Dynamic Coverage Control and Estimation in Collaborative Networks of Human-Aerial/Space Co-Robots

by

William M. Bentz

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2020

Doctoral Committee:

Assistant Professor Dimitra Panagou, Chair
Professor Ella M. Atkins
Professor Ilya V. Kolmanovsky
Assistant Professor Ram Vasudevan

William M. Bentz

wbentz@umich.edu

ORCID iD: 0000-0001-5916-362X

I dedicate this dissertation to my mother and father who encouraged an inventive mind, to my sisters who shared their love of outer space with me, to my friends who keep me humble, to my mentor who taught me to infuse my experimental work with provably correct theory, and to my partner who loves and supports me always.

Acknowledgments

Over the past five years, I have enjoyed the great privilege of the life of an academic. Some would call the experience of a doctoral researcher an *isolating and emotionally taxing battle with imposter syndrome*. I do not disagree. There were certainly times at which I was overwhelmed with feelings of futility and hopelessness—questioning whether I was providing solutions to problems that no one had asked for. Nonetheless, those sporadic triumphs, those moments at which months of theoretical speculation were physically realized in the laboratory, provided the intermittent reinforcement one needs to maintain sanity and purpose.

I would never had found those moments of clarity had it not been for Enkhmurun Bayasgalan and Tru Hoang. I, along with these two Master's students, matriculated at the University of Michigan in the Fall of 2015. For two years, we called the Distributed Aerospace Systems and Control Laboratory our home. During that time, I was faced with an intense loss in my personal life that threw me into a deep depression. Tru and Murun provided my first real experience with friendship in Michigan. They came to the lab nearly every day and devoted an incredible amount of time towards teaching me how to solder, repair quadrotor frames, and program motor controllers. Together, we learned the ins and outs of the Robot Operating System and every nuance of our motion capture system. By the time they graduated, I had become an expert in the idiosyncrasies and unique characteristics of each computer, sensor, and piece of flight hardware within our lab. I could not have done this without their guidance and companionship. Furthermore, I may not have been emotionally suited to continue my PhD studies had it not been for our many lunches at Ahmo's, our trips to the movie theatre, and the hours we spent together at the Duderstadt Computer and Video Game Archive. I will always value our friendship and look fondly upon those formative years in my graduate education.

I also wish to acknowledge two staff members within the Department of Aerospace Engineering, namely Aaron Borgman and Denise Phelps. Aaron is an incredibly skilled computer engineer notable for his time at Intel having helped to roll out the Pentium 4 processor (the CPU that powered my childhood). Aaron always had an open door policy and I found myself frequenting his office for advice on circuit design, a direly needed solid state component, or a quick repair which my hands lacked the dexterity and experience to

accomplish. Aaron, along with the rest of the technical staff, serve as an incredible asset to the department. I am amazed time and again that so few of my fellow students are aware of this resource of knowledge.

Denise has been incredibly helpful towards navigating the world of university bureaucracy. I recall Denise introducing herself at my department orientation five years ago. She told us that she saw the graduate students of this department as her own kids and that we were free to stop by her office for any concern whether big or small. As a graduate student, few questions I posed ever had simple solutions—unless I posed them to Denise Phelps. Denise is a walking reference manual when it comes to graduation requirements, benefits enrollment, workspace allocation, and every miscellaneous university and department policy imaginable. Denise's knowledge, her smile, and her laugh have made my life, along with the lives of thousands of my colleagues over the last 40 years, that much easier.

I wish to offer a special *thank you* to Professor Ella Atkins. During my first year, I was fortunate enough to have taken her graduate course on aerospace information systems. This class truly opened my mind. In the period of a few months, Dr. Atkins introduced me to a wide range of topics including Markov decision processes, hybrid control systems, finite state automata, and Bayesian statistics. Each one of these tools has proven instrumental to the completion of my PhD. Had I chosen an alternative course in the winter of 2016, my CV would look quite a bit different.

I've saved the most important acknowledgement for last. I am so very grateful to Professor Dimitra Panagou for having provided the offer that would forever change my life. As an undergraduate student, I was jaded and cynical towards academia. When Dr. Panagou first contacted me, I was in the final stages of accepting an offer from the U.S. Navy to attend flight school. Dr. Panagou saw something in me that I had no idea existed. She perceived my passion for mathematics and my love of tinkering and recognized a potential that no academic had before. I would never have succeeded as a researcher without her unique blend of guidance, empathy, and patience mixed with the highest standards for unique and rigorous work.

Lest we forget—"No bucks, no Buck Rodgers." I wish to acknowledge the support of the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-14-2-0001 U.S. Army TARDEC in Warren, MI. I also wish to acknowledge the support of an Early Career Faculty grant from NASA. With respect to the latter organization, it is my sincere wish that the work in this dissertation will significantly contribute to and shape the agency's mission of space exploration in the pursuit of human knowledge.

Table of Contents

List of Figures

Figure

List of Tables

Table

List of Appendices

Appendix

List of Abbreviations

sUAS small unmanned aircraft systems

MWSNs mobile wireless sensor networks

DARPA The Defense Advanced Research Projects Agency

UAVs unmanned aerial vehicles

AERCam autonomous extravehicular robotic camera

NASA the National Aeronautics and Space Administration

ISS International Space Station

SPHERES Synchronized Position Hold Engage and Reