}{d_{max}} = \frac{1}{N-1}$ (Olfati-Saber and Murray, 2004), we set $\alpha = \frac{1}{14-1} \approx 0.08$. The tolerance ϵ defining the time until consensus was set to 0.01. All simulations were run on a desktop computer with 16 GB of RAM and an Intel Xeon 3.0 GHz 16 core processor with an NVIDIA Quadro

M4000 graphics processor.

2.5.1 Numerical Simulations

We performed large ensembles of Monte Carlo simulations to investigate the effect of the number of agents N, the spatial grid dimension c, and the resulting agent density N/c^2 on the expected time until the agents reach consensus, i.e., agree that the feature of interest is present. Quantifying the effect of these factors is necessary in order to determine the number of agents that should search a given area. This would help first responders to optimally distribute resources for searching a disaster-affected environment.

Each agent is modeled as a point mass that can move between adjacent nodes on the graph $\mathcal{G}_s$, as illustrated in Figure 2.2. We assume that the agents can localize on $\mathcal{G}_s$. The set of neighbors $\mathcal{N}_k^a$ of an agent a at time k consists of all agents that occupy the same spatial node as agent a at that time. The feature can by detected by an agent located at nodes $\mathcal{Z}^r = \{4, 5, 6\}$ of the spatial grid, and the reference information state of the feature is defined as $\xi^r = 1$.

To investigate the dependence of the expected time to reach consensus, $\mathbb{E}[\mathcal{T}_c]$, on the number of agents N and the spatial grid dimension c, we simulated scenarios with different combinations of $N \in \{2, 3, \ldots, 14\}$ and $c \in \{5, 8, 10, 12, 15, 20\}$ meters. For each scenario, we ran 1000 simulations with random initial agent positions and computed the mean time μ at which the agents reached consensus. Figure 2.5 plots the values of μ versus N and c for each simulated scenario, and Figure 2.6 plots μ versus the corresponding agent density, N/c^2. We observe from these figures that a decrease in the agent density results in an increase in μ. This can be attributed to low agent encounter rates with other agents and with feature nodes at low agent densities. Using the curve fitting toolbox in MATLAB and data from Figure 2.6 we see that

there is an exponential relation between $\mathbb{E}[\mathcal{T}_c]$ and N/c^2 given by $\mathbb{E}[\mathcal{T}_c] = ae^{-b\frac{N}{c^2}}$ with $a = 0.008, b = -15.84$. Figure 2.6 shows that the expected time until consensus does not decrease appreciably for agent densities above approximately $N/c^2 = 0.05$. Thus, for a given grid size c^2, it may not be necessary to deploy more than about $\lceil 0.05c^2 \rceil$ agents ($0.05c^2$ rounded up to the next integer) to search the area.

For selected combinations of N and c, we also computed the standard deviation σ of the time to reach consensus over the corresponding 1000 simulations. Figure 2.7 plots $\mu \pm \sigma$ versus N for a fixed grid dimension $c = 5$, and Figure 2.8 plots $\mu \pm \sigma$ versus c for a fixed number of agents $N = 5$. Figure 2.7 shows that for a relatively small grid size ($c = 5$), both μ and σ do not vary substantially with N. Thus, a small number of agents would be sufficient to search such an environment, since increasing the agent density would not significantly speed up the search or reduce the variability in time until consensus. Figure 2.8 indicates that for a fixed group size of $N = 5$ agents, both μ and σ increase monotonically with the size of the grid. This trend suggests that more agents should be deployed if the predicted time until consensus and/or the variability in this time is too high for a given environment.

We illustrate the agents' consensus dynamics with two cases of the simulation runs. Figure 2.10 plots the time evolution of the agent information states for each case. In the first case, $N = 2$ agents traverse a spatial grid with dimension $c = 3$. From Figure 2.9, we see that the time until consensus, i.e. the time at which both agents' information states converge within ϵ of the reference state $\xi^r = 1$, is approximately 160 s. We also simulate $N = 5$ agents that traverse a spatial grid with dimension $c = 10$. Figure 2.10 shows that the time until consensus has increased to about 570 s in this case, which is within one standard deviation σ of the mean consensus time μ computed from our Monte Carlo analysis, as shown in Figure 2.8 for $c = 10$.

We also studied the effect on $\mathbb{E}[\mathcal{T}_c]$ of uncertainty in the agents' identification of

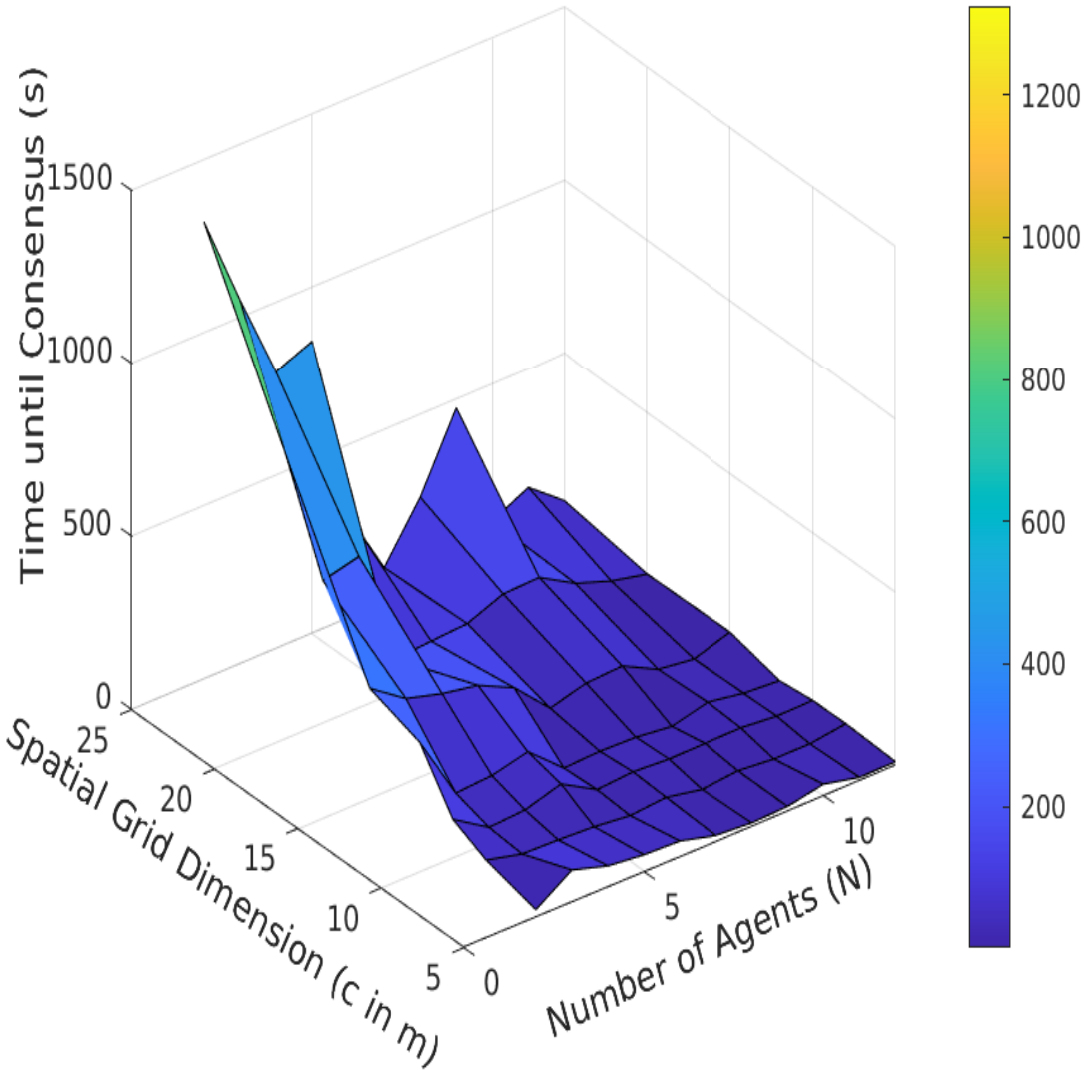


Figure 2.5: Mean Time (s) Until Consensus Is Reached, μ, Versus Number of Agents N and Spatial Grid Dimension c. Each Value of μ Is Averaged Over 1000 Monte Carlo Simulations of Scenarios With the Corresponding Values of N and c.

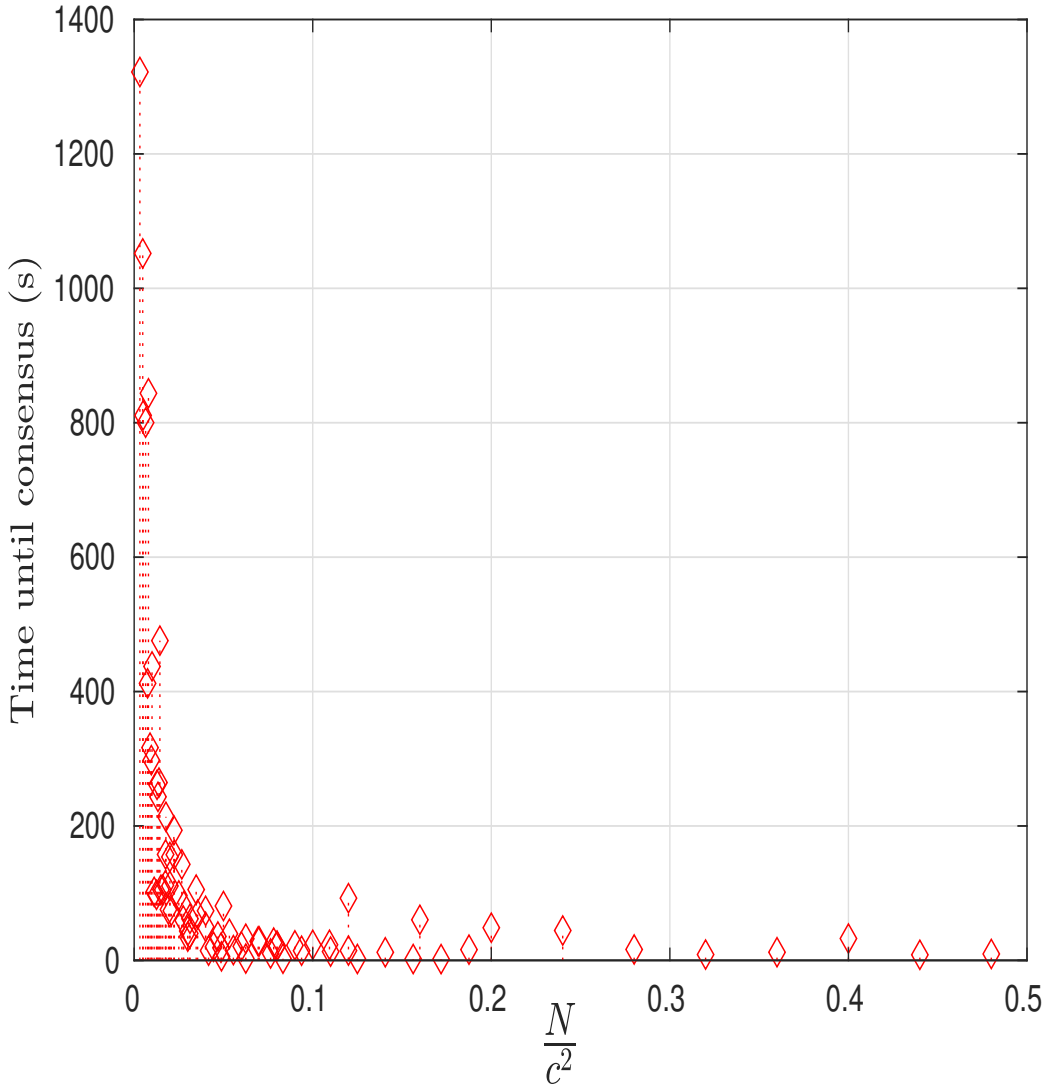


Figure 2.6: Mean Time (s) Until Consensus Is Reached, μ, Versus Agent Density N/c^2 for the Simulation Data Plotted In Figure 2.5.

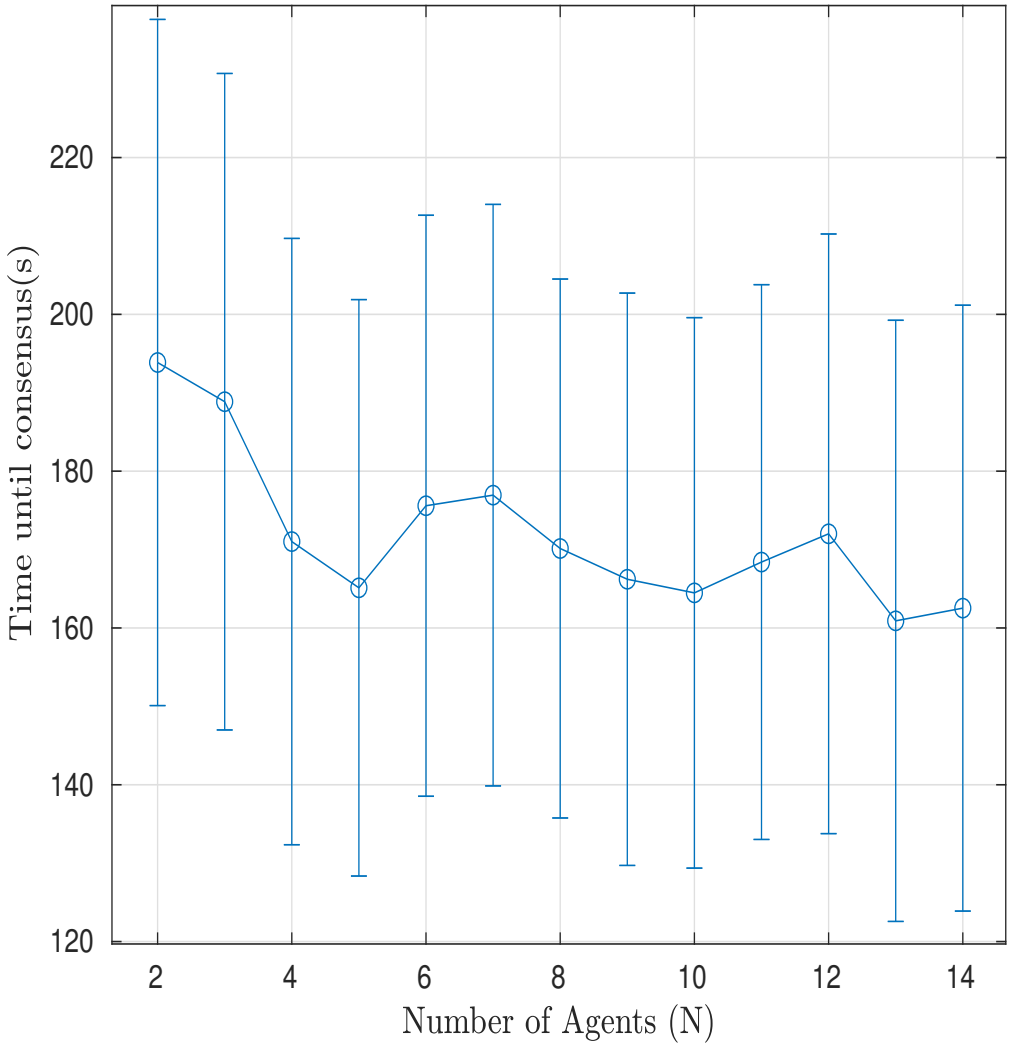


Figure 2.7: Time Until Consensus Is Reached, Averaged Over 1000 Monte Carlo Simulations of Scenarios With Varying Numbers of Agents N and Grid Dimension $c = 5$.

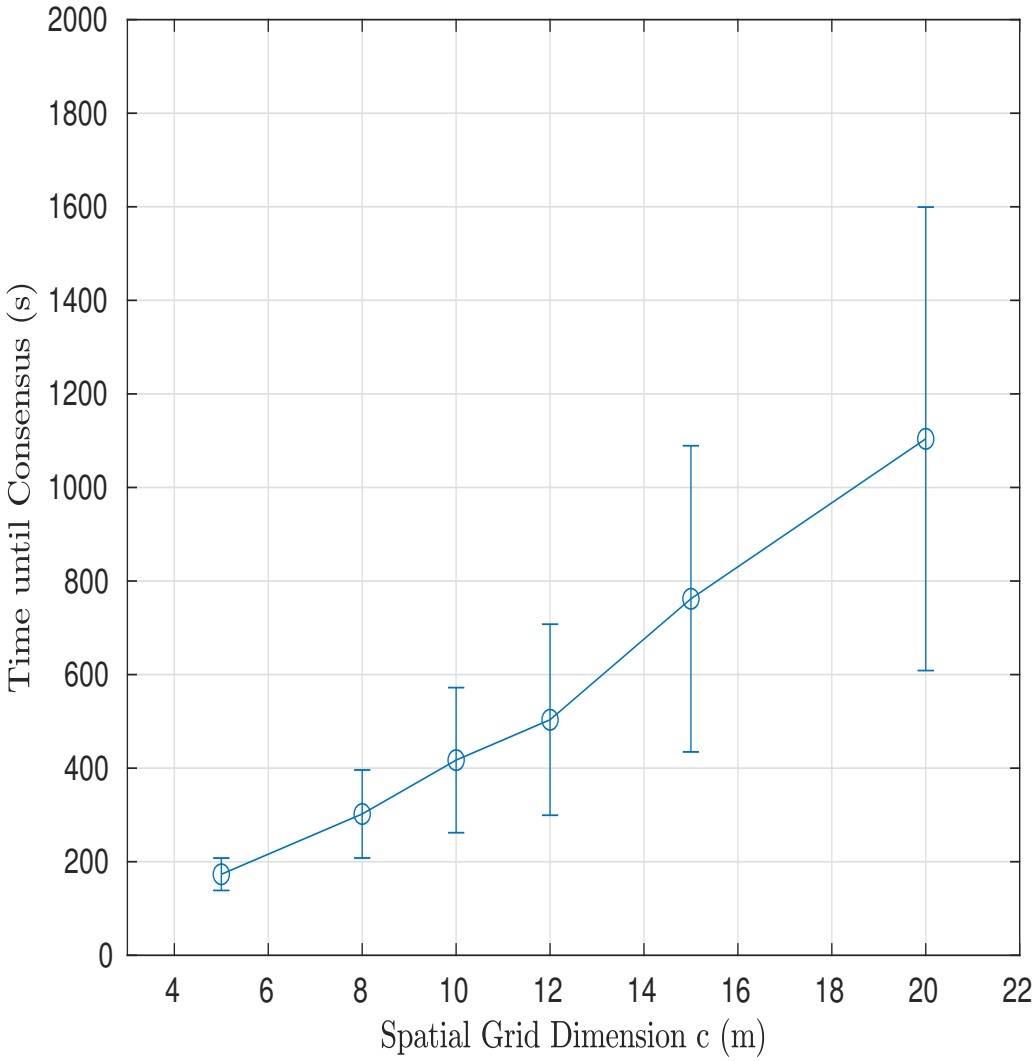


Figure 2.8: Time Until Consensus Is Reached, Averaged Over 1000 Monte Carlo Simulations of Scenarios With Varying c and $N = 5$. The Circles Mark Mean Times μ, and the Error Bars Show Standard Deviations σ.

Reference information state	Time until consensus is reached (s) $\mu \pm \sigma$
$\xi^r = 1$	140 ± 35
$\xi^r \sim \mathrm{N}(1, 0.02)$	175 ± 68

Table 2.1: Time Until Consensus Is Reached ($\mu \pm \sigma$), Computed From 1000 Monte Carlo Simulations of Scenarios With $N = 5$, $c = 5$ and Different Values of ξ^r.

the feature nodes (i.e., ξ^r is a random variable), which may arise in practice due to factors such as sensor noise, occlusion of features, and inter-agent communication failures. We ran 1000 Monte Carlo simulation runs, for each of two scenarios, all with $N = 5$ agents moving on a spatial grid with dimension $c = 5$ m. For each scenario, Table 2.1 shows the mean μ and standard deviation σ of the time until the agents reach consensus. To investigate the effect of uncertainty in feature identification, we specified that agents either perfectly identify the feature, in which case $\xi^r = 1$, or obtain noisy measurements of the feature, for which $\xi^r \sim \mathrm{N}(1, 0.02)$. From Table 2.1, we observe that the addition of noise to the agents' measurements of the feature results in an increase in both μ and σ. However, despite information uncertainty, the agents successfully achieve consensus.

2.5.2 3D Physics Simulations

We also tested our search strategy in physics-based simulations. A snapshot of the Gazebo simulation environment is shown in Figure 2.1. The agents are modeled as quadrotors with a plus frame configuration. We assume that the agents can accurately localize in the environment using on-board inertial and GPS sensors. The analysis of our probabilistic consensus strategy under localization uncertainty is beyond the scope of this paper. We also assume that the feature of interest is known to be present in the environment, but its location is unknown.

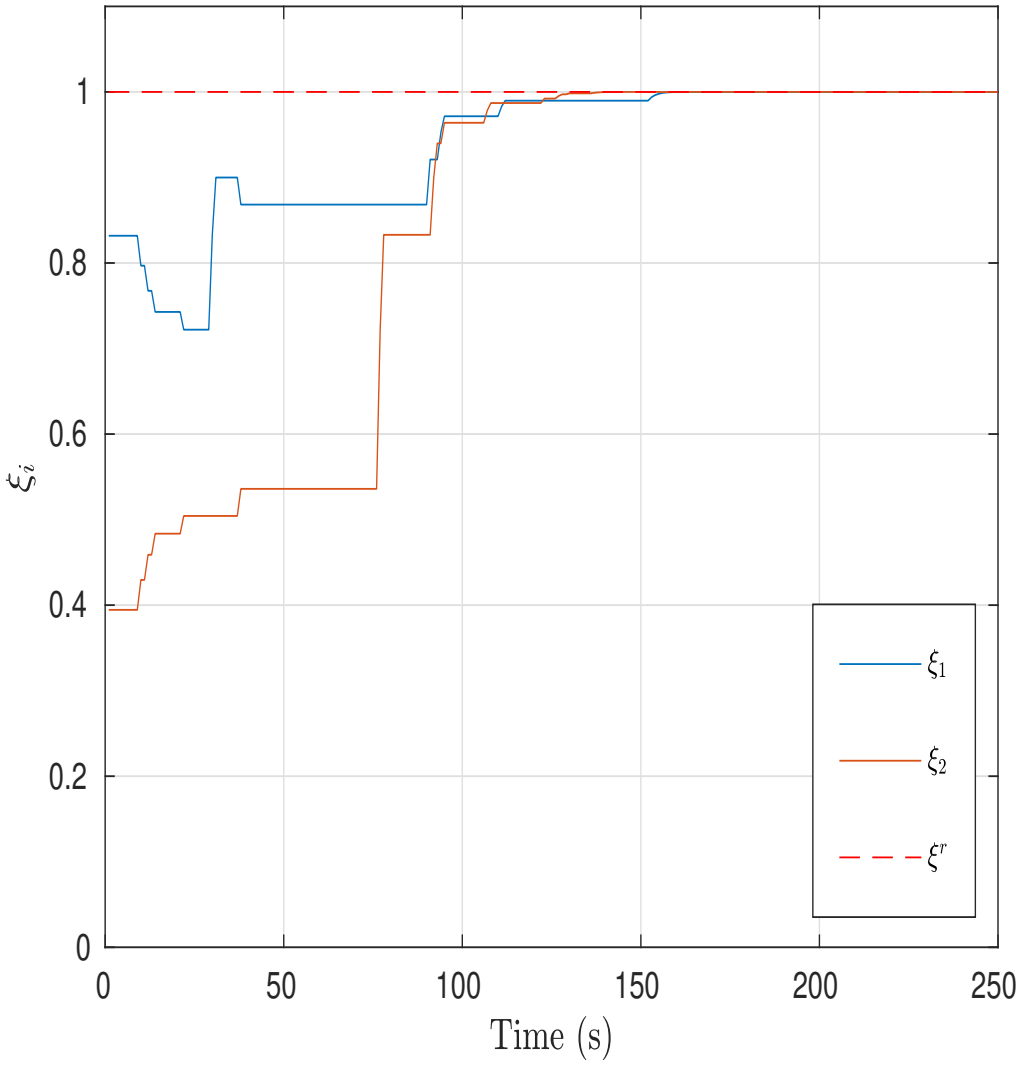


Figure 2.9: Time Evolution of the Agent Information States $\xi_a[k]$ in Simulations of $N = 2$ Agents Moving On a 3×3 Grid.

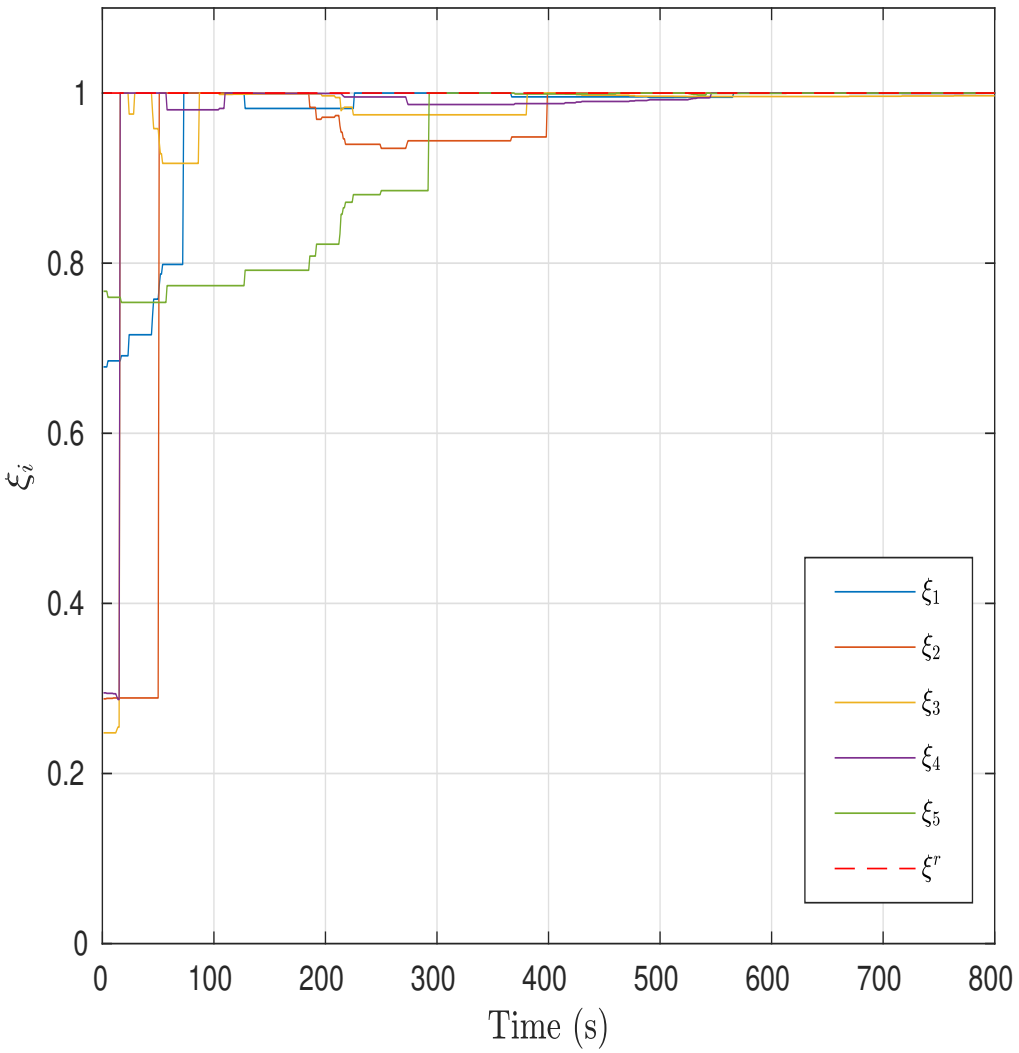


Figure 2.10: Time Evolution of the Agent Information States $\xi_a[k]$ in Simulations of $N = 5$ Agents Moving On a 10×10 Grid.

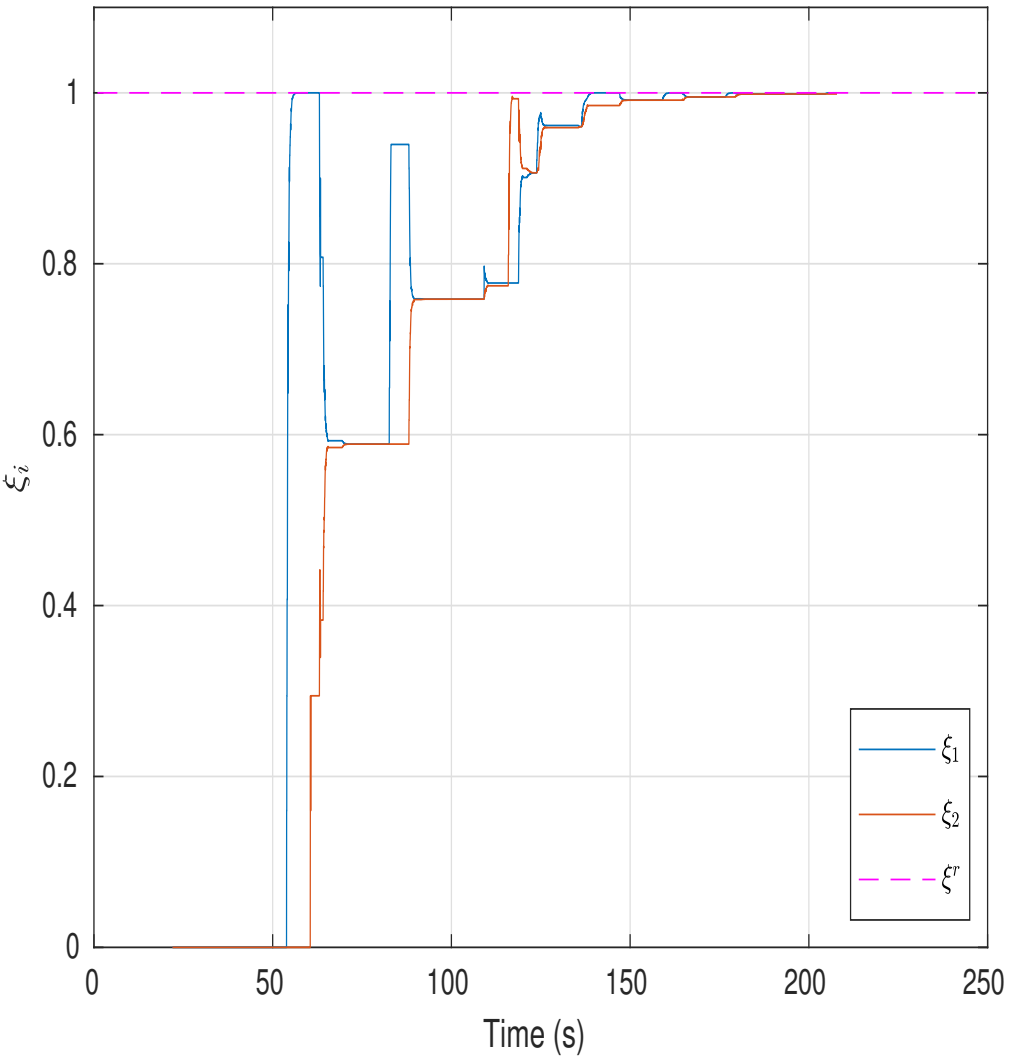


Figure 2.11: Time Evolution of the Robot Information States $\xi_a[k]$ in Gazebo Simulation Runs of $N = 2$ Robots Moving On a 3×3 Grid.

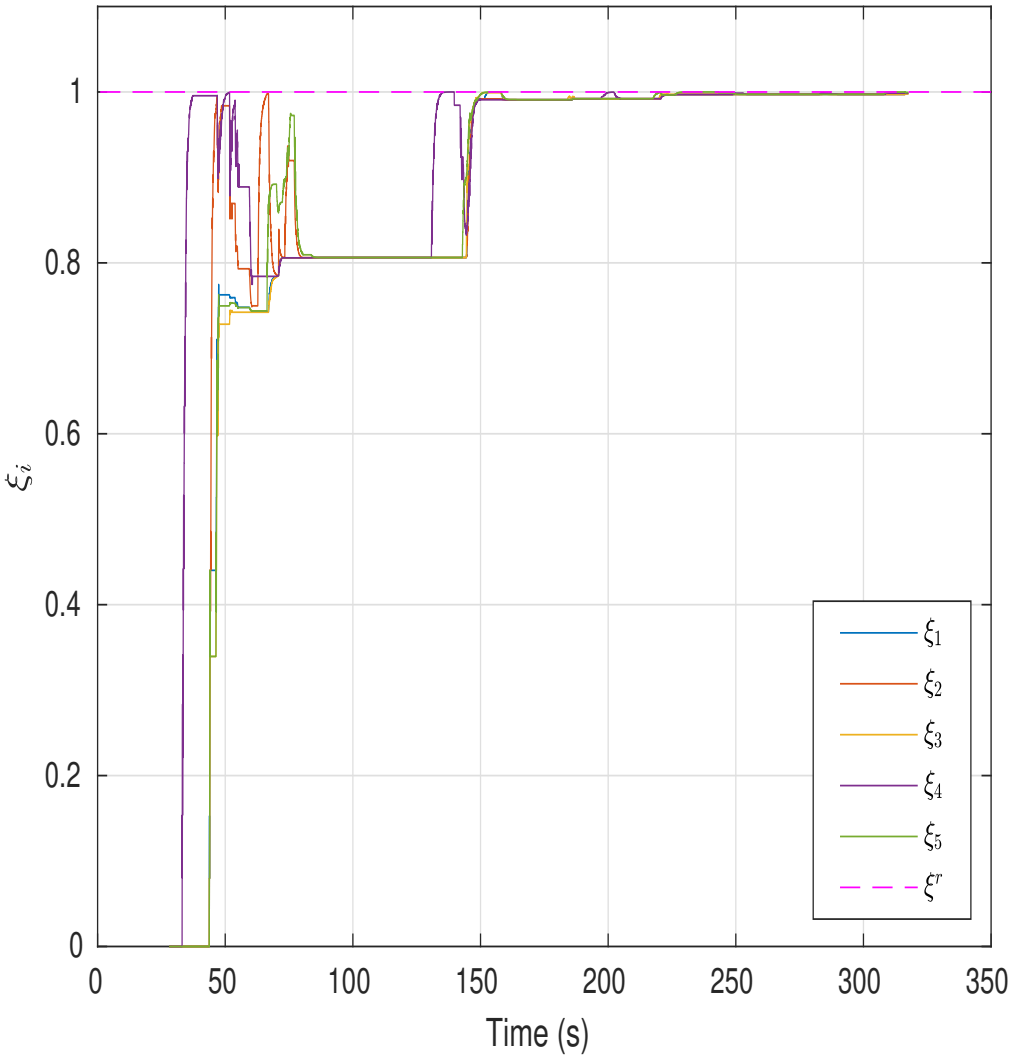


Figure 2.12: Time Evolution of the Robot Information States $\xi_a[k]$ in Gazebo Simulation Runs of (B) $N = 5$ Robots Moving On a 5×5 Grid.

Each quadrotor is equipped with a downward-facing RGB camera with a resolution of 1080×720. The feature of interest is modeled as a magenta box, which the agents detect from their camera images using a color-based classifier. We added zero-mean Gaussian noise with standard deviation 0.07 to the photo-metric intensity in the camera sensor model. We also used a standard plumb bob distortion model to account for camera lens distortion. The quadrotors are spaced 0.5 m apart in altitude in order to prevent collisions. The altitude difference causes slight disparities in the quadrotors' field-of-view (FOV), but this does not significantly affect the performance of the search strategy.

We simulated two scenarios: $N = 2$ robots at altitudes 0.5 m and 1 m traversing a 3×3 grid, and $N = 5$ robots at altitudes between 1 m and 3 m traversing a 5×5 grid. Figure 2.11 and Figure 2.12 plot the time evolution of the agent information states over a single simulation run of each scenario. The information states sometimes display steep drops in value, as in the plots of ξ_2 and ξ_4 in Figure 2.12 from 50 s to 70 s. These drops can be attributed to the following factors: (1) an agent updates its information state with states communicated by its neighbors, according to the consensus protocol; (2) an agent that is at the feature node stops detecting the feature below when another agent at a lower altitude enters its field of view, occluding the feature; (3) spurious measurements like false positives may have been introduced by an agent's sensors. Despite the unmodeled effects of the second and third factors on the information states, the agents still successfully reach consensus during the Gazebo simulations. We see that the time until consensus is reached in Figure 2.11 and Figure 2.12 is about 210 s and 250 s, respectively. The delays in these times compared to the times in the Monte Carlo simulations in Figure 2.7-Figure 2.8 can be attributed to the second and third factors described above and to the inertia of the quadrotor, which affect the Gazebo simulations but not the Monte Carlo simulations.

2.6 Conclusions

In this chapter, we have presented a probabilistic search strategy for multiple agents with local sensing and communication capabilities. The agents explore a bounded environment according to a DTDS Markov motion model and share information with neighboring agents. We proved that the agents achieve consensus almost surely on the presence of a static feature of interest in the environment. Thus, agents that do not detect the feature through direct measurement will eventually recognize its presence through information exchange with other agents. Importantly, this result does not require any assumptions on the connectivity of the agents' communication network. Thus, the search strategy is suitable for applications in which network connectivity is difficult to maintain, such as disaster scenarios. We investigated the performance of our strategy in both numerical and physics-based simulations.

In the next chapter, we extend this approach to a decentralized strategy for tracking multiple static targets by formulating the tracking procedure as a renewal-reward process on the underlying Markov chain.

Chapter 3

DECENTRALIZED MULTI-TARGET TRACKING WITH A PHD FILTER

This chapter presents research and results from Shirsat and Berman (2021) [**A. Shirsat** and S. Berman, "Decentralized Multi-Target Tracking With Multiple Quadrotors Using a PHD Filter," in AIAA Scitech 2021 Forum, 2021]

3.1 Overview

Multi-target tracking is an established field of research with origins in the study of point processes Daley and Vere-Jones (2007), with most early applications in radar and sonar based tracking. In real-world scenarios, there is often uncertainty in the existence, locations, and dynamics of targets, as well as uncertainty in sensor measurements of targets that arise from sensor noise and false detections (clutter) around the real targets. Random Finite Set (RFS) models provide a probabilistic framework for multi-target tracking that can account for these uncertainties and ensure statistical guarantees on the accuracy of the estimated number of targets and their states.

Unlike RFS-based estimators, many classical probabilistic multi-target tracking approaches require techniques for data association, which is computationally intensive. Such approaches include multiple hypothesis tracking Jensfelt and Kristensen (2001); Reid (1979); Yoon *et al.* (2018), in which an exhaustive search on all possible combinations of tracks and data associations is performed, and joint probabilistic data association Bar-Shalom and Fortmann (1988); Schulz *et al.* (2001); Stauch *et al.* (2018). The papers Mahler *et al.* (2001); Mahler (2003, 2007) are foundational works on estimation methods based on Random Finite Sets, and they have made concepts

from point process theory for multi-target tracking more accessible to engineering disciplines. The paper Clark *et al.* (2006) presents an implementation of a multi-target tracking approach as a probability hypothesis density (PHD) filter, and Vo and Ma (2006) provides examples of scenarios with targets that follow either linear or non-linear motion models. In Sung and Tokekar (2017), a PHD filter-based approach is proposed for simultaneous search and tracking of an unknown number of moving targets by a single robot with a limited sensing field of view. There have also been works on multi-target search and tracking using multi-robot systems with communication networks that are always connected Kamath *et al.* (2007); Dames and Kumar (2013); Dames (2019) and that use decentralized controllers to maintain connectivity Hung *et al.* (2016). In real-world applications, constraints on the robots' communication ranges limit the area that the robots can explore.

This chapter addresses the problem of tracking multiple targets without requiring a connected communication network using a multi-robot search strategy with the same probabilistic motion model and inter-robot communication constraints as in Chapter 2. We consider scenarios in which the quadrotors move according to a DTDS Markov chain model on a finite spatial grid, as illustrated in Figure 3.1, while searching for multiple stationary targets. A robot detects the presence of a target by obtaining sensor measurements of target's states. We assume that the robots' sensors have limited fields of view and that the robots share information about the targets only with other robots within their local communication range. We model the multi-robot target tracking procedure as a *renewal-reward process*, in which the reward is defined as the estimated number of targets and the targets' states, such as their spatial locations.

The remainder of the chapter, is organized as follows. Section 3.2 presents the problem statement. Section 3.3 discusses the DTDS Markov chain motion model that

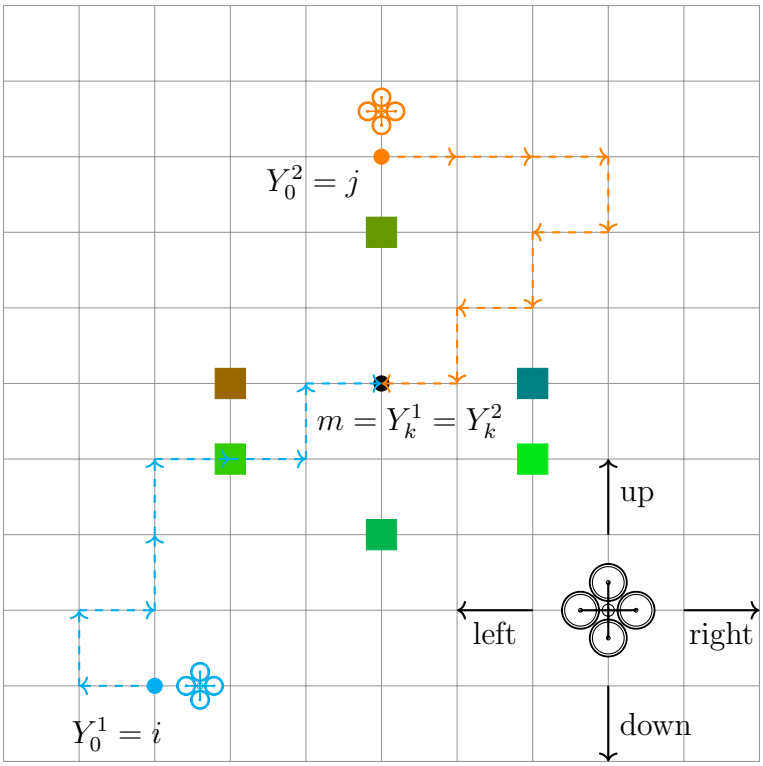


Figure 3.1: Illustration of Our Multi-Robot Multi-Target Tracking Strategy, Showing Sample Paths for Two Quadrotors (Orange and Blue) on a Square Grid. The Quadrotors Search the Environment for a Set of Static Targets, Represented by the Squares, As They Perform a Random Walk on the Grid.

the robots follow. We then provide a brief discussion on the theory of renewal-reward processes and their application to our problem in Section 3.4. In Section 3.5, we present the Random Finite Set formulation and its first-order moment, the Probability Hypothesis Density (PHD) filter, as an estimation framework for detecting and tracking multiple targets. In Section 3.5.1, we describe the Gaussian Mixture approximation of the PHD filter from (Vo and Ma, 2006). We then validate our strategy with simulations in Section 3.6 and finally state our conclusions in Section 3.7.

3.2 Problem Formulation

We consider an unknown, bounded environment that contains a finite, non-zero number of static targets, indexed by the set $\mathcal{I} \subset \mathbb{Z}_+$. The environment is discretized

into a square grid, and the four vertices of each grid cell are referred to as *nodes*. Let $\mathcal{S} \subset \mathbb{Z}_+$ denote the set of S nodes, and let $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ be an undirected graph associated with this grid, where $\mathcal{V}_s$ is the set of nodes and $\mathcal{E}_s$ is the set of edges (i, j) that signify the pairs of nodes $i, j \in \mathcal{V}_s$ between which the quadrotors can travel. A group of N quadrotors, indexed by the set $\mathcal{N}$, explores the environment using a random walk strategy: each quadrotor performs a random walk on the grid, moving from its current node i to an adjacent node j with transition probability p_{ij} at each discrete time k. We assume that each quadrotor is able to localize itself in this environment; i.e., that it knows which node it currently occupies. We also assume that quadrotors can communicate with one another only if they occupy the same node. We also assume that the quadrotors have perfect localization.

The number of targets estimated by each quadrotor is updated at every time step k. Let the i^{th} target detected by quadrotor a_j at time k be $m_{i,k}^{a_j} \in \mathbb{R}_+$ which is a tuple, composed of the *state* of the target, which is a time-varying property of the target like its location within the quadrotor's field of view (FoV), the pixels that it occupies in the quadrotor's camera image, and a unique identification label. Let $\mathcal{M}_k^{a_i} = \{m_{1,k}^{a_i}, \ldots, m_{n_m,k}^{a_i}\}$ be the set of states of all targets detected by the quadrotor a_j at time k, where n_m is the maximum number of features that a quadrotor can detect simultaneously. The value of n_m is limited by the computational capabilities and the available memory on the robot. As the quadrotor explores the environment, the number of targets that it detects and their states vary, as new targets appear in the FoV of the quadrotor and existing targets disappear. An *observation set* obtained by a quadrotor at a particular time consists of both measurements that are associated with actual targets and measurements arising from clutter. The objective of multi-target tracking is to jointly estimate, at each time step, the number of targets and the targets' states from a series of noisy and cluttered observation sets. The concept

of a *random finite set (RFS)* is useful for formulating this problem, since within the FoV of a quadrotor, the number of targets and their states are time-varying and not completely known. A random finite set, as defined in (Mahler, 2007), is a set with a random number of elements which are themselves random. In other words, a RFS is a random variable whose possible values are unordered finite sets. A computationally tractable approach to set-based estimation is to utilize the first statistical moment of an RFS, known as the Probability Hypothesis Density (PHD) or its *intensity function*, for multi-target tracking. We propose to use the Gaussian Mixture formulation of the PHD filter (GM-PHD) for each quadrotor, as it is less computationally expensive than the particle filter implementation.

Figure 3.1 illustrates our multi-target tracking approach with two quadrotors and six stationary targets. The quadrotors explore the grid according to the random walk motion model defined in Section 3.3, and they estimate the number of targets and their positions within their limited sensing FoV using the GM-PHD filter described in Section 3.5.1. Sample trajectories are shown for each quadrotor as a sequence of arrows that indicate its direction of motion. At time step k, the first *renewal epoch*, the quadrotors meet at node m and exchange *rewards*, defined as each quadrotor's estimates of the number of targets that it has detected up until time k and their positions, as described in Section 3.4. The implementation of this strategy is described in pseudocode in Algorithm 2 and Algorithm 3 and Algorithm 4. We extract only unique target states during simulation by using set union methods, as described in Algorithm 5.

3.3 Robot Motion Model

Let $Y_k^{a_i} \in \mathcal{S}$ be the random variable that represents the location of quadrotor a_i at time k on the spatial grid. For each quadrotor a_i, the probability mass function

Algorithm 2: Control Strategy for Robot $a_i \in \mathcal{N}$

Step 0: Initialization

a_i, $J_\gamma^{(a_i)}$, $\mu_\gamma^{(a_i)}$, $P_\gamma^{(a_i)}$, $Y_0^{(a_i)}$, F_{k-1}, Q_{k-1}, H_k, R_k,,

$\kappa_0^{(a_i)}(z)$, $p_{\mathbf{S}}$, $p_{\mathbf{D}}$, $w_0^{(a_i)}$, $\mathcal{M}_0^{(\cdot)}$

Step 1: Random Walk

$[Y_k^{(a_i)}] = \text{MarkovRandomWalk}(Y_{k-1}^{(a_i)})$;

Step 2: GM-PHD Filter

a *Predicted State Components*

b *Updated State Components*

c *Pruning and Merging Components*

d *Multi-target State Extraction*

$\pi_k \in \mathbb{R}^{1\times S}$ of $Y_k^{a_i}$ evolves according to a discrete-time discrete-space (DTDS) Markov chain given by:

$$\pi_{k+1} = \mathbf{P}\pi_k, \tag{3.1}$$

where the *state transition matrix* $\mathbf{P} \in \mathbb{R}^{S\times S}$ has elements $p_{ij} \in [0, 1]$ at row i and column j. The time evolution of the probability mass function of $Y_k^{a_i}$ is expressed using the Markov property as follows:

$$Pr(Y_{k+1}^{a_i} = j_{k+1}|Y_k^{a_i} = j_k, \ldots, Y_0^{a_i} = j_0) \quad = \quad Pr(Y_{k+1}^{a_i} = j_{k+1}|Y_k^{a_i} = j_k), \tag{3.2}$$

where j_k is a specific node in the spatial grid that the quadrotor may occupy at time k. In other words, Equation (3.2) states that the future location of the quadrotor depends only on its current location and is statistically independent of any previous locations. We assume that the DTDS Markov chain is time-homogeneous, which implies that $Pr(Y_{k+1}^{a_i} = j_{k+1}|Y_k^{a_i} = j_k)$ is same for all quadrotors at all time steps.

Algorithm 3: Components of GMPHD Filter for $a_i \in \mathcal{N}$

a *Predicted State Components*

Apply steps 1 and 2 from Table 1 in (Vo and Ma, 2006)

$J_{k-1}^{(a_i)}$=$J_{\gamma}^{(a_i)}$; $w_{k-1}^{(a_i)}$=$w_{\gamma}^{(a_i)}$; $\mu_{k-1}^{(a_i)} = \mu_{\gamma}^{(a_i)}$

$[w_{k|k-1}^{(a_i)}, \mu_{k|k-1}^{(a_i)}, P_{k|k-1}^{(a_i)}, J_{k|k-1}^{(a_i)}] =$
predictGMPHD(J_{k-1},$w_{k-1}^{(a_i)}$,$\mu_{k-1}^{(a_i)}$,$P_{k-1}^{(a_i)}$,F_{k-1},Q_{k-1},$p_{\mathbf{S}}$);

b *Updated State Components*

Apply steps 3 and 4 from Table 1 in (Vo and Ma, 2006)

$[w_k$,μ_k,P_k,$J_k]$=

updateGMPHD(H_k,$\mu_{k|k-1}$,R_k,$P_{k|k-1}$,$p_{\mathbf{D}}$,$J_{k|k-1}$,$w_{k|k-1}$, $\kappa_k^{(a_i)}(z)$,$Z_k^{(a_i)}$);

c *Pruning and Merging Components*

Apply all steps from Table 2 in (Vo and Ma, 2006)

d *Multi-target State Extraction*

Apply all steps from Table 3 in (Vo and Ma, 2006) with $thresh_{state} = 0.5$

$[\hat{X}_k^{(a_i)}$,$\hat{w}_k^{(a_i)}$,$\hat{P}_k^{(a_i)}] =$

extractMTStateGMPHD($w_k^{(a_i)}$,$\mu_k^{(a_i)}$,$P_k^{(a_i)}$,$thresh_{state}$);

$\mathcal{M}_k^{(a_i)} = \mathcal{M}_{k-1}^{(a_i)} \bigcup \hat{X}_k^{(a_i)}$;

Thus, the entries of $\mathbf{P}$ can be defined as follows:

$$p_{ij} = Pr(Y_{k+1}^{a_i} = j_{k+1} | Y_k^{a_i} = j_k), \ \forall j_k \in S, \ k \in \mathbb{Z}_{\geq 0}, \ a_i \in \mathcal{N}. \tag{3.3}$$

Assuming that each quadrotor chooses its next position from a uniform random distribution, we can compute the entries of $\mathbf{P}$ as follows:

$$p_{ij} = \begin{cases} \frac{1}{d_i+1}, & (i,j) \in \mathcal{E}_s, \\ 0, & \text{otherwise}, \end{cases} \tag{3.4}$$

where d_i is the degree of the node $i \in \mathcal{S}$. Since each entry $p_{ij} \geq 0$, we use the notation $\mathbf{P} \geq 0$. We see that $\mathbf{P}^m \geq 0$ for $m \geq 1$. Hence, $\mathbf{P}$ is a *non-negative matrix*. Then, from Theorem 5 in (Grimmett and Stirzaker, 2001), we can say that $\mathbf{P}$ is a stochastic matrix.

We define Equation (3.1) as the *spatial Markov chain*. From the construction of the spatial Markov chain, every quadrotor has a positive probability of moving from node $i \in S$ to any node $j \in S$ in a finite number of time steps. Thus, the Markov chain is said to be *irreducible*, and consequently, $\mathbf{P}$ is an *irreducible* matrix. Now applying Lemma 8.4.4 in (Horn and Johnson, 1990), we know that there exists a real unique positive left eigenvector of $\mathbf{P}$. Since $\mathbf{P}$ is a stochastic matrix, we have that $\rho(\mathbf{P}) = 1$, where $\rho(\mathbf{P})$ denotes the spectral radius of $\mathbf{P}$. Thus, we can conclude that this real unique positive left eigenvector is the *stationary distribution* associated with the spatial Markov chain. Since we have shown that the Markov chain is irreducible and has a stationary distribution π that satisfies $\pi\mathbf{P} = \pi$, we can conclude from Theorem 21.12 in (Levin and Peres, 2017) that the Markov chain is *positive recurrent*. Thus, all states in the Markov chain are positive recurrent, which implies that each quadrotor will keep visiting every state on the finite spatial grid infinitely often. We will use this result to prove results on the associated renewal-reward process, which is discussed next.

3.4 Renewal-Reward Process

We now define a random variable $\tau_j^{a_i} \in \mathbb{R}_{\geq 0}$ as the j^{th} interval between two successive times at which quadrotor a_i and any another quadrotor occupy the same node. This time interval is referred to as the *inter-arrival time*. A *renewal epoch* is a time instant at which two quadrotors meet at the same node. For each quadrotor a_i, we define the counting process $T^{a_i}(k) \in \mathbb{Z}_{\geq 0}$ as the number of times a_i has met any other quadrotor by time k. At each renewal epoch, quadrotor a_i updates its *reward*, defined as the number of all detected targets and their locations, with the number of targets and locations detected by the quadrotor(s) that occupies its current node and transmits this information to a_i. We use the definition of a *renewal process* given by Definition 7.1 in (Ross, 2014). If the sequence of non-negative random variables $\{\tau_0^{a_i}, \tau_1^{a_i}, \ldots, \}$ is independent and identically distributed, then the counting process $T^{a_i}(k)$ is said to be a renewal process. We demonstrate that $T^{a_i}(k)$ is a renewal process at the end of this section.

For a renewal process having inter-arrival times $\tau_0^{a_i}, \tau_1^{a_i}, \ldots$, we define $S_n^{a_i} = \sum_{j=1}^{n} \tau_j^{a_i}$ as the n^{th} renewal epoch, with $S_0^{a_i} = 0$ for all $a_i \in \mathcal{N}$. From the definition of a renewal process, we can infer that the number of renewal epochs by time k is greater than or equal to n if and only if the n^{th} renewal epoch occurs before or at time k; that is,

$$T^{a_i}(k) \geq n \quad \Leftrightarrow \quad S_n^{a_i} \leq k. \tag{3.5}$$

Now consider that at each renewal epoch, quadrotor a_i receives a reward. The reward $R_n^{a_i}$ earned by quadrotor a_i when the n^{th} renewal occurs is defined as follows:

$$R_n^{a_i} = \mathcal{M}_k^{a_i} \bigcup_{a_j \neq a_i} \mathcal{M}_k^{a_j}, \quad Y_k^{a_i} = Y_k^{a_j} \text{ and } a_j \in \mathcal{N}. \tag{3.6}$$

Equation (3.5) and Equation (3.6) together define a *renewal-reward process*. Each quadrotor a_i calculates $\mathcal{M}_k^{a_i}$ by estimating the number of targets and their spatial

distribution using a PHD filter. In Section 3.5, we describe some fundamental theory on target detection and tracking using this type of filter.

Given the quadrotor motion model defined in Section 3.3, we can model the dynamics of all the quadrotors' movements on the spatial grid by a composite Markov chain with states $\psi_k = (Y_k^{a_1}, Y_k^{a_2}, \ldots, Y_k^{a_N}) \in \mathcal{H}$, where $\mathcal{H} = \mathcal{S}^{\mathcal{N}}$. Note that $S = |\mathcal{S}|$ and $|\mathcal{H}| = S^N$. We now define another undirected graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ associated with this composite Markov chain. The vertex set $\hat{\mathcal{V}}$ is a set of all possible realizations $\hat{\imath} \in \mathcal{H}$ of ψ_k. Here $\hat{\imath}(a_l)$ represents the a_l^{th} entry of $\hat{\imath}$, which corresponds to the spatial node $i \in \mathcal{S}$ occupied by robot $a_l \in \mathcal{N}$ and $l \in \mathcal{I}$. We define the edge set $\hat{\mathcal{E}}$ of graph $\hat{\mathcal{G}}$ as follows: $(\hat{\imath}, \hat{\jmath}) \in \hat{\mathcal{E}}$ if and only if $(\hat{\imath}(a_l), \hat{\jmath}(a_l)) \in \mathcal{E}_s$ for all robots $a_l \in \mathcal{N}$. Let $\mathbf{Q} \in \mathbb{R}^{|\mathcal{H}| \times |\mathcal{H}|}$ be the state transition matrix associated with this composite Markov chain. An element of $\mathbf{Q}$, denoted by $q_{\hat{\imath}\hat{\jmath}}$, is the probability that in the next time step, each robot a will move from spatial node $\hat{\imath}(a_l)$ to node $\hat{\jmath}(a_l)$. These elements are computed from the transition probabilities defined by Equation (3.4) as follows:

$$q_{\hat{\imath}\hat{\jmath}} = \prod_{a_l=1}^{N} p_{\hat{\imath}(a_l)\hat{\jmath}(a_l)}, \qquad \forall \hat{\imath}, \hat{\jmath} \in \mathcal{H} \;\; \& \;\; l \in \mathcal{I}. \tag{3.7}$$

As an illustration, consider a set of two robots, $\mathcal{N} = \{a_1, a_2\}$, that move on the graph $\mathcal{G}_s$ shown in Figure 3.2. The robots can stay at their current node in the next time step or travel between nodes i and j and between nodes j and l, but they cannot travel between nodes i and l. Figure 3.3 shows a subset of the resulting composite graph $\hat{\mathcal{G}}$, which has the set of nodes

$$\hat{\mathcal{V}} = \{(i,i), (i,j), (i,l), (j,i), (j,j), (j,l), (l,i), (l,j), (l,l)\}$$

Each node in $\hat{\mathcal{V}}$ is labeled by a single index $\hat{\imath}$, e.g., $\hat{\imath} = (i,j)$, with $\hat{\imath}(a_1) = i$ and $\hat{\imath}(a_2) = j$. Given the connectivity of the spatial grid defined by $\mathcal{E}_s$, we can for example identify $((i,j),(i,l))$ as an edge in $\hat{\mathcal{E}}$, but not $((i,j),(l,l))$. Since $N = 2$ and

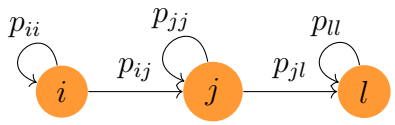


Figure 3.2: The Arrows Signify Directed Edges Between Pairs of Distinct Nodes or Self-Edges. The Edge Set Is $\mathcal{E}_s = \{(i,i),(j,j),(l,l),(i,j),(j,l)\}$.

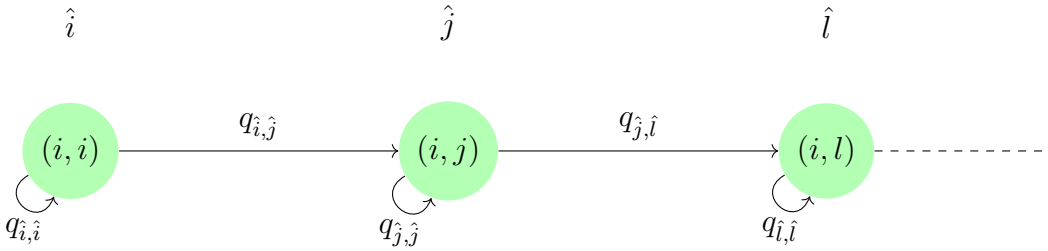


Figure 3.3: A Subset of the Composite Graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ for Two Agents That Move on the Graph $\mathcal{G}_s$ Shown In $\mathcal{G}_s$ shown in Figure 3.2.

$S = 3$, we have that $|\mathcal{H}| = 3^2 = 9$. For each $\hat{\imath}, \hat{\jmath} \in \hat{\mathcal{V}}$, we can compute the transition probabilities in $\mathbf{Q} \in \mathbb{R}^{9\times 9}$ from Equation (3.7) as:

$$q_{\hat{\imath}\hat{\jmath}} = Pr\left(\psi_{k+1} = \hat{\jmath} \mid \psi_k = \hat{\imath}\right) = p_{\hat{\imath}(a_1)\hat{\jmath}(a_1)} p_{\hat{\imath}(a_2)\hat{\jmath}(a_2)}, \quad k \in \mathbb{Z}_+. \tag{3.8}$$

We now prove that $T^{a_i}(k)$ is a renewal process.

Theorem 3.4.1. *$T^{a_i}(k)$ is a renewal process on the composite Markov chain ψ_k.*

Proof. Suppose that an initial time instant k_0, the locations of all N robots on the spatial grid are represented by the node $\hat{\imath} \in \hat{\mathcal{V}}$. Consider another set of robot locations at time $k_0 + k$, where $k > 0$, represented by the node $\hat{\jmath} \in \hat{\mathcal{V}}$. The transition of the robots from configuration $\hat{\imath}$ to configuration $\hat{\jmath}$ in k time steps corresponds to a random walk of length k on the composite Markov chain ψ_k from node $\hat{\imath}$ to node $\hat{\jmath}$. It also corresponds to a random walk by each robot a_i on the spatial grid from node $\hat{\imath}(a_i)$ to node $\hat{\jmath}(a_i)$ in k time steps. By construction, the graph $\mathcal{G}_s$ is strongly connected and each of its nodes has a self-edge. Therefore, there exists a discrete time $n > 0$ such that, for each robot $a_i \in \mathcal{N}$, there exists a random walk on the spatial grid from node $\hat{\imath}(a_i)$ to node $\hat{\jmath}(a_i)$ in n time steps. Consequently, there always exists a random

walk of length n on the composite Markov chain ψ_k from node $\hat{\imath}$ to node $\hat{\jmath}$. Therefore, ψ_k is an irreducible Markov chain. All states of an irreducible Markov chain belong to a single communication class. In this case, all states are *positive recurrent*. As a result, ψ_k is *positive recurrent*. Thus, each state in ψ_k is visited infinitely often from all other states in ψ_k. A state with this property is said to *regenerate* (or *renew*) infinitely often. We can then conclude from Proposition 67 in (Serfozo, 2009) that $T^{a_i}(k)$ is a regenerative process on ψ_k. Since every regenerative process is a renewal process, $T^{a_i}(k)$ is a renewal process. □

Algorithm 4: Renewal-Reward Computation for Robots $(a_i, A_j) \in \mathcal{N}$

Given: $\mathcal{M}_k^{(a_i)}$, $\mathcal{M}_k^{(a_j)}$, $Y_k^{(a_i)}$, $Y_k^{(a_j)}$

for $k \in 1 : t_{final}$ **do**

  for $n_1 \in 1 : |\mathcal{N}|$ **do**

    $l = 1$; $n = 1$;

    for $n_2 \in n_1 + 1 : |\mathcal{N}|$ **do**

      if $Y_k^{(n_1)} = Y_k^{(n_2)}$ **then**

        $R_n^l = \mathcal{M}_k^{(n_1)} \bigcup \mathcal{M}_k^{(n_2)}$;

        $l = l + 1$; $n = n + 1$;

      end

    end

  end

end

3.5 Probability Hypothesis Density Filter

Let $M_k^{a_i} \leq n_m$ be the number of targets identified by quadrotor a_i at time step k. Suppose that at time $k-1$, the target states are $x_{k-1,1}^{a_i},\ x_{k-1,2}^{a_i},\ \ldots,\ x_{k-1,M_{k-1}^{a_i}}^{a_i} \in \mathcal{X}$,

Algorithm 5: Exchange of Set of Estimated States Between Robots $(a_i, A_j) \in \mathcal{N}$

Given: $X_{k-1}^{(a_i,a_j)}$, $\hat{X}_k^{(a_i)}$, $\hat{X}_k^{(a_j)}$

for $l_1 \in 1 : size(|\hat{X}_k^{(a_i)}|, 2)$ **do**

  for $l_2 \in 1 : size(|\hat{X}_k^{(a_j)}|, 2)$ **do**

    $X_{temp} = \hat{X}_{k,l_1}^{(a_i)} \bigcup \hat{X}_{k,l_2}^{(a_j)}$;

    if $X_{temp} \notin X_{k-1}^{(a_i,a_j)}$ **then**

      $X_k^{(a_i,a_j)} = X_{k-1}^{(a_i,a_j)} \bigcup X_{temp}$;

    end

    else

      $X_k^{(a_i,a_j)} = X_{k-1}^{(a_i,a_j)}$;

    end

  end

  $\mathcal{M}_k^{(a_i,a_j)} = X_k^{(a_i,a_j)}$;

end

where $\mathcal{X}$ is the set of target states. At the next time step, some of these targets might disappear from the quadrotor's field of view (FoV), and new targets may appear. This results in $M_k^{a_i}$ new states $x_{k,1}^{a_i}, x_{k,2}^{a_i}, \ldots, x_{k,M_k^{a_i}}^{a_i}$. Note that the order in which the states are represented has no significance in the RFS multi-target tracking formulation. The quadrotor a_i makes $N_k^{a_i}$ measurements $z_{k,1}^{a_i}, \ldots, z_{k,N_k^{a_i}} \in \mathcal{Z}$ at time k, where $\mathcal{Z}$ is the set of measurements. The order in which the measurements are made is not significant. The states of the targets identified by quadrotor a_i at time k (i.e., the multi-target state) and the measurements obtained by the quadrotor at time k can both be represented as finite sets:

$\mathbf{S}^{a_i}_{k|k-1}(\xi)$: RFS of targets with previous state ξ at time $k-1$ that survive at time k

$\mathbf{B}^{a_i}_{k|k-1}(\xi)$: RFS of targets spawned at time k from targets with previous state ξ at time $k-1$

$\mathbf{\Gamma}^{a_i}_k$: RFS of targets that are spontaneously born at time k

$$X^{a_i}_k = \{x^{a_i}_{k,1}, \ldots, x^{a_i}_{k,M^{a_i}_k}\} \in \mathcal{F}(\mathcal{X}), \tag{3.9}$$

$$Z^{a_i}_k = \{z^{a_i}_{k,1}, \ldots, z_{k,N^{a_i}_k}\} \in \mathcal{F}(\mathcal{Z}), \tag{3.10}$$

where $\mathcal{F}(\mathcal{X})$ is the multi-target state space and $\mathcal{F}(\mathcal{Z})$ is the measurement space. For a quadrotor a_i, given multi-target state $X^{a_i}_{k-1}$ at time $k-1$, each $x^{a_i}_{k-1} \in X^{a_i}_{k-1}$ either continues to exist (survives) at time k with probability $p^{a_i}_{\mathbf{S},k}(x^{a_i}_{k-1})$ or disappears (dies) at time k with probability $1-p^{a_i}_{\mathbf{S},k}(x^{a_i}_{k-1})$. The conditional probability density at time k of a transition from state $x^{a_i}_{k-1}$ to state $x^{a_i}_k$ is given by $f^{a_i}_{k|k-1}(\cdot|\cdot)$.

We now define the RFS model for the time evolution of the multi-target state, which incorporates motion of the targets relative to the quadrotor, appearance (birth) of targets, and disappearance (death) of targets:

$$X^{a_i}_k = \left[\bigcup_{\xi \in X^{a_i}_{k-1}} \mathbf{S}^{a_i}_{k|k-1}(\xi)\right] \bigcup \left[\bigcup_{\xi \in X^{a_i}_{k-1}} \mathbf{B}^{a_i}_{k|k-1}(\xi)\right] \bigcup \mathbf{\Gamma}^{a_i}_k \tag{3.11}$$

At each time step, a quadrotor a_i detects a target with state $x^{a_i}_k \in X^{a_i}_k$ with probability $p^{a_i}_{\mathbf{D},k}(\cdot)$, or misses it with probability $1-p^{a_i}_{\mathbf{D},k}(\cdot)$. The conditional probability of obtaining a measurement $z^{a_i}_k \in Z^{a_i}_k$ from $x^{a_i}_k$ is characterized by the multi-target likelihood function, $g^{a_i}_k(\cdot|\cdot)$. We can now define the RFS model for the time evolution of the multi-target measurement, which incorporates measurements of actual targets

along with clutter:

$$Z_k^{a_i} = \mathbf{K}_k^{a_i} \bigcup \left[\bigcup_{x \in X_k^{a_i}} \mathbf{\Theta}_k^{a_i}(x) \right] \tag{3.12}$$

$\mathbf{K}_k^{a_i}$: RFS of measurements arising from clutter at time k

$\mathbf{\Theta}_k^{a_i}(x)$: RFS of measurements of the multi-target state $X_k^{a_i}$ at time k

The multi-target Bayes filter propagates the multi-target posterior density $p_k^{a_i}(\cdot \mid Z_{1:k}^{a_i})$ in time via recursion as:

$$p_{k|k-1}^{a_i}(X_k^{a_i}|Z_{1:k-1}^{a_i}) = \int_{X^{a_i} \in \mathcal{F}(\mathcal{X})} f_{k|k-1}^{a_i}(X_k^{a_i}|X^{a_i}) p_{k-1}^{a_i}(X^{a_i}|Z_{1:k-1}^{a_i}) \mu_s(dX^{a_i}) \tag{3.13}$$

$$p_k^{a_i}(X_k^{a_i}|Z_{1:k}^{a_i}) = \frac{g_k^{a_i}(Z_k^{a_i}|X_k^{a_i}) p_{k|k-1}^{a_i}(X_k^{a_i}|Z_{1:k-1}^{a_i})}{\int_{X^{a_i} \in \mathcal{F}(\mathcal{X})} g_k(Z_k^{a_i}|X^{a_i}) p_{k|k-1}^{a_i}(X^{a_i}|Z_{1:k-1}^{a_i}) \mu_s(dX^{a_i})} \tag{3.14}$$

where μ_S is a suitable reference measure on $\mathcal{F}(\mathcal{X})$ of target states $X^{a_i} \in \mathcal{F}(\mathcal{X})$, $g_k^{a_i}(\cdot|\cdot)$ represents the multi-target likelihood function, and $f_{k|k-1}^{a_i}(\cdot|\cdot)$ represents the multi-target transition density. For further details, see (Clark *et al.*, 2006; Vo and Ma, 2006).

We will approximate the integrals above using the framework of the probability hypothesis density (PHD) filter, with the assumptions that: (1) each target evolves and generates observations independently of the others; (2) clutter is Poisson distributed and independent of target-originated measurements; (3) the multi-target RFS is Poisson distributed. For a RFS $X^{a_i} \in \mathcal{X}$ with probability distribution $p^{a_i}(\cdot)$, there is a non-negative function v on $\mathcal{X}$, defined as the *intensity function*, such that for each region $\mathcal{S} \subset \mathcal{X}$,

$$\int |X^{a_i} \cap \mathcal{S}| p^{a_i}(dX) = \int_{\mathcal{S}} v(x) dx. \tag{3.15}$$

Then we can model the posterior intensity and its recursion as follows:

$$v_{k|k-1}^{a_i}(x) = \int p_{\mathbf{S},k}^{a_i}(\xi) f_{k|k-1}^{a_i}(x|\xi) v_{k-1}^{a_i}(\xi) d\xi + \int \beta_{k|k-1}^{a_i}(x|\xi) v_{k-1}^{a_i}(\xi) d\xi + \gamma_k^{a_i}(x), \quad (3.16)$$

$$v_k^{a_i}(x) = [1 - p_{\mathbf{D},k}^{a_i}(x)] v_{k|k-1}^{a_i}(x) + \sum_{z \in Z_k^{a_i}} \frac{p_{\mathbf{D},k}^{a_i}(x) g_k^{a_i}(z|x) v_{k|k-1}^{a_i}(x)}{\kappa_k^{a_i}(z) + \int p_{\mathbf{D},k}^{a_i}(\xi) g_k^{a_i}(z|\xi) v_{k|k-1}^{a_i}(\xi)}. \quad (3.17)$$

In these equations, $v_k^{a_i}$ and $v_{k|k-1}^{a_i}$ denote the intensities associated with, respectively, the multi-target posterior density $p_k^{a_i}(\cdot|\cdot)$ and the multi-target predicted density $p_{k|k-1}^{a_i}(\cdot|\cdot)$ that are defined by the recursion in Equation (3.13) and Equation (3.14). The function $\gamma_k^{a_i}(\cdot)$ is the intensity of the RFS $\mathbf{\Gamma}_k^{a_i}$, $\beta_{k|k-1}^{a_i}(\cdot|\xi)$ is the intensity of the RFS $\mathbf{B}_{k|k-1}(\xi)$, and $\kappa_k^{a_i}(\cdot)$ is the intensity of the RFS $\mathbf{K}_k^{a_i}$. The quadrotor a_i can estimate the number of targets as:

$$\hat{N} = \int v(x) dx. \quad (3.18)$$

The estimate $\hat{N}$ is used to update the number of elements of $\mathcal{M}_k^{a_i}$, and the intensity $v_k^{a_i}(x)$ computed from Equation (3.17) is used to update the states of the $\hat{N}$ targets. Then each element of $\mathcal{M}_k^{a_i}$ is represented as the following tuple:

$$m_{k,l}^{a_i} = \langle\ l,\ v_k^{a_i}(x)\ \rangle, \quad (3.19)$$

where l is a label for the tracked target, such as one of its properties, e.g. its color, shape, size or its position in the environment.

3.5.1 Gaussian Mixture PHD Filter

The PHD filter as described in Equation (3.16) and Equation (3.17) does not admit a closed-form solution in general, and the numerical integration suffers from the curse of dimensionality. Thus, for implementation purposes, we consider a sub-optimal solution of the PHD filter that models approximates it as a mixture of Gaussians, as described in (Vo and Ma, 2006). The Gaussian Mixture PHD (GM-PHD) filter provides a closed-form solution to the PHD filter under the following assumptions:

A.1 Each target generates observations independently of the others.

A.2 The clutter process is Poisson distributed and is independent of target-generated measurements.

A.3 Each target's state evolves according to a linear model with Gaussian process noise, and each quadrotor's sensor has a linear measurement model with Gaussian sensor noise, i.e.

$$f_{k|k-1}^{a_i}(x|\xi) = \mathbb{N}(x; F_{k-1}, Q_{k-1}), \tag{3.20}$$

$$g_k^{a_i}(z|\xi) = \mathbb{N}(z; H_k x, R_k), \tag{3.21}$$

where the notation $\mathbb{N}(\cdot\ ; \mu, \sigma)$ denotes a Gaussian density with mean μ and covariance σ, F_{k-1} is the state transition matrix, Q_{k-1} is the process noise covariance, H_k is the observation or measurement matrix, and R_k is the sensor noise covariance.

A.4 The detection probability is state-dependent and is modeled as

$$p_{\mathbf{D},k}^{a_i}(x) = \begin{cases} p_{\mathbf{D}} & ||q_k^{a_i} - x|| \in \mathcal{B}_r(q_k^{a_i}), \\ 0 & \text{otherwise}, \end{cases} \tag{3.22}$$

where $q_k^{a_i}$ denotes the grid coordinates of robot a_i at time k and $\mathcal{B}_r(q_k^{a_i})$ represents the FoV of the sensor on robot a_i, which we model as a disc of radius r centered at the robot location $q_k^{a_i}$. The survival probability is assumed to be constant:

$$p_{\mathbf{S},k}^{a_i}(x) = p_{\mathbf{S},k}. \tag{3.23}$$

A.5 The birth and spawning intensities are modeled as Gaussian mixtures of the form

$$\gamma_k^{a_i}(x) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^{(i)} \mathbb{N}(x; \mu_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}), \tag{3.24}$$

$$\beta_{k|k-1}^{a_i}(x|\xi) = \sum_{j=1}^{J_{\beta,k}} w_{\beta,k}^{(j)} \mathbb{N}(x; F_{\beta,k-1}^{(j)}\xi + d_{\beta,k-1}^{(j)}, Q_{\beta,k-1}^{(j)}), \tag{3.25}$$

where $J_{\gamma,k}$, $w_{\gamma,k}^{(i)}$, $\mu_{\gamma,k}^{(i)}$, and $P_{\gamma,k}^{(i)}$ are known parameters of the birth intensity, and $J_{\beta,k}$, $w_{\beta,k}^{(i)}$, $F_{\beta,k-1}^{(j)}$, $d_{\beta,k-1}^{(j)}$, $Q_{\beta,k-1}^{(j)}$, and $P_{\beta,k-1}^{(j)}$ are known parameters of the spawn intensity of a target with state ξ at time $k-1$. For more details on the parameters, please refer to (Vo and Ma, 2006).

Using the above assumptions, we can rewrite Equation (3.16) and Equation (3.17) as follows. The intensity associated with the multi-target predicted density can be approximated as a Gaussian mixture:

$$v_{k|k-1}^{a_i}(x) = v_{\mathbf{S},k|k-1}^{a_i}(x) + v_{\beta,k|k-1}^{a_i}(x) + \gamma_k^{a_i}(x), \tag{3.26}$$

where

$$v_{\mathbf{S},k|k-1}^{a_i}(x) = p_{\mathbf{S},k} \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathbb{N}(x; \mu_{\mathbf{S},k|k-1}^{(i)}, P_{\mathbf{S},k|k-1}^{(i)}), \tag{3.27}$$

$$\mu_{\mathbf{S},k|k-1}^{(i)} = F_{k-1}\mu_{k-1}^{(i)}, \tag{3.28}$$

$$P_{\mathbf{S},k|k-1}^{(i)} = Q_{k-1} + F_{k-1}P_{k-1}^{(i)}F_{k-1}^T, \tag{3.29}$$

$$v_{\beta,k|k-1}^{a_i}(x) = \sum_{i=1}^{J_{k-1}} \sum_{l=1}^{J_{\beta,k}} w_{k-1}^{(i)} w_{\beta,k}^{(l)} \mathbb{N}(x; \mu_{\beta,k|k-1}^{(i,l)}, P_{\beta,k|k-1}^{(i,l)}), \tag{3.30}$$

$$\mu_{\beta,k|k-1}^{(i,l)} = F_{\beta,k-1}^{(l)}\mu_{k-1}^{(i)} + d_{\beta,k-1}^{(l)}, \tag{3.31}$$

$$P_{\beta,k|k-1}^{(i,l)} = Q_{\beta,k-1}^{(l)} + F_{\beta,k-1}^{(l)}P_{\beta,k-1}^{(i)}(F_{\beta,k-1}^{(l)})^T, \tag{3.32}$$

in which J_{k-1}, $w_{k-1}^{(i)}$, $\mu_{k-1}^{(i)}$, and $P_{k-1}^{(i)}$ are known parameters of the intensity function at time $k-1$ (Vo and Ma, 2006).

Then the intensity associated with the multi-target posterior density can be approximated as a Gaussian mixture:

$$v_k^{a_i}(x) = [1 - p_{\mathbf{D},k}^{a_i}(x)]v_{k|k-1}^{a_i}(x) + \sum_{z \in Z_k^{a_i}} v_{\mathbf{D},k}^{a_i}(x; z), \tag{3.33}$$

where

$$v_{\mathbf{D},k}^{a_i}(x;z) = \sum_{j=1}^{J_{k|k-1}} w_k^{(j)}(z)\mathbb{N}(x;\mu_{k|k}^{(j)}(z), P_{k|k}^{(j)}), \tag{3.34}$$

$$w_k^{(j)}(z) = \frac{p_{\mathbf{D},k}^{a_i}(x) w_{k|k-1}^{(j)} \mathbb{N}(z; H_k\mu_{k|k-1}, R_k + H_k P_k H_k^T)}{\kappa_k^{a_i}(z) + p_{\mathbf{D},k}^{a_i}(x) \sum_{l=1}^{J_{k|k-1}} w_{k|k-1}^{(l)} \mathbb{N}(z; H_k\mu_{k|k-1}, R_k + H_k P_k H_k^T)}, \tag{3.35}$$

$$\mu_{k|k}^{(j)}(z) = \mu_{k|k-1}^{(j)}(z) + K_k^{(j)}(z - H_k\mu_{k|k-1}^{(j)}(z)), \tag{3.36}$$

$$P_{k|k}^{(j)} = [I - K_k^{(j)} H_k] P_{k|k-1}^{(j)}, \tag{3.37}$$

$$K_k^{(j)} = P_{k|k-1}^{(j)} H_k^T [H_k P_{k|k-1}^{(j)} H_k^T + R_k]^{-1}. \tag{3.38}$$

3.6 Simulation Results

In this section, we validate our approach with simulations in MATLAB. First, we model a scenario with a bounded environment with dimensions 5m $\times$ 5m that contains 3 stationary targets which must be located by 3 robots, as shown in Figure 3.4. The state of each target is defined as its $x-y$ position coordinates, $x = [p_x, p_y]^T$. A robot's sensor measurement of a target's state is modeled according to Equation (3.21). Each robot has a circular FoV of radius $r_{FOV} = 0.6$m, centered at the robot's position on the spatial grid. We assume that each robot is able to accurately localize itself on the grid, and that there are no obstacles present in the environment.

Since each agent has a limited FoV, We assume that the targets that are detected at time step k survive in the next time step with probability $p_{\mathbf{S},k} = 0.1$ for all robots. Since the targets are stationary, $F_{k-1} = \mathbf{I}_2$, the 2×2 identity matrix. We also set $Q_{k-1} = 0.2\mathbf{I}_2$. As the robots explore the environment, new targets might appear in their FoV. We account for this by allowing 4 new targets to be birthed at each time step, depending upon the robot's position on the grid, with weights $w_{\gamma,k} = (w_{\gamma,k}^{(i)})_{i=1}^4 = [0.1, 0.1, 0.1, 0.1]^T$. Thus, the birth intensity at each time step from Equation (3.24)

is modeled as

$$\gamma_k^{a_i}(x) = 0.1\mathbb{N}(x; \mu_{\gamma,k}^{(1)}, P_{\gamma,k}^{(1)}) + 0.1\mathbb{N}(x; \mu_{\gamma,k}^{(2)}, P_{\gamma,k}^{(2)}) + \\ 0.1\mathbb{N}(x; \mu_{\gamma,k}^{(3)}, P_{\gamma,k}^{(3)}) + 0.1\mathbb{N}(x; \mu_{\gamma,k}^{(4)}, P_{\gamma,k}^{(4)}) \tag{3.39}$$

where $P_\gamma^{(l)} = 0.5\mathbf{I}_2$ and

$$\mu_\gamma^{(l)} = \begin{bmatrix} p_{x,k}^{a_i} + r_{birth}\cos(\theta_l) \\ p_{y,k}^{a_i} + r_{birth}\sin(\theta_l) \end{bmatrix}, \tag{3.40}$$

in which $q_k^{a_i} = [p_{x,k}^{a_i} p_{y,k}^{a_i}]$ denotes the $x-y$ coordinates of robot a_i at time step k, corresponding to its current node $Y_k^{a_i}$; $r_{birth} = 0.8r_{FOV}$, so that the targets are birthed only near the boundary of FOV; and $\theta_l = [\pi/4,\ 3\pi/4,\ 5\pi/4,\ 7\pi/4]^T$ rad, the angles at which targets are likely to appear. We assume that there are no spawned targets. Each target is detected with a probability of $p_{\mathbf{D}} = 0.8$, and a quadrotor's observation of a target follows the measurement model Equation (3.21) with $H_k = \mathbf{I}_2$ and $R_k = 0.25\mathbf{I}_2$. The observations are immersed in clutter that can be modeled as a Poisson RFS $\mathbf{K}_k^{(\cdot)}$ with intensity $\kappa_k^{(\cdot)}(z) = \lambda_C A_s \mathbb{U}(z)$, where $\lambda_C = 3.98 \times 10^{-3}$ is the clutter intensity; A_s is the area of the sensor's circular FoV, which is approximately 1m^2; and $\mathbb{U}(z)$ is the uniform density over A_s.

We assume that all robots start at random positions on the grid and have no knowledge of the number of targets or their states (positions). The robots explore the environment according to the random walk model Equation (3.1). As the robots detect the targets, they recursively update their estimates of the number of targets and their positions using the GM-PHD framework described in Section 3.5.1. We set $T = 1 \times 10^{-3}$ as the pruning threshold and $U = 4$ as the merging distance threshold (see Table II in (Vo and Ma, 2006) for details on these parameters).

Figure 3.5 plots the inter-arrival times over time during 300 s of the simulation. Each inter-arrival time $\tau_j^{a_m a_n}$ ends at a renewal epoch, i.e. a time when any two robots a_m and a_n meet at a node, which can be identified in the figure as the time at

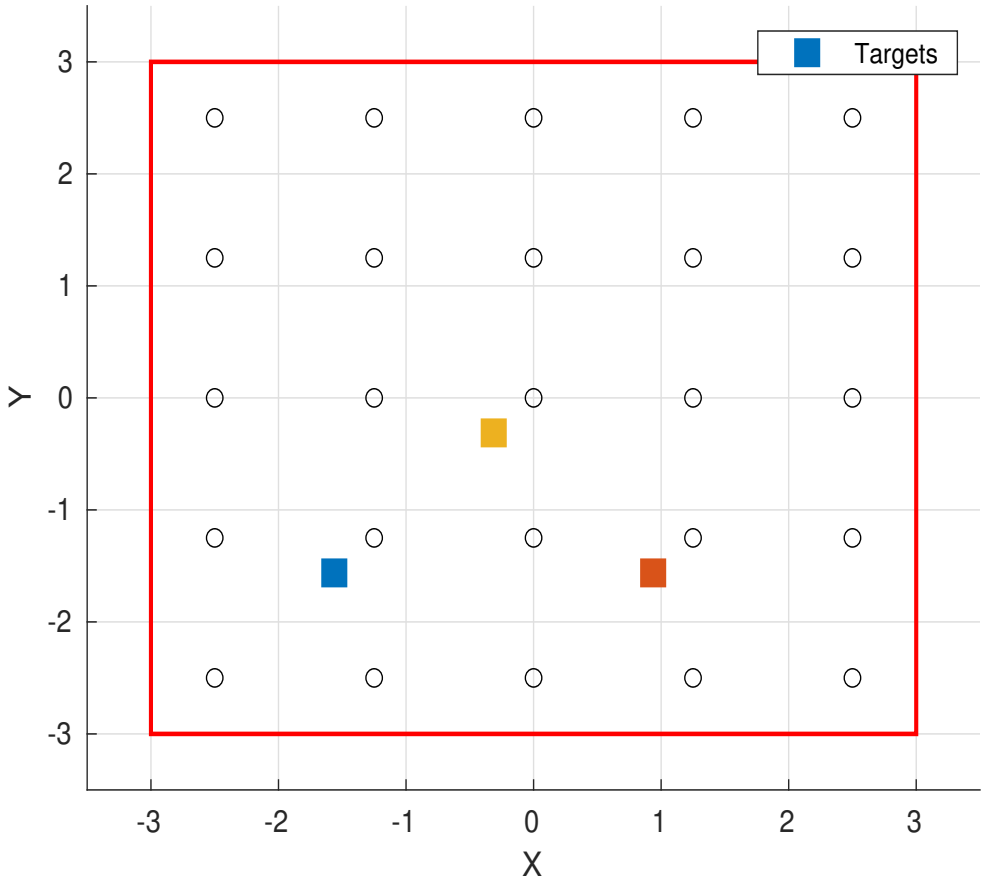


Figure 3.4: A 5m × 5m Square Environment, With Hollow Circles Denoting the Grid Nodes and Squares Denoting the Targets. The Red Border Denotes the Boundary of the Area That Is Explored by 3 Robots.

the corresponding peak of the graph. At this time, the next inter-arrival time $\tau_{j+1}^{a_m a_n}$ is initialized to zero. Figure 3.6 plots the time evolution of the cardinality of the reward Equation (3.6) earned by each robot, which is the estimated number of targets. The average inter-arrival time over this simulation run was calculated to be $\mathbb{E}[\tau_k^{(\cdot)}] \sim 68$ s, and the time required for the cardinality of all robots' rewards to equal the actual number of targets, $n = 3$, was $t_{reward} \sim 150$ s.

Thus, for a scenario with both a robot density (number of robots per m^2) and

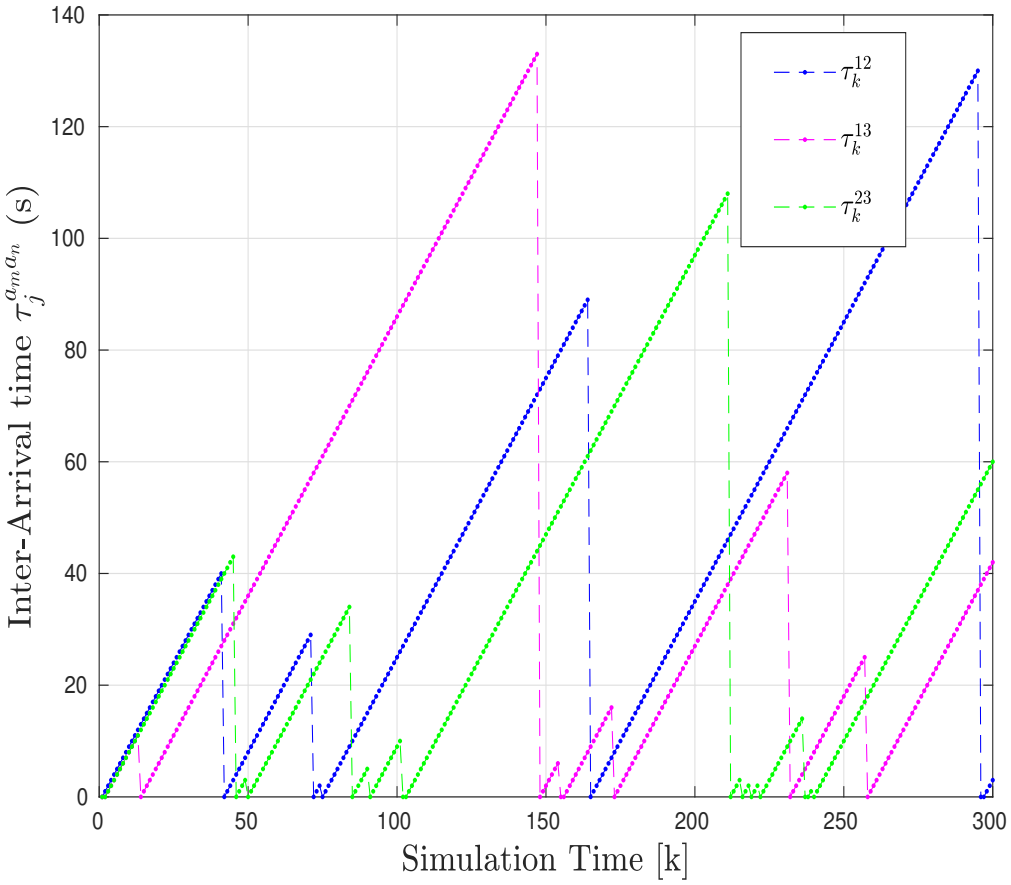


Figure 3.5: Inter-arrival Times During a Simulation of 3 Robots Exploring the Environment In Figure 3.4.

a target density (number of targets per m^2) of $3/25 = 0.12$ m^{-2}, there must be about $\frac{t_{reward}}{\mathbb{E}[\tau_j^{(\cdot)}]} \approx 2.2$ renewals, i.e. at least 2 renewals, for all robots to achieve the same reward cardinality (estimated number of targets). Figure 3.7, Figure 3.9, and Figure 3.11 show the true positions of the targets and their estimated positions by each robot at the end of the simulation time. Figure 3.8, Figure 3.10, and Figure 3.12 show the corresponding PHD intensity for each robot as a Gaussian mixture model with $n = 3$ components (the number of targets), computed from Equation (3.33). We

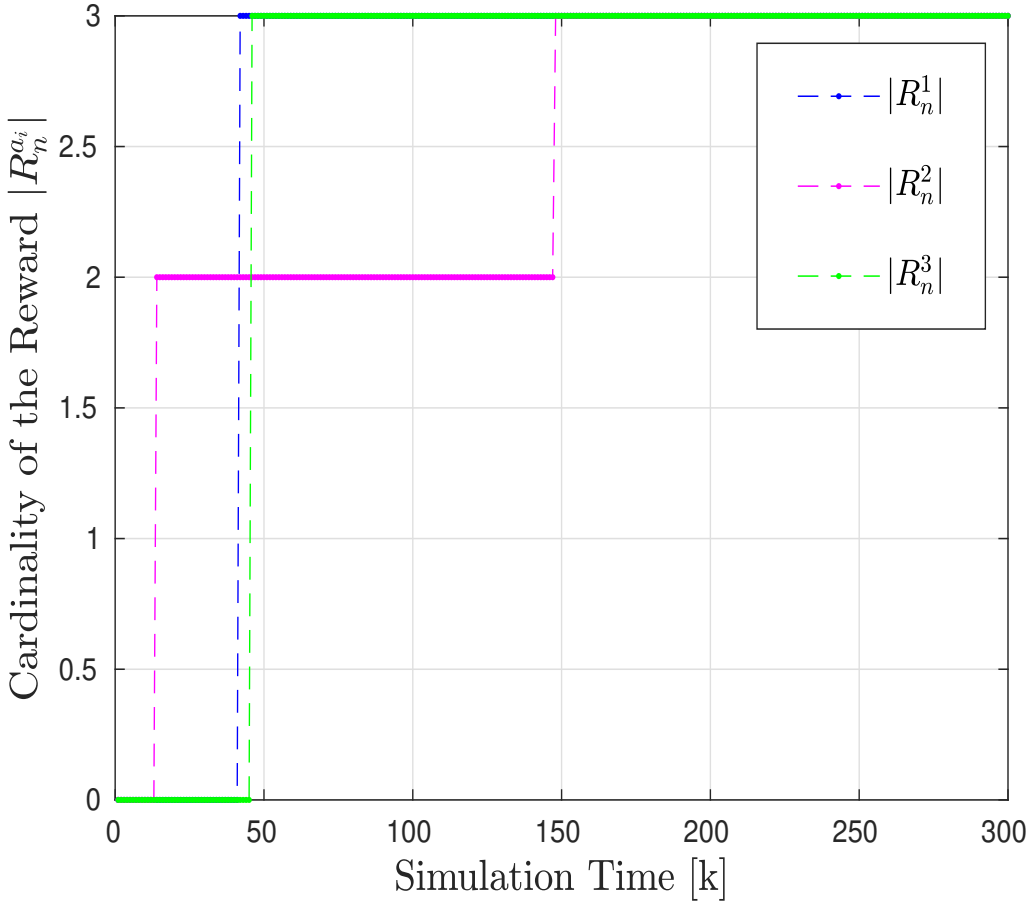


Figure 3.6: Cardinality of the Reward Accumulated by Each of the Robots at Each Time Step.

obtain the number of targets estimated by each robot a_i as

$$\hat{N}^{a_i} = \left\lceil \sum_{l=1}^{n} w_k^{(l),a_i} \right\rceil, \tag{3.41}$$

where the weights $w_k^{(\cdot),a_i}$ for robots $a_i = 1, 2, 3$ are given by the peak intensities in Figure 3.8, Figure 3.10, and Figure 3.12, respectively. The estimated positions of the targets are obtained from positions of these peak intensities.

We also evaluated our approach through Monte Carlo simulations of three scenarios, with 100 simulation runs for each scenario. In all scenarios, 20 robots explored

Scenarios	1	2	3
Mean inter-arrival time (s)	20	190	430
Mean reward (%)	10	33	65

Table 3.1: Mean Inter-arrival Time and Mean Reward Percentage Over 100 Simulation Runs Each for 3 Scenarios.

a grid according to the random walk model Equation (3.1) in order to track a set of stationary targets. The robots were initialized at random positions on the grid, and the positions of the targets were kept the same over all 100 runs for each scenario. In Scenario 1, simulated for 1000 s, the grid has dimensions 15m $\times$ 15m and there are 10 targets; in Scenario 2, simulated for 2000 s, the grid has dimensions 20m $\times$ 20m and there are 15 targets; and in Scenario 3, simulated for 3000 s, the grid has dimensions 30m $\times$ 30m and there are 20 targets. The mean inter-arrival time and mean reward percentage for each scenario, averaged over all 100 runs, are given in Table 3.1. The mean reward percentage is computed from the ratio of the mean number of targets detected by the robots until the mean inter-arrival time to the actual number of targets in the scenario. Table 3.1 shows that the mean inter-arrival time increases as the density of robots in the environment decreases, which is due to the lower rate of robots encounters with one another in larger environments, on average. The table also shows that as the density of targets in the environment decreases, the percentage of targets identified before the mean inter-arrival time increases, on average. This indicates that in the scenarios simulated, the longer inter-arrival times for larger environments tend to enable identification of a higher number of targets, despite the lower target density.

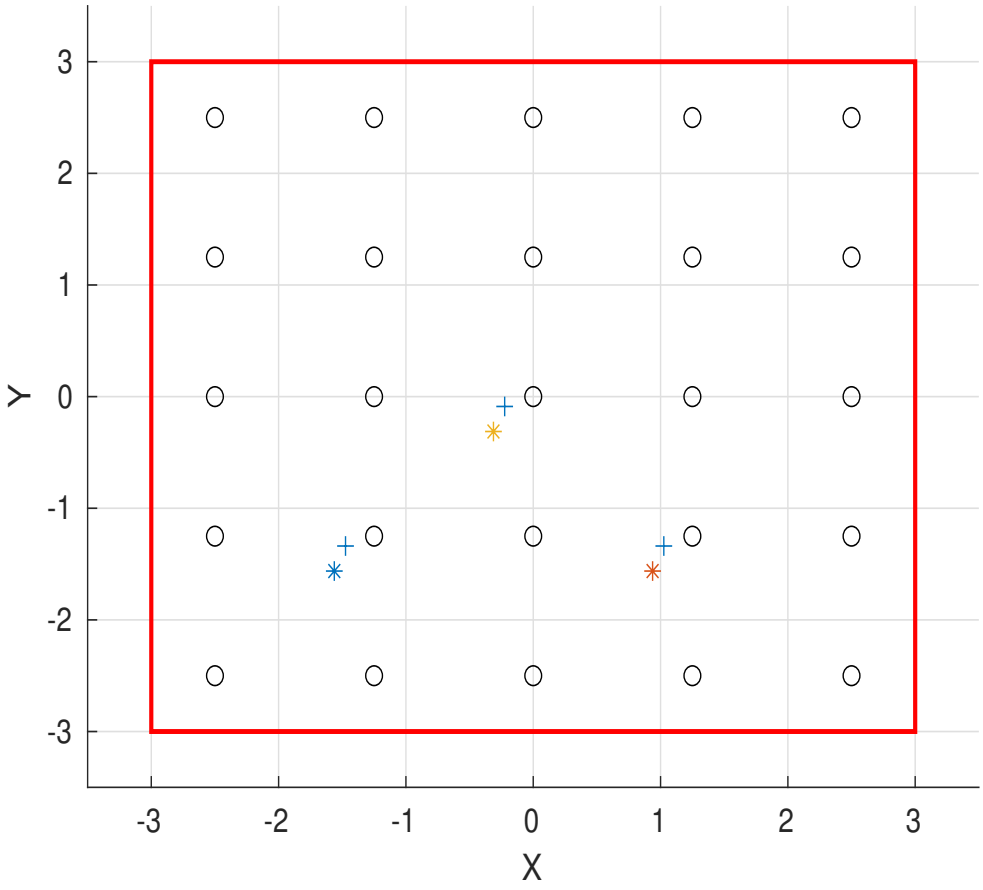


Figure 3.7: Multi-Target Tracking by Robot 1. Estimated (+) and True (∗) Target Positions.

3.7 Conclusion and Future Work

In this chapter, we demonstrated theoretically that a group of robots equipped with limited sensing and communication capabilities, moving according to a DTDS Markov chain model on a spatial grid, is able to detect and track the number and states of multiple stationary targets in the environment using the Gaussian Mixture formulation of the PHD filter from the RFS framework. We verified our results with numerical simulations in MATLAB.

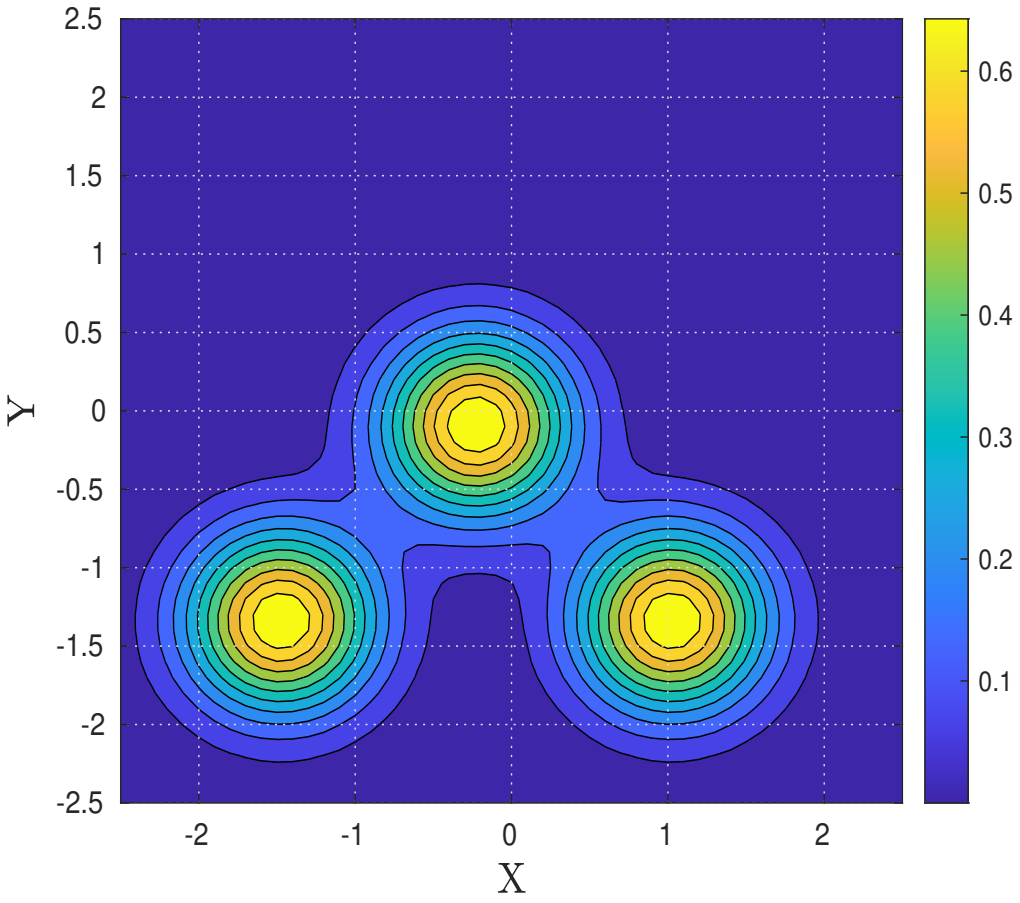


Figure 3.8: GM-PHD Intensities Computed From Equation (3.33) by Robot 1.

In the next chapter, we will show our strategy for reconstructing discrete distributions in a distributed fashion, using a distributed version of *Chernoff fusion* algorithm.

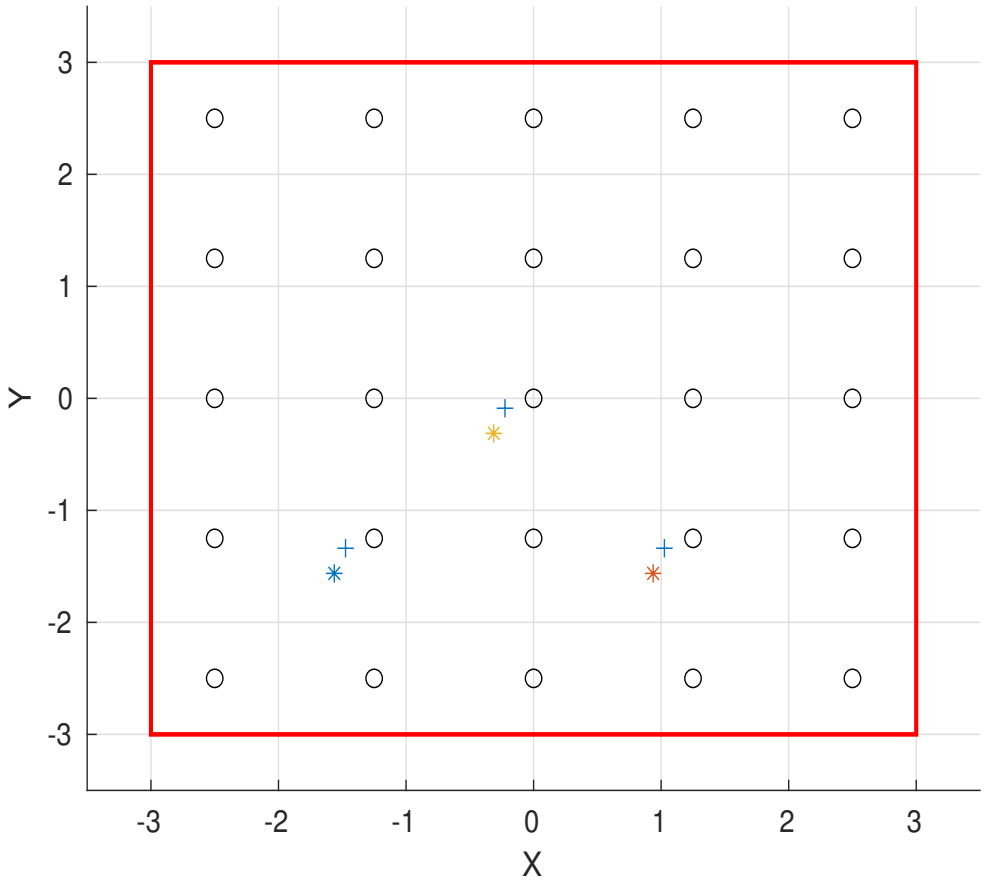


Figure 3.9: Multi-Target Tracking by Robot 2. Estimated (+) and True (∗) Target Positions.

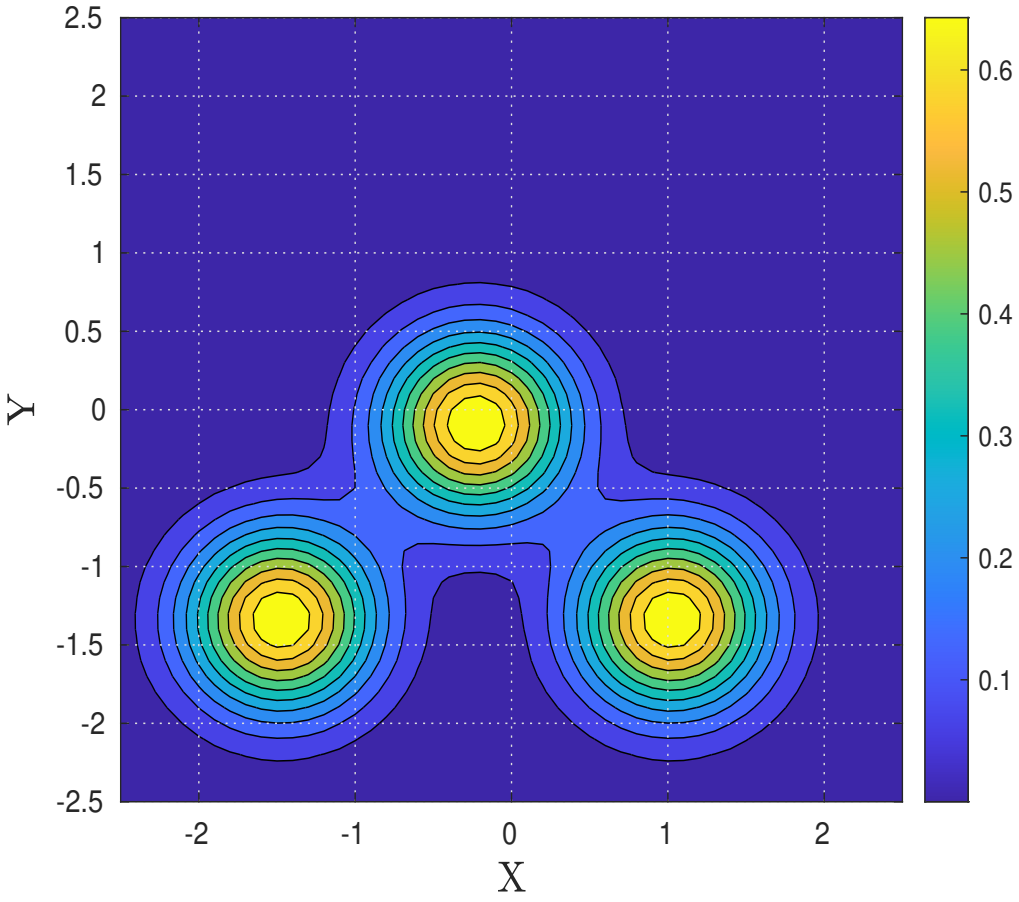


Figure 3.10: GM-PHD Intensities Computed From Equation (3.33) by Robot 2.

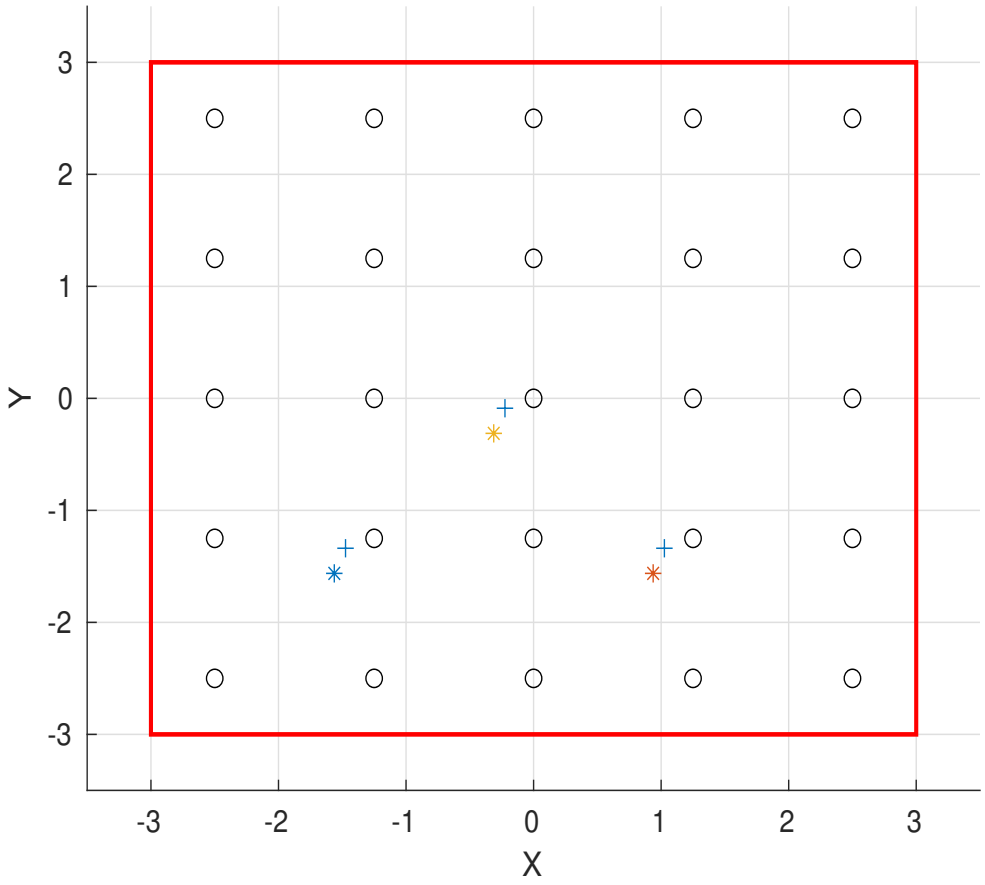


Figure 3.11: Multi-Target Tracking by Robot 3. Estimated (+) and True (∗) Target Positions.

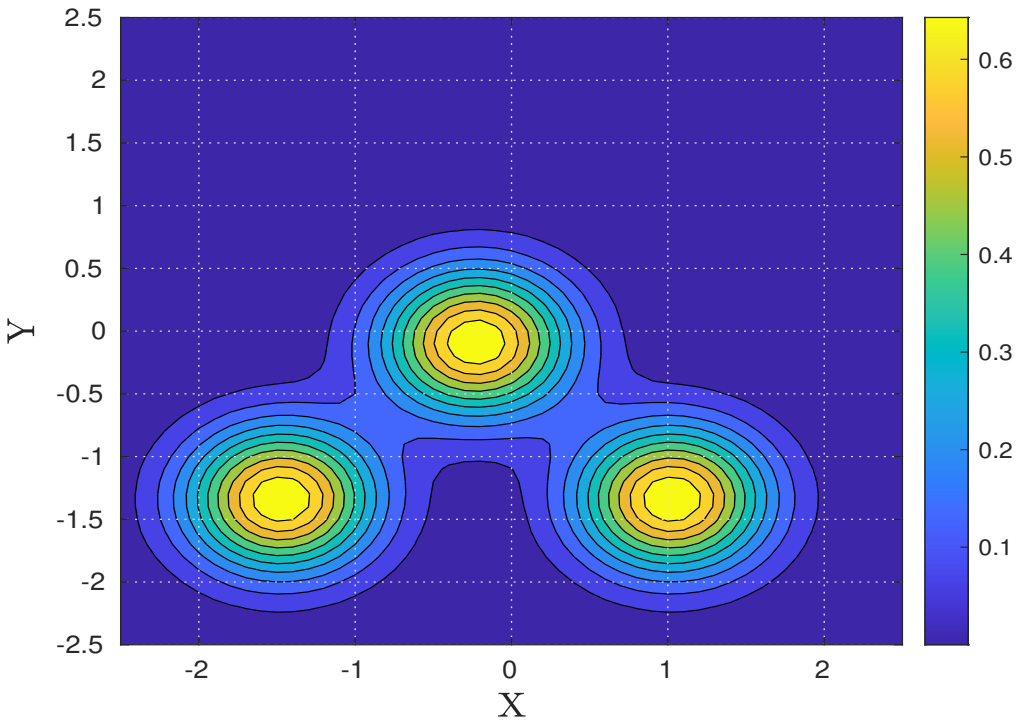


Figure 3.12: GM-PHD Intensities Computed From Equation (3.33) by Robot 3.

Chapter 4

DISTRIBUTED FEATURE RECONSTRUCTION USING CHERNOFF FUSION

This chapter presents research and results from Shirsat *et al.* (2022)[**A. Shirsat**, S. Mishra, W. Zhang, and S. Berman, "Probabilistic Consensus on Feature Distribution for Multi-Robot Systems With Markovian Exploration Dynamics,"In revision for IEEE Robotics and Automation Letters (RA-L), 2022]

4.1 Overview

In this chapter, we generalize the consensus objective of our probabilistic multi-robot search strategy to agreement on a discrete distribution of static features, modeled as an occupancy grid map, using results on *opinion pools* Stone (1961); Degroot (1974); Bailey *et al.* (2012). We consider a group of robots that move according to a DTDS Markov chain on a finite 2D spatial grid, as shown in Figure 4.1, and that can detect features using their on-board sensors. The proposed strategy is distributed and asynchronous, and it preserves the required communication bandwidth by relying only on local inter-robot communication.

The remainder of the chapter is organized as follows. We present our probabilistic exploration strategy and information fusion protocol in Section 4.2. We describe some relevant properties of DTDS Markov chains in Section 4.3, and we derive the main result that guarantees the convergence of each robot's feature distribution to the actual distribution in Section 4.4. We present the results of our numerical simulations in Section 4.5 and our SITL simulations in Section 4.6. Section 4.9 concludes the chapter.

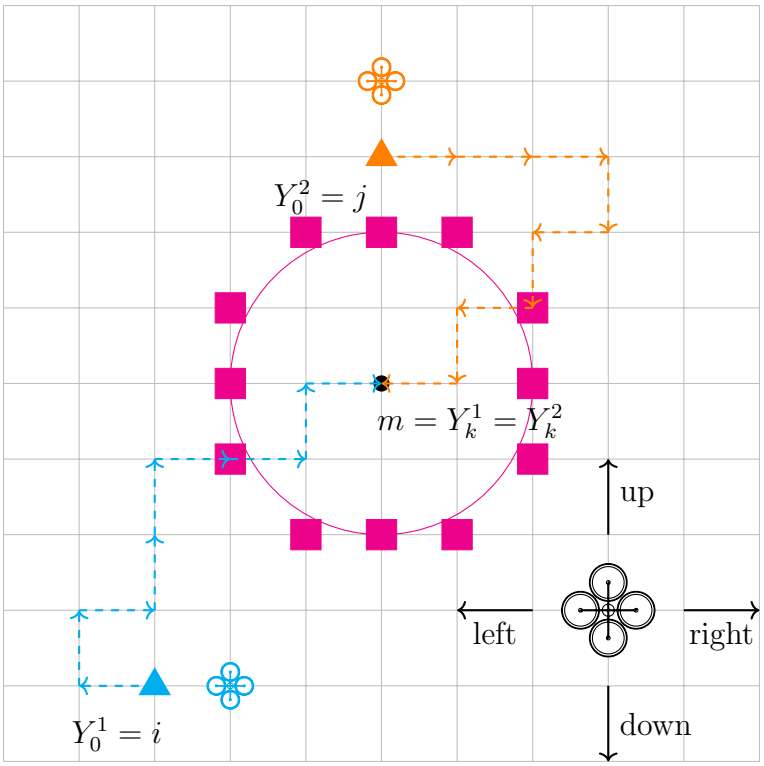


Figure 4.1: Illustration of Our Multi-Robot Exploration Strategy, Showing Sample Paths for Two Quadrotors (Orange and Blue) on a Square Grid. The Quadrotors Search the Environment for a Set of Static Features (the Magenta Squares Representing a Discretized Circle) As They Perform a Random Walk on the Grid.

4.2 Problem Formulation

Consider a bounded square environment $\mathcal{B} \subset \mathbb{R}^2$ with sides of length B. We discretize $\mathcal{B}$ into a square grid of nodes spaced at a distance δ apart. The set of nodes is denoted by $\mathcal{S} \subset \mathbb{Z}_+$, and we define $S = |\mathcal{S}|$. A set of N robots, $\mathcal{A} = \{1, 2, \ldots, N\}$, each modeled as a point mass, explore the environment by performing a random walk on the grid. We assume that there are no obstacles in the environment that impede the robots' motion. Let $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ be an undirected graph associated with this finite spatial grid, where $\mathcal{V}_s = \mathcal{S}$ is the set of nodes and $\mathcal{E}_s$ is the set of edges (i, j). The edges signify pairs of nodes $i, j \in \mathcal{V}_s$, called *neighboring nodes*, between which robots can travel. We assume that the robots can localize on $\mathcal{G}_s$.

Let $Y_k^a \in \mathcal{S}$ be a random variable that defines the node that robot $a \in \mathcal{A}$ occupies at discrete time k. Robot a moves from its current node i to a neighboring node j at the next time step with a transition probability $p_{ij} \in [0, 1]$. We define $\mathbf{P} \in \mathbb{R}^{S \times S}$ as the *state transition matrix* consisting of elements p_{ij} at row i and column j. Let $\pi_k \in \mathbb{R}^{1 \times S}$ denote the probability mass function (PMF) of Y_k^a for each robot a, or alternatively, the distribution of the robot population over the grid at time k. This distribution evolves over time according to a DTDS Markov chain model of order one:

$$\pi_{k+1} = \pi_k \mathbf{P}. \tag{4.1}$$

We assume that each robot can exchange information with other robots that are within a communication radius $r_{comm} \leq 0.5\delta$. Let $\mathcal{G}_c[k] = (\mathcal{V}_c, \mathcal{E}_c[k])$ be an undirected graph in which $\mathcal{V}_c = \mathcal{A}$, the set of robots, and $\mathcal{E}_c[k]$ is the set of all pairs of robots $(a, b) \in \mathcal{A} \times \mathcal{A}$ that can communicate with each other at time k. Let $\mathbf{M}[k] \in \mathbb{R}^{N \times N}$ be the *adjacency matrix* with elements $m_{ab}[k] = 1$ if $(a, b) \in \mathcal{E}_c[k]$ and $m_{ab}[k] = 0$ otherwise. For each robot $a \in \mathcal{A}$, we define the set of *neighbors* of robot a at time k as $\mathcal{N}_k^a \triangleq \{b \in \mathcal{A} : (a, b) \in \mathcal{E}_c[k]\}$.

A set of discrete features is distributed over the grid at nodes in the set $\mathcal{B}^r \subseteq \mathcal{S}$. The robots know a priori that these features are present in the environment, but do not know their distribution. We assume that when a robot is located at a node in $\mathcal{B}^r$, it can detect the presence of a feature at that node using its on-board sensors. Each node in the grid is associated with a binary occupancy value, defined as $\bar{l} \in (0.5, 1)$ if the robot detects a feature at that node and $1 - \bar{l}$ if it does not.

We define the *occupancy vector* for robot a at time k as $\bar{\theta}_k^a = [\theta_k^a(1) \ \ldots \ \theta_k^a(S)] \in \mathbb{R}^{1 \times S}$, where

$$\theta_k^a(s) = \begin{cases} \bar{l}, & s \in \mathcal{B}^r \ (occupied) \\ 1 - \bar{l}, & o.w \ (unoccupied) \end{cases} \tag{4.2}$$

The occupancy vector for each robot indicates its estimate of the nodes that are occupied by features. The feature PMF, or *occupancy distribution*, estimated by robot a at time k from its own sensor measurements,

$$f_k^a(s) = \frac{\theta_k^a(s)}{\sum_{i \in \mathcal{S}} \theta_k^a(i)} \tag{4.3}$$

We denote $\theta^{ref}(s)$ as the *reference occupancy vector* that is being estimated by all the robots, defined as follows:

$$\theta^{ref}(s) = \begin{cases} \bar{l}, & s \in \mathcal{B}^r \\ 1 - \bar{l}, & o.w \end{cases} \tag{4.4}$$

Here, $f_k^a(s)$ also represents the *opinion* (Degroot, 1974; Stone, 1961) of robot a at time k for the occupancy distribution. Since the robots do not know the occupancy distribution a priori, we specify that they all initially consider the grid to be unoccupied, i.e.,

$$f_0^a(s) = \frac{1 - \bar{l}}{S(1 - \bar{l})}, \quad a \in \mathcal{A} \tag{4.5}$$

This is defined as the *nominal distribution* for all the robots, denoted by $f^{nom}(s) = f_0^{(\cdot)}(s)$. We also denote a vector $\theta^{nom}(s)$ as the *nominal occupancy vector* for all robots, which represents all nodes as unoccupied. We define $f^{ref}(s)$ as the *reference PMF*, the actual feature distribution, corresponding to $\theta^{ref}(s)$.

We define a fusion weight $\omega_k^{(a,b)}$ as the following *Metropolis weight* (Calafiore and Abrate, 2009; Xiao *et al.*, 2005):

$$\omega_k^{(a,b)} = \begin{cases} \frac{1}{1+|\mathcal{N}_k^b|}, & b \in \mathcal{N}_k^a \setminus \{a\} \\ 1 - \sum_{b \in \mathcal{N}_k^a \setminus \{a\}} \omega_k^{(a,b)}, & a = b,\ a \in \mathcal{A} \\ 0, & o.w \end{cases} \tag{4.6}$$

Note that $\omega_k^{(a,b)} \geq 0$ and $\sum_{b \in \mathcal{N}_k^a} \omega_k^{(a,b)} = 1$. We then define the *consensus* or *opinion weighting matrix* $\Omega_k \in \mathbb{R}^{N \times N}$ at time k, which consists of elements $\omega_k^{(a,b)}$ at row a and column b.

Given the robot exploration dynamics in Equation (4.1), each robot a updates its opinion $f_k^a(s)$, computed from Equation (4.3), to the following PMF $f_{k+1}^{cher}(s)$ at the next time step, which it computes from the opinions of other robots within its communication range according to the *Chernoff fusion rule* (Farrell and Ganesh, 2009):

$$f_{k+1}^{cher}(s) = \frac{\prod_{b \in \mathcal{N}_k^a \cup \{a\}} [f_k^b(s)]^{\omega_k^{(a,b)}}}{\sum_{s \in \mathcal{S}} \prod_{b \in \mathcal{N}_k^a \cup \{a\}} [f_k^b(s)]^{\omega_k^{(a,b)}}}, \quad a \in \mathcal{A} \tag{4.7}$$

Applying Theorem 1 from (Battistelli and Chisci, 2014), we can say that $f_{k+1}^{cher}(s)$ is the local neighbor fused feature PMF at time $k+1$ of the all robots $a \in \mathcal{A}$ that are in $\mathcal{N}_k^a$. When only two robots $a, b \in \mathcal{A}$ are within communication range, this update rule becomes:

$$f_{k+1}^{cher}(s) = \frac{[f_k^a(s)]^{\omega} [f_k^b(s)]^{1-\omega}}{\sum_{s \in \mathcal{S}} [f_k^a(s)]^{\omega} [f_k^b(s)]^{1-\omega}}, \tag{4.8}$$

where the Metropolis weight $\omega \equiv \omega_k^{(a,b)}$ is defined in Equation (4.6).

Then each robot a compares $f_{k+1}^{cher}(s)$ with $f^{nom}(s)$ to generate a new fused occupancy vector as follows:

$$\theta_{k+1}^{cher}(s) = \begin{cases} \bar{l} & f_{k+1}^{cher}(s) > f^{nom}(s), \ s \in \mathcal{S} \\ 1 - \bar{l} & o.w \end{cases} \tag{4.9}$$

Then, each robot a generates a new occupancy vector $\theta_{k+1}^a(s)$ by comparing both its occupancy vector at the previous time step, $\theta_k^a(s)$, and the fused occupancy vector $\theta_{k+1}^{cher}(s)$, to $\theta^{nom}(s)$:

$$\theta^a_{k+1}(s) = \begin{cases} \bar{l} & \theta^a_k(s) > \theta^{nom}(s) \text{ or} \\ & \theta^{cher}_{k+1}(s) > \theta^{nom}(s), s \in \mathcal{S} \\ \theta^a_k(s) & o.w \end{cases} \tag{4.10}$$

To quantify the convergence of each robot's feature distribution to the reference distribution, we use the *Hellinger distance*, which measures the similarity between two PMFs. The Hellinger distance between the feature PMF $f^a_k(s)$ of a robot $a \in \mathcal{A}$ at time k and the reference PMF $f^{ref}(s)$ is given by

$$\mathbf{D}_H(f^a_k(s), f^{ref}(s)) = \sqrt{1 - \rho(f^a_k(s), f^{ref}(s))}\ , \tag{4.11}$$

where $\rho(f^a_k(s), f^{ref}(s))$ is the *Bhattacharya coefficient*, defined as:

$$\rho(f^a_k(s), f^{ref}(s)) = \sum_{s \in \mathcal{S}} \sqrt{f^a_k(s) \cdot f^{ref}(s)}\ . \tag{4.12}$$

This distance takes values in $[0, 1]$. We define the vector $\mathbf{D}_H \in \mathbb{R}^{N \times 1}_{\geq 0}$ with each entry $a \in \mathcal{A}$ given by $\mathbf{D}_H(f^a_k(s), f^{ref}(s))$.

The pseudo code in Algorithm 6 implements this fusion strategy for two robots a and b. In this algorithm, the normalizing constant c is the denominator of Equation (4.8). Figure 4.1 illustrates the proposed strategy for a scenario with two quadrotors. The quadrotors start at the spatial grid nodes indexed by i and j and move on the grid according to the DTDS Markov chain dynamics in Equation (4.1). The figure shows sample paths of the quadrotors. The orange quadrotor detects the feature, indicated by a magenta square, when it moves to a node in the set $\mathcal{B}^r$ (at these nodes, the feature is within the quadrotor's sensing range). The quadrotors meet at grid node m after $k = 9$ time steps and fuse occupancy distributions according to Equation (4.8). They stop exploring when the Hellinger distance between their fused opinions is less than a small positive number ϵ; otherwise, they continue to random-walk on the grid until a specified maximum time T.

Algorithm 6: Distributed Chernoff Fusion Protocol for Robots $a, B \in \mathcal{A}$ Computed by Robot a at Time k

input : $f_k^a,\ f_k^b,\ f^{ref},\ Y_k^a,\ Y_k^b,\ k,\ T,\ \epsilon$

output: f_{k+1}^a

if $\mathbf{D}_H(f_k^a, f^{ref}) > \epsilon \ \ and \ \ 1 < k \leq T$ **then**

  if $Y_k^a = Y_k^b$ **then**

    $\omega = 0.5$

  else

    $\omega = 1.0$

  end

  $c = \sum_{s \in \mathcal{S}} [f_k^a(s)]^{\omega} \cdot [f_k^b(s)]^{1-\omega}$

  $logf_{cher} = \omega \log(f_k^a) + (1-\omega)\log(f_k^b) - \log(c)$

  $f_{k+1}^a = f_{k+1}^{cher} = \exp(logf_{cher})$

end

4.3 Properties of DTDS Markov Chains

The Markov chain in Equation (4.1) is characterized in terms of the time-invariant state transition matrix $\mathbf{P}$, which is defined by the state space of the spatial grid representing the discretized environment. Hence, the Markov chain is *time-homogeneous*, which implies that $Pr(Y_{k+1}^a = j \mid Y_k^a = i)$ is the same for all robots at all times k. The entries of $\mathbf{P}$, which are the state transition probabilities, can therefore be defined as

$$p_{ij} = Pr(Y_{k+1}^a = j \mid Y_k^a = i),\ \forall i, j \in \mathcal{S},\ k \in \mathbb{Z}_+,\ \forall a \in \mathcal{A}. \tag{4.13}$$

Since each robot chooses its next node from a uniform distribution, these entries

can be computed as

$$p_{ij} = \begin{cases} \frac{1}{d_i+1}, & (i,j) \in \mathcal{E}_s, \\ 0, & o.w, \end{cases} \tag{4.14}$$

where d_i is the degree of the node $i \in \mathcal{S}$, defined as $d_i = 2$ if i is a corner node of the spatial grid, $d_i = 3$ if it is on an edge between two corner nodes, and $d_i = 4$ otherwise. Since each entry $p_{ij} \geq 0$, we use the notation $\mathbf{P} \geq 0$. We see that $\mathbf{P}^m \geq 0$ for $m \geq 1$, and therefore $\mathbf{P}$ is a *non-negative matrix*. From Theorem 5 in (Grimmett and Stirzaker, 2001), we can conclude that $\mathbf{P}$ is a stochastic matrix. If π is a stationary distribution of Markov chain Equation (4.1), then $\forall k \in \mathbb{Z}_+$,

$$\pi \mathbf{P}^k = \pi. \tag{4.15}$$

From the construction of the Markov chain, each robot has a positive probability of moving from any node $i \in \mathcal{S}$ to any other node $j \in \mathcal{S}$ of the spatial grid in a finite number of time steps. As a result, the Markov chain is *irreducible*, and therefore $\mathbf{P}$ is an irreducible matrix. By Lemma 8.4.4 (Perron-Frobenius) in (Horn and Johnson, 1990), there exists a real unique positive left eigenvector of $\mathbf{P}$. Moreover, since $\mathbf{P}$ is a stochastic matrix, its spectral radius $\rho(\mathbf{P})$ is 1. Therefore, this left eigenvector is the stationary distribution of the corresponding Markov chain.

Since we have shown that the Markov chain is irreducible and has a stationary distribution π, which satisfies $\pi\mathbf{P} = \pi$, we can conclude from Theorem 21.12 in (Levin and Peres, 2017) that the Markov chain is *positive recurrent*. Thus, all states in the Markov chain are positive recurrent, which implies that each robot will keep visiting every state on the finite spatial grid infinitely often.

By Theorem 1 of (Battistelli and Chisci, 2014), Equation (4.8) achieves opinion consensus over a graph with a fixed and strongly connected topology. We extend this result to graphs with topologies that evolve according to the switching dynamics on the composite Markov chain described in this section. We demonstrate that under our opinion fusion scheme, all the robots will reach a consensus on the feature distribution.

The dynamics of all robots' movements on the spatial grid can be modeled by a composite Markov chain with states $\mathbf{Y}_k = (Y_k^1, Y_k^2, \ldots, Y_k^N) \in \mathcal{M}$, where $\mathcal{M} = \mathcal{S}^{\mathcal{A}}$. Note that $S = |\mathcal{S}|$ and $|\mathcal{M}| = S^N$. We define an undirected graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ that is associated with the composite Markov chain. The vertex set $\hat{\mathcal{V}}$ is the set of all possible realizations $\hat{\imath} \in \mathcal{M}$ of $\mathbf{Y}_k$. The notation $\hat{\imath}(a)$ represents the a^{th} entry of $\hat{\imath}$, which is the node $i \in \mathcal{S}$ occupied by robot a. We define the edge set $\hat{\mathcal{E}}$ as follows: $(\hat{\imath}, \hat{\jmath}) \in \hat{\mathcal{E}}$ if and only if $(\hat{\imath}(a), \hat{\jmath}(a)) \in \mathcal{E}_s$ for all robots $a \in \mathcal{N}$. Let $\mathbf{Q} \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{M}|}$ be the state transition matrix associated with the composite Markov chain. The elements of $\mathbf{Q}$, denoted by $q_{\hat{\imath}\hat{\jmath}}$, are computed from the transition probabilities defined in Equation (4.14) as follows:

$$q_{\hat{\imath}\hat{\jmath}} = \prod_{a=1}^{N} p_{\hat{\imath}(a)\hat{\jmath}(a)}, \qquad \forall \hat{\imath}, \hat{\jmath} \in \mathcal{M}. \tag{4.16}$$

Here, $q_{\hat{\imath}\hat{\jmath}}$ is the probability that in the next time step, each robot a will move from node $\hat{\imath}(a)$ to node $\hat{\jmath}(a)$.

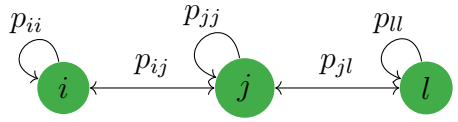


Figure 4.2: A Graph $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ Defined on the Set of Spatial Nodes $\mathcal{V}_s = \{i, j, l\}$. The Arrows Signify Undirected Edges Between Pairs of Distinct Nodes and Self-Edges. The Edge Set of the Graph Is $\mathcal{E}_s = \{(i,i), (j,j), (l,l), (i,j), (j,l)\}$.

For example, consider a set of two robots, $\mathcal{N} = \{1, 2\}$, that move on the graph $\mathcal{G}_s$ in Figure 4.2. In the next time step, the robots can stay at their current node or travel

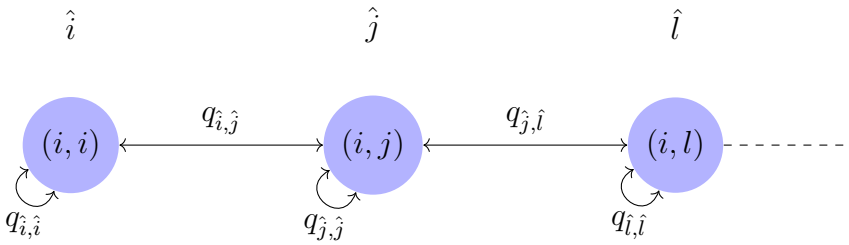


Figure 4.3: A Subset of the Composite Graph $\hat{\mathcal{G}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ For 2 Robots That Move on the Graph $\mathcal{G}_s$ Shown In Figure 4.2.

between nodes i and j and between nodes j and l, but they cannot travel between nodes i and l. Figure 4.3 shows a subset of the resulting composite graph $\hat{\mathcal{G}}$,whose entire set of nodes is $\hat{\mathcal{V}} = \{(i,i),(i,j),(i,l),(j,i),(j,j),(j,l),(l,i),(l,j),(l,l)\}$. Each node in $\hat{\mathcal{V}}$ is labeled by an index $\hat{\imath}$, e.g., $\hat{\imath} = (i,j)$, with $\hat{\imath}(1) = i$ and $\hat{\imath}(2) = j$. Given the connectivity of the spatial grid defined by $\mathcal{E}_s$, we can for example identify $((i,j),(i,l))$ as an edge in $\hat{\mathcal{E}}$, but not $((i,j),(l,l))$. Since $N = 2$ and $S = 3$, we have that $|\mathcal{M}| = 3^2 = 9$. For each $\hat{\imath}, \hat{\jmath} \in \hat{\mathcal{V}}$, we can compute the transition probabilities in $\mathbf{Q} \in \mathbb{R}^{9\times 9}$ from Equation (4.16) as follows:

$$q_{\hat{\imath}\hat{\jmath}} = Pr\left(\mathbf{Y}_{k+1} = \hat{\jmath} \mid \mathbf{Y}_k = \hat{\imath}\right) = p_{\hat{\imath}(1)\hat{\jmath}(1)} p_{\hat{\imath}(2)\hat{\jmath}(2)}, \quad k \in \mathbb{Z}_+. \tag{4.17}$$

We now prove that all robots will reach consensus on the feature distribution and also that this distribution will converge to the reference distribution.

Theorem 4.4.1. *Consider a group of N robots, moving on a finite spatial grid with DTDS Markov chain dynamics Equation* (4.1)*, that update their opinions $f_k^a(s)$ for the feature distribution on the grid according to Equations* (4.3)*,* (4.7)*,* (4.9) *and* (4.10)*. Then for each robot $a \in \mathcal{A}$, $f_k^a(s)$ will converge to $f^{ref}(s)$ as $k \to \infty$ almost surely.*

Proof. Suppose at initial time k_0, the locations of the robots on the spatial grid are given by the node $\hat{\imath} \in \hat{\mathcal{V}}$. Consider another set of robot locations at a future time

$k_0 + k$, given by the node $\hat{j} \in \hat{\mathcal{V}}$. The transition of the robots from configuration $\hat{\imath}$ to configuration $\hat{\jmath}$ in k time steps corresponds to a random walk of length k on the composite Markov chain $\mathbf{Y}_k$ from node $\hat{\imath}$ to node $\hat{\jmath}$. It also corresponds to a random walk by each robot a on the spatial grid from node $\hat{\imath}(a)$ to node $\hat{\jmath}(a)$ in k time steps. By construction, the graph $\mathcal{G}_s$ is strongly connected and each of its nodes has a self-edge. Thus, there exists a discrete time $n > 0$ such that, for each robot a, there exists a random walk on the spatial grid from node $\hat{\imath}(a)$ to node $\hat{\jmath}(a)$ in n time steps. Consequently, there always exists a random walk of length n on the composite Markov chain $\mathbf{Y}_k$ from node $\hat{\imath}$ to node $\hat{\jmath}$, and therefore $\mathbf{Y}_k$ is an irreducible Markov chain. All states of an irreducible Markov chain belong to a single communication class. In this case, all states are *positive recurrent*; as a result, each state of $\mathbf{Y}_k$ is visited infinitely often by the group of robots. Moreover, because the composite Markov chain is irreducible, we can conclude that $\cup_{k\in\mathbb{Z}_+}\mathcal{G}_c[k] = \mathcal{G}_0$, where $\mathcal{G}_0$ is the complete graph on the set of robots. Therefore $\mathcal{G}_0$ is strongly connected. Hence, each robot will meet every other robot at some node $s \in \mathcal{S}$ infinitely often. Since $\mathbf{Y}_k$ is irreducible and, from Equation (4.10), we have that $\theta_k^a(s) \leq \theta_{k+1}^a(s) \leq \theta^{ref}(s),\ \forall a \in \mathcal{A},\ \forall s \in \mathcal{S}$, it follows from Equations (4.9) and (4.10) that $\theta_k^a(s) \to \theta^{ref}(s)$ as $k \to \infty$. Consequently, $f_k^a(s) \to f^{ref}(s)$ as $k \to \infty$ almost surely. □

4.5 Numerical Simulation Results

In the numerical simulations, we consider a set of robots $\mathcal{A} = \{1, 2, 3, 4\}$ moving on a 5.6 m × 5.6 m domain that is discretized into a square grid with $c = 8$ nodes on each side, with a distance $\delta = 0.7$ m between adjacent nodes. The robots switch from one node to another at each time step according to the Markov chain dynamics in Equation (4.1). The state transition probabilities $p_{ij},\ i, j \in \mathcal{S} = \{1, 2, \ldots, c^2\}$, that are associated with the spatial graph $\mathcal{G}_s$ are computed from Equation (4.14).

We set the value of $\bar{l} = 0.8$ in Equation (4.2) in both the numerical and Software-In-The-Loop simulations. We distribute the features on the set of nodes $\mathcal{B}^r = \{19, 20, 21, 26, 30, 34, 38, 42, 46, 51, 52, 53\}$, which represents a discrete approximation of a circular distribution on the grid. The set of neighbors $\mathcal{N}_k^a$ of robot a at time k consists of all robots that are located at the same node as robot a at that time.

All robots are initialized at uniformly random nodes in $\mathcal{S}$. Prior to exploration, the robots assume that all the grid nodes are unoccupied by features, and hence the vector of Hellinger distances is initially $\mathbf{D}_H = \mathbf{0} \in \mathbb{R}^{N\times 1}$. During their exploration of the grid, when robots encounter each other at the same node, they exchange their current feature PMFs and fuse them according to Equation (4.7).

Figure 4.4 and Figure 4.5 show the feature PMFs computed by each robot at $k = 240$ s and $k = 500$ s, respectively. We observe in Figure 4.4 that by $k = 240$ s, all robots have partially reconstructed the feature PMF, with robot 4 having the closest reconstruction as measured by $\mathbf{D}_H(f_{240}^4(s), f^{ref}(s)) \sim 0.07$ units in Figure 4.6.

In Figure 4.5, we see that all robots have successfully reconstructed the feature PMF by $k = 500$ s. The robots' consensus on the reference PMF is also apparent from Figure 4.6, which verifies that all Hellinger distances are zero at that time. Several distances $\mathbf{D}_H(f_k^a(s), f^{ref}(s))$ increase at times k between 400 s and 500 s, due to the numerical inaccuracies in the fusion. We also ran Monte Carlo numerical simulations with different numbers of robots, $N = \{4, 8, 12, 16\}$, to investigate the effect of N and the effect of consensus on the performance of the strategy in terms of the time for all robots' feature PMFs to converge to the reference PMF.

Figure 4.7 plots the resulting time, averaged over 100 simulations (error bars show standard deviations), until the feature PMFs of all N robots converge to the reference PMF in the consensus and no-consensus cases. The figure shows that for $N = 4$ robots, the mean time until convergence for both the consensus and no-consensus

strategies are similar, with a significant overlap in their standard deviations. This indicates that for small numbers of robots, both strategies perform similarly. However, as N increases, there is a widening gap between the mean times until convergence of the consensus and no-consensus strategies, with the times for the strategy with consensus being consistently lower. For $N = 16$ robots, the strategy with consensus is faster than the one without consensus by a factor of $\sim$2 (830 s/438 s).

4.6 Software-in-the-Loop Results

We also validate our approach in Software-In-The-Loop (SITL) simulations using Gazebo and ROS Melodic. We simulate the same scenario as in the numerical simulations, with the set of robots $\mathcal{A}$ consisting of two quadrotors, *Robot 1* and *Robot 2*. Figure 4.8 illustrates the simulation setup. The quadrotors are simulated using the Rotors (Furrer *et al.*, 2016) package in Gazebo, and the PX4 flight control stack (Meier *et al.*, 2015) is implemented in SITL to execute all low-level control tasks for each quadrotor. The discrete feature distribution that the quadrotors must reconstruct is represented by the ArUco markers on the ground plane, located at positions along the red dotted circle.

The quadrotors fly at different altitudes (1 m and 2 m) to avoid collisions. This eliminates the need for obstacle detection and avoidance strategies, which are beyond the scope of this work. To detect the ArUco markers, each quadrotor is equipped with a simulated VGA resolution RGB camera that takes images at 30 fps and is oriented to face the ground plane. The image window size is heuristically adjusted to account for the difference in perspective resulting from the robots' difference in altitude. The quadrotors are assigned static IP addresses and exchange their feature distributions over wireless communication when they meet at the x, y position of the same node (at different altitudes).

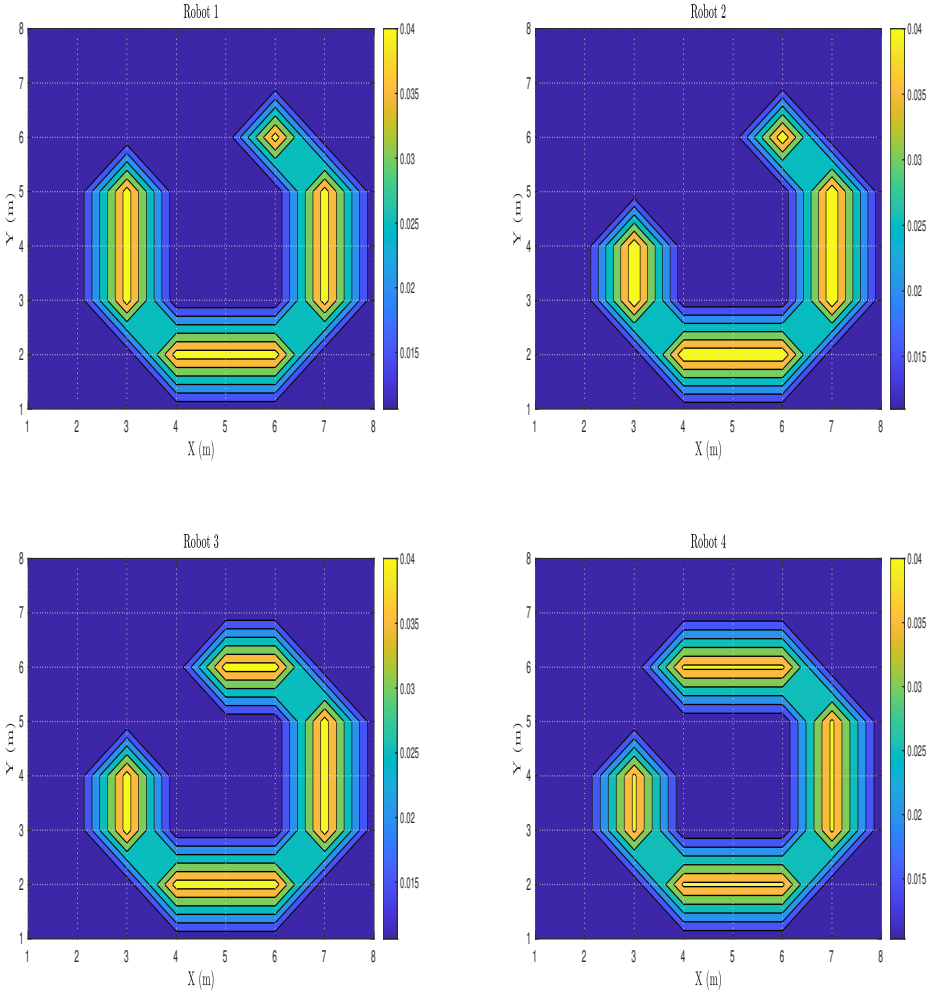


Figure 4.4: Feature PMF $f_k^a(s)$ of 4 Robots at Time $k = 240$ S in the Numerical Simulation.

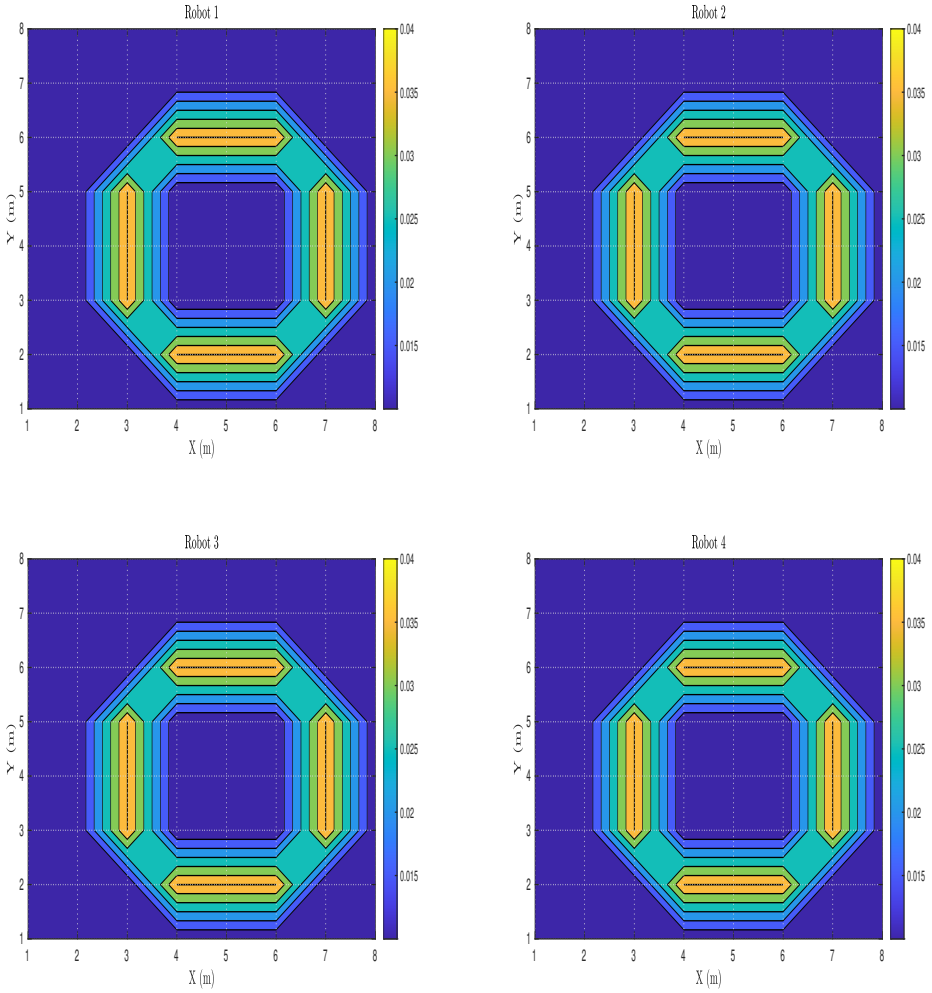


Figure 4.5: Feature PMF $f_k^a(s)$ of 4 Robots at Time $k = 500$ S in the Numerical Simulation.

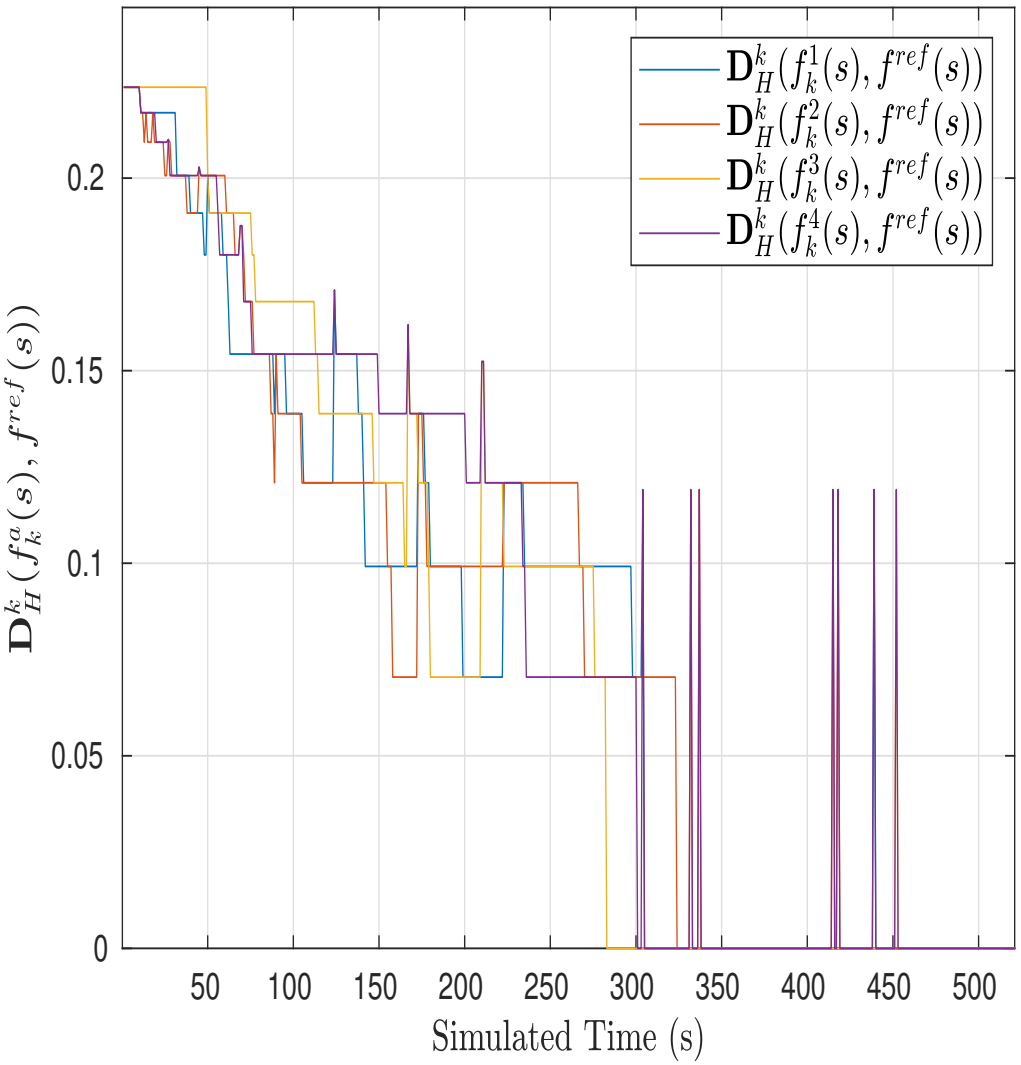


Figure 4.6: Time Evolution of $\mathbf{D}_H(f_k^a(s), f^{ref}(s))$ For Each Robot $a \in \{1, 2, 3, 4\}$ In the Numerical Simulation.

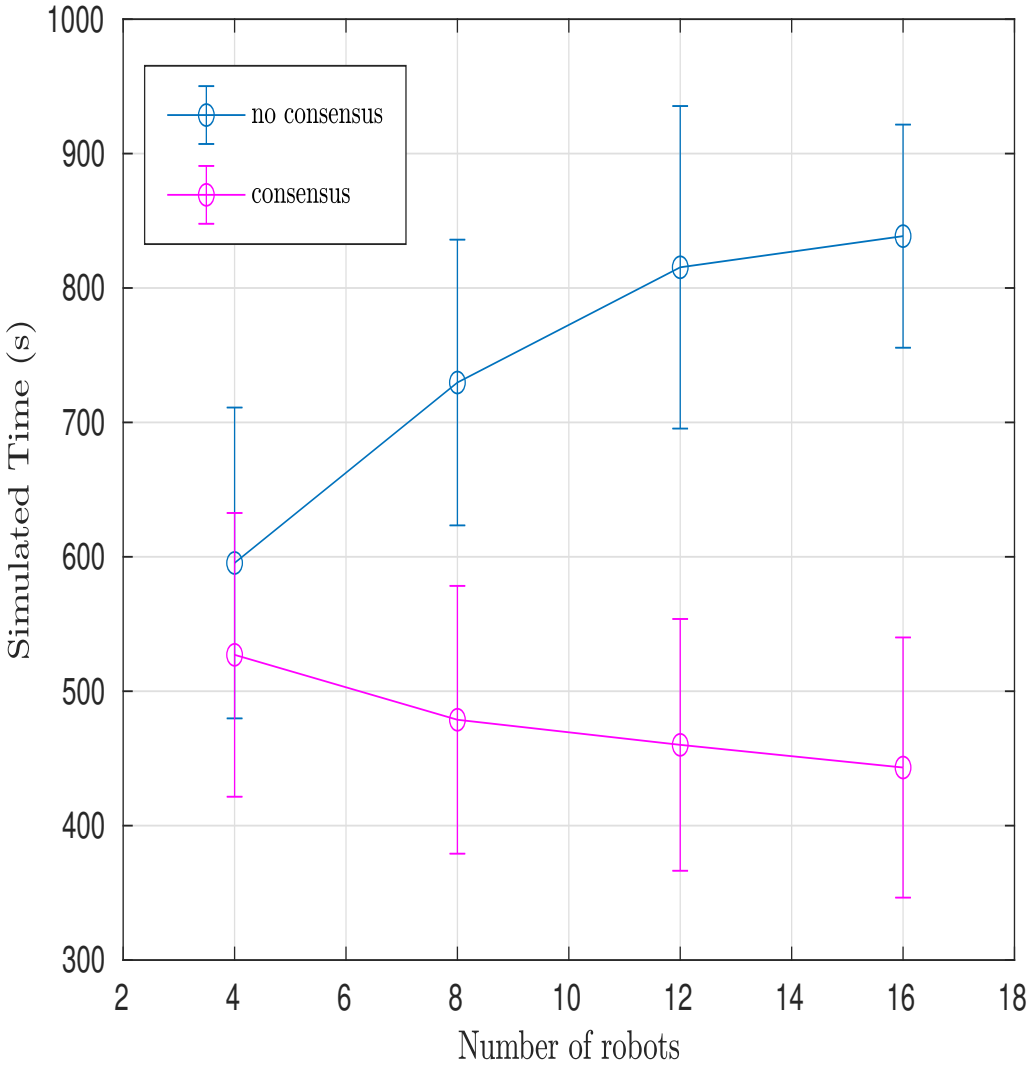


Figure 4.7: Time Until Convergence of Robots' Feature PMFs to the Reference PMF in Numerical Simulations of the Feature Reconstruction Strategy With and Without Consensus, For $N = \{4, 8, 12, 16\}$ Robots.

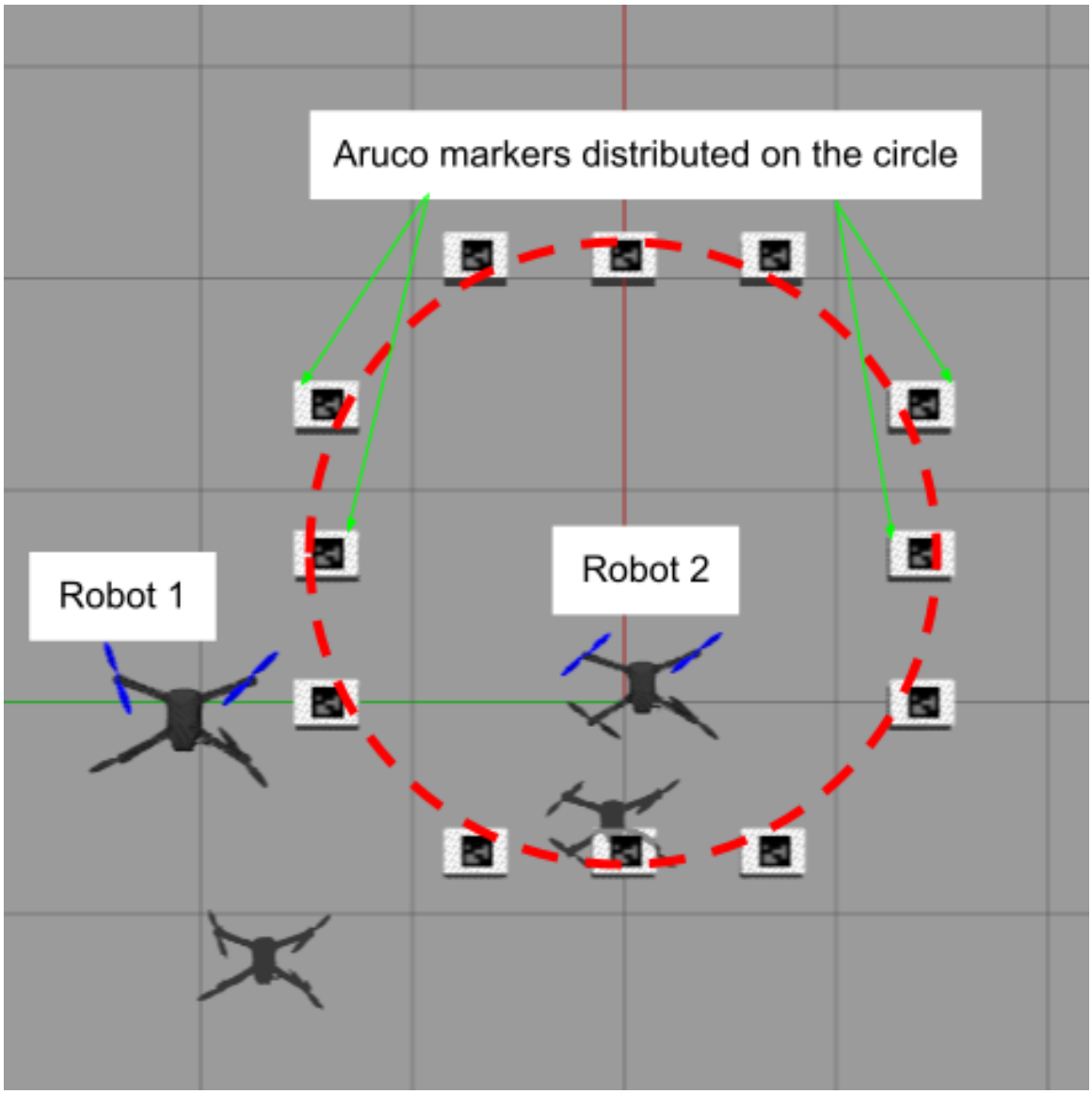


Figure 4.8: Software-in-the-Loop (SITL) Setup in Gazebo Using the Rotors (Furrer *et al.*, 2016) Package. Robots 1 and 2 Are Quadrotors, and the ArUco Markers Represent a Discrete Approximation of a Continuous Circular Feature (Red Dotted Line).

4.6.1 System Architecture

A diagram of our system architecture is shown in Figure 4.9. We use a hierarchical control scheme composed of low-level and high-level control blocks. The following is a description of each block in Figure 4.9

- *Gazebo Simulator*: Gazebo provides global localization for the simulated quadrotors. It outputs the 3D position of each robot a, $[p_x^a, p_y^a, p_z^a]$, and its orientation.

- *MAVROS*: The low-level controllers (*Low Level Unit*) use the ROS package MAVROS (Ermakov, 2004) to generate the control commands for the simulated quadrotors from the high-level controller (*High Level Unit*).

- *Gazebo Simulator*: Gazebo provides global localization for the simulated quadrotors. It outputs the 3D position of each robot a, $[p_x^a, p_y^a, p_z^a]$, and its orientation.

- *Random Walker*: The 3D quadrotor positions from Gazebo are used to determine the quadrotors' current nodes on the grid and the nodes they should move to in the next time step, according to the DTDS Markov chain in Equation (4.1). The new node locations are mapped to the commanded quadrotor velocities at time $k+1$, V_{com}^{k+1}, which are converted by *MAVROS* into velocity set points V_{sp}^{k+1} and communicated to the *Low Level Unit*. This block performs the necessary corrections to the quadrotors' poses, which are rendered in Gazebo.

- *Feature Detector*: An RGB image from the quadrotor's bottom-facing camera is used to detect the presence or absence of an ArUco marker at the quadrotor's current node, and the feature PMF $f_k^a(s)$ is computed according to Equation (4.3).

- *ADHOC*: When two quadrotors are located at the same node, they exchange

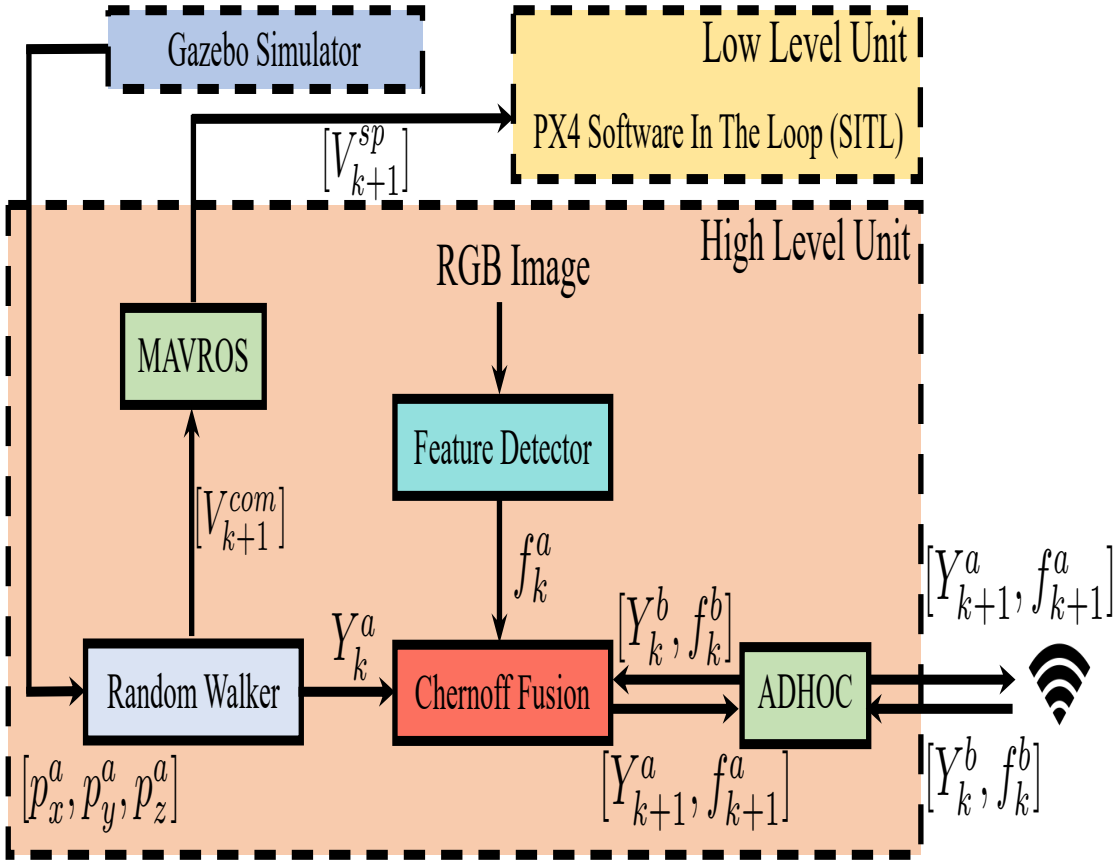


Figure 4.9: SITL Simulation System Architecture Block Diagram.

their feature PMFs using the ZeroMQ (Hintjens, 2013) asynchronous messaging library.

- *Chernoff Fusion*: This block executes the fusion protocol in Algorithm 6. It computes the quadrotor's feature PMF from its own measurements and from the feature PMFs transmitted by other quadrotors at its current node through the *ADHOC* block.

4.7 Simulation Results

Figure 4.10 through Figure 4.13 plot the feature PMFs computed by both quadrotors during two SITL simulation runs. Figure 4.10 and Figure 4.11 show the feature PMFs at $k = 240$ s (4 min) and $k = 330$ s (5.5 min), respectively, of the first simulation run. The figures indicate that *Robot 1* reconstructs most of the feature distribution within 4 min, and both robots fully reconstruct the distribution within 5.5 min.

Figure 4.12 and Figure 4.13 plot the feature PMFs at $k = 240$ s (4 min) and $k = 530$ s (~ 8.8 min), respectively, of the second simulation run. Figure 4.12 shows that in this simulation run, the two robots do not meet and exchange feature PMFs within the first 4 min, since their feature PMFs are completely distinct at that time. By ~ 8.8 min, Figure 4.13 shows that both robots have fully reconstructed the feature distribution, which matches the reconstructed distribution in Figure 4.11. Thus, Figure 4.10 through Figure 4.13 demonstrate that our approach ultimately results in accurate feature reconstruction, but that the convergence time to full reconstruction can differ between runs due to the randomness in the robot paths over the grid.

4.8 Real World Constraints

The performance of our approach in real-world environments will be affected by aerodynamic interactions between quadrotors, uncertainty in positioning, and feature occlusion by a quadrotor that enters other's field of view. In the SITL simulations, it is difficult to accurately simulate the aerodynamic disturbance on a quadrotor caused by the downwash of a quadrotor flying or hovering above it. These disturbances can be rejected by incorporating a robust disturbance observer (Mishra *et al.*, 2019) into the quadrotor's low-level flight control strategy. Alternatively, the quadrotors could

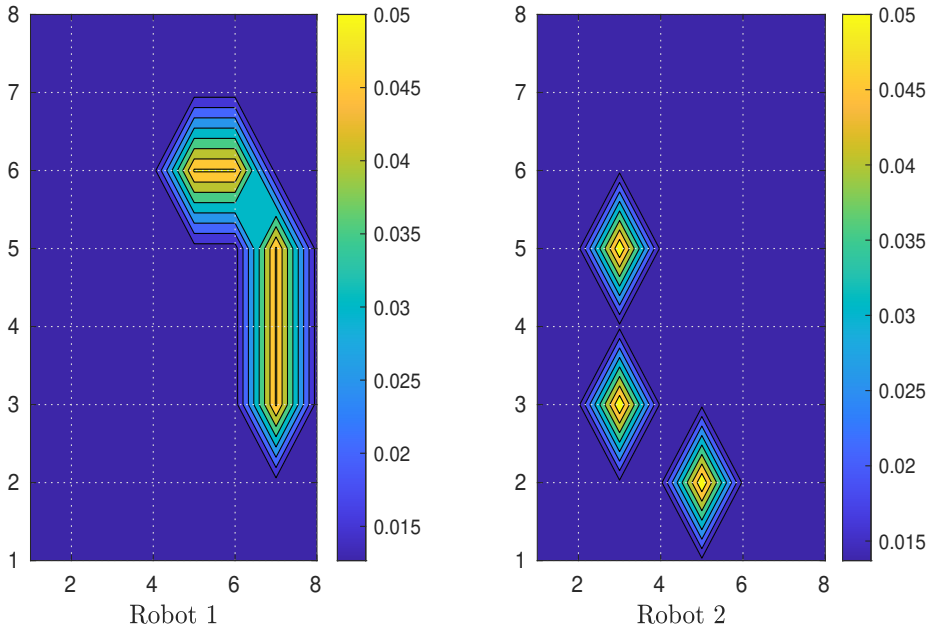


Figure 4.10: Feature PMF $f_k^a(s)$ Of 2 Robots at Time $k = 240$ s In the First Run of the SITL Simulation.

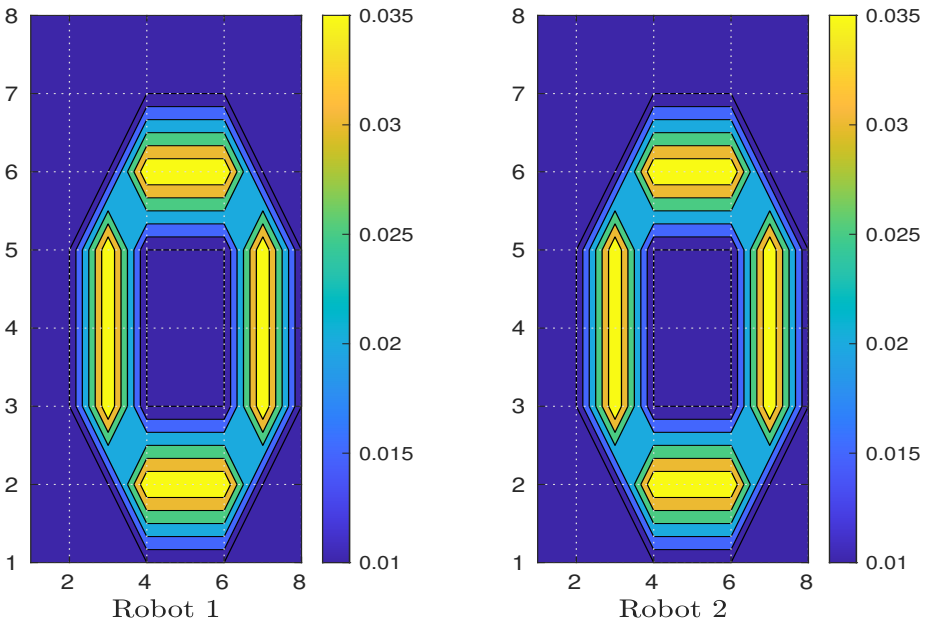


Figure 4.11: Feature PMF $f_k^a(s)$ Of 2 Robots at Time $k = 330$ s In the First Run of the SITL Simulation.

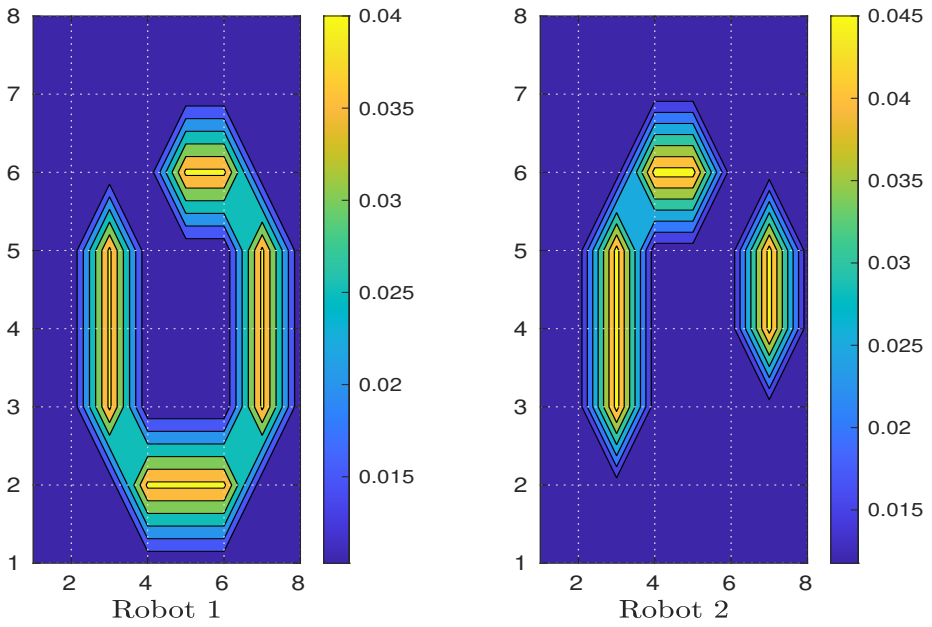


Figure 4.12: Feature PMF $f_k^a(s)$ Of 2 Robots at Time $k = 240$ s In the Second Run of the SITL Simulation.

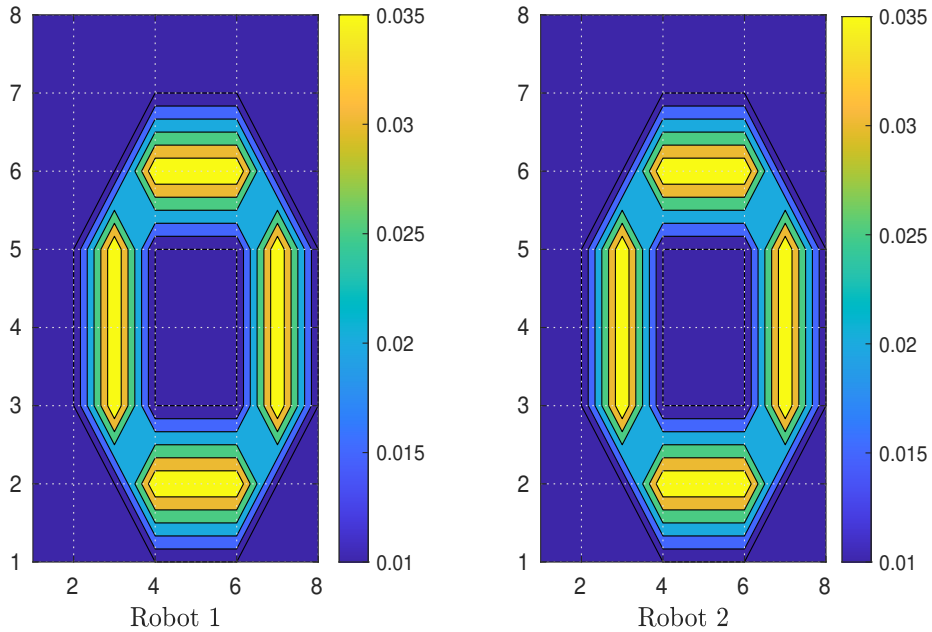


Figure 4.13: Feature PMF $f_k^a(s)$ Of 2 Robots at Time $k = 530$ s In the Second Run of the SITL Simulation.

fly at the same altitude and employ controllers for inter-robot collision avoidance, e.g. using control barrier functions (Wang *et al.*, 2017). Quadrotors may miss feature detections due to misalignment of their camera image window with the ground or to occlusion of the feature by another quadrotor flying below it. Such misalignment's and occlusions sometimes occurred in our SITL simulations; however, the accuracy of the feature reconstruction despite these occurrences demonstrates the robustness of our approach to the resulting missed feature detections.

4.9 Conclusion

In this chapter, we proposed a decentralized multi-robot strategy for reconstructing a discrete feature distribution on a finite spatial grid. Robots update their estimate of the feature distribution using their own measurements during random-walk exploration and estimates from nearby robots, combined using a distributed Chernoff fusion protocol. Our strategy extends established results on consensus of opinion pools for fixed, strongly connected networks to networks with Markovian switching dynamics. We provide theoretical guarantees on convergence to the actual feature distribution in an almost sure sense, and we validate the strategy in both numerical simulations and SITL simulations with quadrotors.

Chapter 5

CONCLUSIONS AND POSSIBLE FUTURE RESEARCH DIRECTIONS

5.1 Conclusion

In this dissertation, we explored the formulation and implementation of consensus strategies for homogeneous swarms that follow random mobility models and use local communication. In Chapter 2, we theoretically proved the convergence of such a probabilistic consensus strategy for detecting the presence of a static target. In Chapter 3, we formulated another consensus strategy for tracking multiple static targets by a swarm of aerial robots. This work used an estimation technique from the Random Finite Set (RFS) framework to account for the limited field of view (FOV) of the robots' sensors and for the uncertainty in both the number of targets and their positions. In Chapter 4, we used results on opinion pools to extend this approach to a consensus strategy for the decentralized fusion of discrete feature distributions, modeled as occupancy grid maps. In this work, we also showed the effectiveness of consensus by comparing it with a no-consensus strategy.

5.2 Future Research

In this section, we will discuss some future directions of this research applied to localization and mapping applications and human-swarm interaction.

5.2.1 Swarm Mapping

Localization and mapping problems with multi-robot systems have been extensively studied since early works from Jennings *et al.* (1999); Fox *et al.* (2000); Thrun

(2001), with the latest research involving machine learning and artificial intelligence techniques for solving subproblems in simultaneous localization and mapping (SLAM) Guo *et al.* (2020). Swarm systems are designed to be scalable and redundant. To achieve this, most swarm systems also need to be inexpensive. The paper Kegeleirs *et al.* (2021) lists various challenges that are associated with deploying swarm systems for mapping applications. Hardware and computation-related challenges do limit the current application of swarms to mapping problems; however, developments in low-power compute and edge compute technologies may provide a possible solution. In a collaborative SLAM architecture, the SLAM problem for a pair of robots a, b is formulated as a maximum likelihood estimation problem as follows:

$$\hat{X}_a, \hat{X}_b \doteq \arg\max_{X_a, X_b} Pr(Z_a, Z_b, Z_{ab} | X_a, X_b),$$

where X_a, X_b represent the state variables, for example the positions and velocities of features, and Z_a, Z_b represent the measurements taken by robots a and b, respectively. In a decentralized collaborative SLAM formulation, each robot computes a local map, and a consistent global map is generated over time by combining robots' measurements to reduce the error from their individual measurements. Such a collaborative SLAM approach will not work for swarms, as inter-robot measurements (which provide the robots' pose transformations from their respective co-ordinate frames) and measurements of certain areas of the environment in general may not be available from all the robots. Swarm systems typically utilize local sensing and local sharing of information, which results in a combinatorial data association problem for generating a consistent global map. This problem can be further complicated by the possible lack of overlapping measurements between members of the swarm that meet and exchange data. This may cause an inconsistent global map to be generated by the robot swarm.

For a swarm system, the data association problem could be solved with techniques from the Random Finite Set (RFS) framework, since it provides a metric for comparing and combining individual robot maps. The global map and the robot measurements can be modeled as Random Finite Sets, which allows for the joint estimation of the number and locations of features and the robot trajectories. Additionally, robots in a swarm can use visual inertial odometry methods to improve their estimates of their own trajectories, and thus generate more consistent local maps. Given these improved local maps, probabilistic consensus algorithms similar to the ones presented in this thesis can be developed to enable the swarm to produce a consistent global map. This potential increase in consistency, however, comes at the cost of requiring more on-board computational resources and power and more accurate sensors. The metric for analyzing map consistency would also need to be modified to account for the combinatorial problems associated with interaction and information exchange among time-varying sets of robots, and to ensure geometric consistency of the local-local map (map generated by a set of interacting robots), local-global map (map formed by the robot and the ground truth map), and global map which is the ground truth map.

5.2.2 Human Swarm Interaction

Human-Swarm Interaction (HSI) Brown *et al.* (2016) describes the mechanisms by which a human operator monitors and controls a swarm of robots. The capacity to control a large group of robots depends on the cognitive bandwidth of the human operator Chen and Barnes (2014). With increasing numbers of robots, the human operator will not able to control the granular behavior of each robot, but could be able to control and modify the collective behavior of the swarm. HSI involves modeling this collective behavior with the human operator in the loop and designing robot

control policies to achieve desired swarm tasks. HSI can be implemented using a *leader-follower* control architecture, where the *leader* is the human operator and the *followers* are the robots. The leader is responsible for generating the control commands that the followers interpret to perform the required task, such as dispersion, flocking, and formation control. However, this architecture would require all the followers to continuously monitor the leader, and thus would be constrained by the robots' sensing and communication capabilities (e.g., the dispersion of a swarm could be limited by the line of sight of the robots' sensors).

One way of counteracting this limitation would be to use a hierarchical *"supervisor-leader-follower"* scheme. Here, the *supervisor* is the human operator, the *leader* is one of the robots, and the *followers* are the remaining robots in the swarm. The leader in this scheme could be equipped with more resources than the followers, such as a larger power source, faster processors, or communication gear. The leader would interpret the intention of the human supervisor and communicate the required control command that the followers need to execute to achieve the desired behavior.

One possible application for this scheme would be to coordinate a dance performance by a human supervisor with a swarm of robots. Dancing with robots, also called *choreorobotics* Cuan (2021), is an emerging field in which models from swarm robotics can be used to formulate control and estimation policies that implement a choreographed performance between humans and robots.

The framework from this dissertation can be extended to develop a supervisor-leader-follower consensus scheme in which the followers try to reach an agreement on the activity to perform based on the signal or parameter broadcast by the leader, which responds to discrete commands from a human operator. Each follower can be programmed to respond to the leader's broadcast by executing prescribed trajectories that are generated by a path planning framework. The leader would need to have

the appropriate sensors or communication hardware (e.g., RGBD cameras, long-range radio antennas) to be able to discern inputs from the human supervisor, which could be, for example, in the form of hand gestures, speech, or computer commands. To reduce the complexity of the inputs that the leader must detect or estimate, these inputs can be modeled as a set of discrete states that evolve according to a Markov chain, with each state corresponding to a particular input from the human supervisor. The states can be identified *a priori* by a neural network that acts as a classifier, using a supervised learning framework to compute the probabilities of transition between pairs of states.

Angeli, D. and P.-A. Bliman, "Extension of a result by moreau on stability of leaderless multi-agent systems.", in "Proceedings of the 44th IEEE Conference on Decision and Control", pp. 759–764 (IEEE, 2005).

Bailey, T., S. Julier and G. Agamennoni, "On conservative fusion of information with unknown non-Gaussian dependence", in "15th International Conference on Information Fusion", pp. 1876–1883 (2012).

Bajo, J., F. De la Prieta, J. M. Corchado and S. Rodríguez, "A low-level resource allocation in an agent-based cloud computing platform", Applied Soft Computing **48**, 716–728 (2016).

Bandyopadhyay, S., S.-J. Chung and F. Y. Hadaegh, "Probabilistic and distributed control of a large-scale swarm of autonomous agents", IEEE Transactions on Robotics **33**, 5, 1103–1123 (2017).

Bar-Shalom, Y. and T. E. Fortmann, *Tracking and data association*, vol. 179 of *Mathematics in Science and Engineering* (Elsevier Science, 1988).

Battistelli, G. and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability", Automatica **50**, 3, 707–718 (2014).

Biswal, S., K. Elamvazhuthi and S. Berman, "Fastest mixing markov chain on a compact manifold", in "2019 IEEE 58th Conference on Decision and Control (CDC)", pp. 547–554 (IEEE, 2019).

Brown, D. S., M. A. Goodrich, S.-Y. Jung and S. Kerman, "Two invariants of human-swarm interaction", Tech. rep., Air Force Research Lab Rome NY ROME United States (2016).

Burgard, W., M. Moors, C. Stachniss and F. E. Schneider, "Coordinated multi-robot exploration", IEEE Transactions on Robotics **21**, 3, 376–386 (2005).

Calafiore, G. C. and F. Abrate, "Distributed linear estimation over sensor networks", International Journal of Control **82**, 5, 868–882 (2009).

Camazine, S., J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraula and E. Bonabeau, *Self-organization in biological systems* (Princeton university press, 2020).

Chen, J. Y. and M. J. Barnes, "Human–agent teaming for multirobot control: A review of human factors issues", IEEE Transactions on Human-Machine Systems **44**, 1, 13–29 (2014).

Clark, D. E., K. Panta and B.-N. Vo, "The gm-phd filter multiple target tracker", in "2006 9th International Conference on Information Fusion", pp. 1–8 (IEEE, 2006).

Cuan, C., "Dances with robots: Choreographing, correcting, and performing with moving machines", The Drama Review **65**, 1, 124–143 (2021).

Daley, D. J. and D. Vere-Jones, *An introduction to the theory of point processes: volume II: general theory and structure* (Springer Science & Business Media, 2007).

Dames, P. and V. Kumar, "Cooperative multi-target localization with noisy sensors", in "2013 IEEE International Conference on Robotics and Automation (ICRA)", pp. 1877–1883 (IEEE, 2013).

Dames, P. M., "Distributed multi-target search and tracking using the phd filter", Autonomous Robots pp. 1–17 (2019).

Dang, T., F. Mascarich, S. Khattak, C. Papachristos and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments", in "2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)", pp. 3105–3112 (IEEE, 2019).

Degroot, M. H., "Reaching a consensus", Journal of the American Statistical Association **69**, 345, 118–121 (1974).

Dimidov, C., G. Oriolo and V. Trianni, "Random walks in swarm robotics: An experiment with kilobots", in "Swarm Intelligence", edited by M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli and T. Stützle, pp. 185–196 (Springer International Publishing, Cham, 2016).

Domínguez, R., S. Cannella and J. M. Framinan, "Scope: a multi-agent system tool for supply chain network analysis", in "IEEE EUROCON 2015-International Conference on Computer as a Tool (EUROCON)", pp. 1–5 (IEEE, 2015).

Dorigo, M., M. Birattari and T. Stutzle, "Ant colony optimization", IEEE computational intelligence magazine **1**, 4, 28–39 (2006).

Dos Santos, D. S. and A. L. Bazzan, "Distributed clustering for group formation and task allocation in multiagent systems: A swarm intelligence approach", Applied Soft Computing **12**, 8, 2123–2131 (2012).

Duarte, M., V. Costa, J. Gomes, T. Rodrigues, F. Silva, S. M. Oliveira and A. L. Christensen, "Evolution of collective behaviors for a real swarm of aquatic surface robots", PloS one **11**, 3, e0151834 (2016).

Elamvazhuthi, K. and S. Berman, "Mean-field models in swarm robotics: A survey", Bioinspiration & Biomimetics **15**, 1, 015001 (2019).

Elamvazhuthi, K., Z. Kakish, A. Shirsat and S. Berman, "Controllability and stabilization for herding a robotic swarm using a leader: A mean-field approach", IEEE Transactions on Robotics **37**, 2, 418–432 (2020).

Ermakov, V., "MAVROS; http://wiki.ros.org/mavros", URL `http://wiki.ros.org/mavros` (2004).

Farrell, W. J. and C. Ganesh, "Generalized Chernoff fusion approximation for practical distributed data fusion", in "International Conference on Information Fusion", pp. 555–562 (2009).

Ferrari-Trecate, G., A. Buffa and M. Gati, "Analysis of coordination in multi-agent systems through partial difference equations", IEEE Transactions on Automatic Control **51**, 6, 1058–1063 (2006).

Fox, D., W. Burgard, H. Kruppa and S. Thrun, "A probabilistic approach to collaborative multi-robot localization", Autonomous robots **8**, 3, 325–344 (2000).

Fox, D., J. Ko, K. Konolige, B. Limketkai, D. Schulz and B. Stewart, "Distributed multirobot exploration and mapping", Proceedings of the IEEE **94**, 7, 1325–1339 (2006).

Fu, J. and J. Wang, "Adaptive coordinated tracking of multi-agent systems with quantized information", Systems & Control Letters **74**, 115–125 (2014).

Furrer, F., M. Burri, M. Achtelik and R. Siegwart, *RotorS—A Modular Gazebo MAV Simulator Framework*, pp. 595–625 (Springer International Publishing, Cham, 2016).

George, J. and R. A. Freeman, "Robust dynamic average consensus algorithms", IEEE Transactions on Automatic Control **64**, 11, 4615–4622 (2019).

Godsil, C. and G. F. Royle, *Algebraic graph theory*, vol. 207 (Springer Science & Business Media, 2001).

Grimmett, G. and D. Stirzaker, *Probability and random processes* (Oxford University Press, 2001).

Guo, Y., H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun, "Deep learning for 3d point clouds: A survey", IEEE transactions on pattern analysis and machine intelligence **43**, 12, 4338–4364 (2020).

Halloy, J., G. Sempo, G. Caprari, C. Rivault, M. Asadpour, F. Tâche, I. Saïd, V. Durier, S. Canonge, J. M. Amé *et al.*, "Social integration of robots into groups of cockroaches to control self-organized choices", Science **318**, 5853, 1155–1158 (2007).

Han, Z., K. Guo, L. Xie and Z. Lin, "Integrated relative localization and leader–follower formation control", IEEE Transactions on Automatic Control **64**, 1, 20–34 (2018).

Hernandez-Leal, P., B. Kartal and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning", Autonomous Agents and Multi-Agent Systems **33**, 6, 750–797 (2019).

Hintjens, P., *ZeroMQ: messaging for many applications* (O'Reilly Media, Inc., 2013).

Horn, R. A. and C. R. Johnson, *Matrix analysis* (Cambridge University Press, 1990).

Howard, A., L. E. Parker and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection", The International Journal of Robotics Research **25**, 5-6, 431–447 (2006).

Hu, J., L. Xie, K.-Y. Lum and J. Xu, "Multiagent information fusion and cooperative control in target search", IEEE Transactions on Control Systems Technology **21**, 4, 1223–1235 (2012).

Hu, P., J. Ziglar, D. Held and D. Ramanan, "What you see is what you get: Exploiting visibility for 3d object detection", in "Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition", pp. 11001–11009 (2020).

Hung, P. D., T. Q. Vinh and T. D. Ngo, "A scalable, decentralised large-scale network of mobile robots for multi-target tracking", in "Intelligent Autonomous Systems 13", pp. 621–637 (Springer, 2016).

Husain, A., H. Jones, B. Kannan, U. Wong, T. Pimentel, S. Tang, S. Daftry, S. Huber and W. L. Whittaker, "Mapping planetary caves with an autonomous, heterogeneous robot team", in "2013 IEEE Aerospace Conference", pp. 1–13 (IEEE, 2013).

Jennings, C., D. Murray and J. J. Little, "Cooperative robot localization with vision-based mapping", in "Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)", vol. 4, pp. 2659–2665 (IEEE, 1999).

Jensfelt, P. and S. Kristensen, "Active global localization for a mobile robot using multiple hypothesis tracking", IEEE Transactions on Robotics and Automation **17**, 5, 748–760 (2001).

Jing, G. and L. Wang, "Multiagent flocking with angle-based formation shape control", IEEE Transactions on Automatic Control **65**, 2, 817–823 (2019).

Kamath, S., E. Meisner and V. Isler, "Triangulation based multi target tracking with mobile sensor networks", in "Proceedings 2007 IEEE International Conference on Robotics and Automation", pp. 3283–3288 (IEEE, 2007).

Kazakova, V. A., A. S. Wu and G. R. Sukthankar, "Respecializing swarms by forgetting reinforced thresholds", Swarm Intelligence **14**, 3, 171–204 (2020).

Kegeleirs, M., D. Garzón Ramos and M. Birattari, "Random walk exploration for swarm mapping", in "Towards Autonomous Robotic Systems", edited by K. Althoefer, J. Konstantinova and K. Zhang, pp. 211–222 (Springer International Publishing, Cham, 2019).

Kegeleirs, M., G. Grisetti and M. Birattari, "Swarm slam: Challenges and perspectives", Frontiers in Robotics and AI **8**, 23 (2021).

Kellermann, R., T. Biehle and L. Fischer, "Drones for parcel and passenger transportation: A literature review", Transportation Research Interdisciplinary Perspectives **4**, 100088 (2020).

Kennedy, J. and R. Eberhart, "Particle swarm optimization", in "Proceedings of ICNN'95-international conference on neural networks", vol. 4, pp. 1942–1948 (IEEE, 1995).

Khodaverdian, S., "On the synchronization of linear heterogeneous multi-agent systems in cycle-free communication networks", in "Proceedings of 2014 International Conference on Modelling, Identification & Control", pp. 190–195 (IEEE, 2014).

Koenig, N. and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator", in "2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)", vol. 3, pp. 2149–2154 (IEEE, 2004).

Krieger, M. J., J.-B. Billeter and L. Keller, "Ant-like task allocation and recruitment in cooperative robots", Nature **406**, 6799, 992–995 (2000).

Langley, P., J. E. Laird and S. Rogers, "Cognitive architectures: Research issues and challenges", Cognitive Systems Research **10**, 2, 141–160 (2009).

Levin, D. A. and Y. Peres, *Markov chains and mixing times*, vol. 107 (American Mathematical Society, 2017).

Li, H., G. Chen, T. Huang, Z. Dong, W. Zhu and L. Gao, "Event-triggered distributed average consensus over directed digital networks with limited communication bandwidth", IEEE transactions on cybernetics **46**, 12, 3098–3110 (2016).

Lin, C., Z. Lin, R. Zheng, G. Yan and G. Mao, "Distributed source localization of multi-agent systems with bearing angle measurements", IEEE Transactions on Automatic Control **61**, 4, 1105–1110 (2015).

Lin, Z., L. Wang, Z. Han and M. Fu, "Distributed formation control of multi-agent systems using complex laplacian", IEEE Transactions on Automatic Control **59**, 7, 1765–1777 (2014).

Liu, B., H. Su, R. Li, D. Sun and W. Hu, "Switching controllability of discrete-time multi-agent systems with multiple leaders and time-delays", Applied Mathematics and Computation **228**, 571–588 (2014).

Liu, Z., X. You, H. Yang and L. Zhao, "Leader-following consensus of heterogeneous multi-agent systems with packet dropout", International Journal of Control, Automation and Systems **13**, 5, 1067–1075 (2015).

Mahdoui, N., V. Frémont and E. Natalizio, "Communicating multi-uav system for cooperative slam-based exploration", Journal of Intelligent & Robotic Systems **98**, 2, 325–343 (2020).

Mahler, R., D. Hall and J. Llinas, "Random set theory for target tracking and identification", in "Data Fusion Hand Book", p. 14 (CRC Press Boca Raton, 2001).

Mahler, R. P., "Multitarget bayes filtering via first-order multitarget moments", IEEE Transactions on Aerospace and Electronic systems **39**, 4, 1152–1178 (2003).

Mahler, R. P. S., *Statistical Multisource-Multitarget Information Fusion*, vol. 685 (Artech House, Inc., USA, 2007).

Martinez, F., E. Jacinto and D. Acero, "Brownian motion as exploration strategy for autonomous swarm robots", in "2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)", pp. 2375–2380 (IEEE, 2012).

Matei, I., N. C. Martins and J. S. Baras, "Consensus problems with directed Markovian communication patterns", in "2009 American Control Conference", pp. 1298–1303 (IEEE, 2009).

Meier, L., D. Honegger and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms", in "IEEE International Conference on Robotics and Automation (ICRA)", pp. 6235–6240 (2015).

Merabet, G. H., M. Essaaidi, H. Talei, M. R. Abid, N. Khalil, M. Madkour and D. Benhaddou, "Applications of multi-agent systems in smart grids: A survey", in "2014 International conference on multimedia computing and systems (ICMCS)", pp. 1088–1094 (IEEE, 2014).

Mesbahi, M. and M. Egerstedt, *Graph theoretic methods in multiagent networks* (Princeton University Press, 2010).

Michael, N., S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno *et al.*, "Collaborative mapping of an earthquake damaged building via ground and aerial robots", in "Field and Service Robotics", pp. 33–47 (Springer, 2014).

Mishra, S., T. Rakstad and W. Zhang, "Robust attitude control for quadrotors based on parameter optimization of a nonlinear disturbance observer", Journal of Dynamic Systems, Measurement, and Control **141**, 8 (2019).

Murphy, R. R., S. Tadokoro and A. Kleiner, "Disaster robotics", in "Springer Handbook of Robotics", pp. 1577–1604 (Springer, 2016).

Nagatani, K., S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima *et al.*, "Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots", Journal of Field Robotics **30**, 1, 44–63 (2013).

Nowzari, C., E. Garcia and J. Cortés, "Event-triggered communication and control of networked systems for multi-agent consensus", Automatica **105**, 1–27 (2019).

Olfati-Saber, R. and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays", IEEE Transactions on automatic control **49**, 9, 1520–1533 (2004).

Parasuraman, R., J. Kim, S. Luo and B.-C. Min, "Multipoint rendezvous in multirobot systems", IEEE Transactions on Cybernetics **50**, 1, 310–323 (2018).

Preiss, J. A., W. Honig, G. S. Sukhatme and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm", in "2017 IEEE International Conference on Robotics and Automation (ICRA)", pp. 3299–3304 (IEEE, 2017).

Rameau, F., O. Bailo, J. Park, K. Joo and I. S. Kweon, "Real-time multi-car localization and see-through system", International Journal of Computer Vision pp. 1–21 (2022).

Reid, D., "An algorithm for tracking multiple targets", IEEE Transactions on Automatic Control **24**, 6, 843–854 (1979).

Ren, W. and R. W. Beard, "Consensus of information under dynamically changing interaction topologies", in "2004 American Control Conference", vol. 6, pp. 4939–4944 (IEEE, 2004).

Ren, W. and R. W. Beard, "Consensus tracking with a reference state", Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications pp. 55–73 (2008).

Ren, W., R. W. Beard and E. M. Atkins, "Information consensus in multivehicle cooperative control", IEEE Control Systems **27**, 2, 71–82 (2007).

Reynolds, C. W., "Flocks, herds and schools: A distributed behavioral model", in "Proceedings of the 14th annual conference on Computer graphics and interactive techniques", pp. 25–34 (1987).

Rizk, Y., M. Awad and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey", ACM Computing Surveys (CSUR) **52**, 2, 1–31 (2019).

Rocha, R., J. Dias and A. Carvalho, "Cooperative multi-robot systems:: A study of vision-based 3-D mapping using information theory", in "IEEE International Conference on Robotics and Automation (ICRA)", pp. 384–389 (2005).

Ross, S. M., *Introduction to probability models* (Academic Press, 2014).

Rubenstein, M., C. Ahler and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors", in "2012 IEEE international conference on robotics and automation", pp. 3293–3298 (IEEE, 2012).

Russell, S. and P. Norvig, *Artificial intelligence: a modern approach* (Prentice Hall, 2010).

Sampedro, C., A. Rodriguez-Ramos, H. Bavle, A. Carrio, P. de la Puente and P. Campoy, "A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques", Journal of Intelligent & Robotic Systems **95**, 2, 601–627 (2019).

Schulz, D., W. Burgard, D. Fox and A. B. Cremers, "Tracking multiple moving objects with a mobile robot", in "Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001", vol. 1 (IEEE, 2001).

Serfozo, R., *Basics of applied stochastic processes* (Springer Science & Business Media, 2009).

Shirsat, A. and S. Berman, "Decentralized multi-target tracking with multiple quadrotors using a PHD filter", in "AIAA Scitech 2021 Forum", (2021).

Shirsat, A., K. Elamvazhuthi and S. Berman, "Multi-robot target search using probabilistic consensus on discrete markov chains", in "2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)", pp. 108–115 (IEEE, 2020).

Shirsat, A., S. Mishra, W. Zhang and S. Berman, "Probabilistic consensus on feature distribution for multi-robot systems with markovian exploration dynamics", In revision for IEEE Robotics and Automation Letters (RA-L) (2022).

Singh, B., R. Kumar and V. P. Singh, "Reinforcement learning in robotic applications: a comprehensive survey", Artificial Intelligence Review pp. 1–46 (2021).

Song, Q., F. Liu, H. Su and A. V. Vasilakos, "Semi-global and global containment control of multi-agent systems with second-order dynamics and input saturation", International Journal of Robust and Nonlinear Control **26**, 16, 3460–3480 (2016).

Spanos, D. P., R. Olfati-Saber and R. M. Murray, "Dynamic consensus on mobile networks", in "IFAC World Congress", pp. 1–6 (Citeseer, 2005).

Stauch, J., T. Bessell, M. Rutten, J. Baldwin, M. Jah and K. Hill, "Joint probabilistic data association and smoothing applied to multiple space object tracking", Journal of Guidance, Control, and Dynamics **41**, 1, 19–33 (2018).

Stone, M., "The opinion pool", The Annals of Mathematical Statistics **32**, 4, 1339–1342 (1961).

Sumpter, D. J., *Collective animal behavior* (Princeton University Press, 2010).

Sun, Q., Y. Tian and M. Diao, "Cooperative localization algorithm based on hybrid topology architecture for multiple mobile robot system", IEEE Internet of Things Journal **5**, 6, 4753–4763 (2018).

Sung, Y. and P. Tokekar, "Algorithm for searching and tracking an unknown and varying number of mobile targets using a limited fov sensor", in "2017 IEEE International Conference on Robotics and Automation (ICRA)", pp. 6246–6252 (IEEE, 2017).

Tavakoli, M., J. Carriere and A. Torabi, "Robotics, smart wearable technologies, and autonomous intelligent systems for healthcare during the covid-19 pandemic: An analysis of the state of the art and future vision", Advanced Intelligent Systems **2**, 7, 2000071 (2020).

Thipphavong, D. P., R. Apaza, B. Barmore, V. Battiste, B. Burian, Q. Dao, M. Feary, S. Go, K. H. Goodrich, J. Homola *et al.*, "Urban air mobility airspace integration concepts and considerations", in "2018 Aviation Technology, Integration, and Operations Conference", p. 3676 (2018).

Thrun, S., "A probabilistic on-line mapping algorithm for teams of mobile robots", The International Journal of Robotics Research **20**, 5, 335–363 (2001).

Vaquero, T., M. Troesch, M. Sanchez Net, J. Gao and S. Chien, "Energy-aware data routing for disruption tolerant networks in planetary cave exploration", International Workshop on Planning and Scheduling for Space (IWPSS) (2019).

Vincent, R., D. Fox, J. Ko, K. Konolige, B. Limketkai, B. Morisset, C. Ortiz, D. Schulz and B. Stewart, "Distributed multirobot exploration, mapping, and task allocation", Annals of Mathematics and Artificial Intelligence **52**, 2-4, 229–255 (2008).

Vo, B.-N. and W.-K. Ma, "The gaussian mixture probability hypothesis density filter", IEEE Transactions on Signal Processing **54**, 11, 4091–4104 (2006).

Vrancken, J. and M. D. S. Soares, "A real-life test bed for multi-agent monitoring of road network performance", International journal of critical infrastructures **5**, 4, 357–367 (2009).

Wagner, I. A., M. Lindenbaum and A. M. Bruckstein, "Robotic exploration, Brownian motion and electrical resistance", in "International Workshop on Randomization and Approximation Techniques in Computer Science", pp. 116–130 (Springer, 1998).

Wang, L., A. D. Ames and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness", in "2017 IEEE International Conference on Robotics and Automation (ICRA)", pp. 3293–3298 (2017).

Wang, W., Z. He, P. Shi, W. Wu, Y. Jiang, B. An, Z. Hao and B. Chen, "Strategic social team crowdsourcing: Forming a team of truthful workers for crowdsourcing in social networks", IEEE Transactions on Mobile Computing **18**, 6, 1419–1432 (2018).

Wilson, S., R. Gameros, M. Sheely, M. Lin, K. Dover, R. Gevorkyan, M. Haberland, A. Bertozzi and S. Berman, "Pheeno, a versatile swarm robotic research and education platform", IEEE Robotics and Automation Letters **1**, 2, 884–891 (2016).

Winfield, A. F., "Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots", in "Distributed Autonomous Robotic Systems 4", pp. 273–282 (Springer, 2000).

Xiao, L., S. Boyd and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus", in "International Symposium on Information Processing in Sensor Networks", pp. 63–70 (2005).

Yang, G.-Z., B. J. Nelson, R. R. Murphy, H. Choset, H. Christensen, S. H. Collins, P. Dario, K. Goldberg, K. Ikuta, N. Jacobstein *et al.*, "Combating covid-19—the role of robotics in managing public health and infectious diseases", (2020).

Yoon, K., Y.-m. Song and M. Jeon, "Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views", IET Image Processing **12**, 7, 1175–1184 (2018).

Yu, X. and M. A. Hsieh, "Synthesis of a time-varying communication network by robot teams with information propagation guarantees", IEEE Robotics and Automation Letters **5**, 2, 1413–1420 (2020).

Zhang, K., Z. Yang, H. Liu, T. Zhang and T. Basar, "Finite-sample analysis for decentralized batch multi-agent reinforcement learning with networked agents", IEEE Transactions on Automatic Control (2021).

Zou, D., P. Tan and W. Yu, "Collaborative visual slam for multiple agents: A brief survey", Virtual Reality & Intelligent Hardware **1**, 5, 461–482 (2019).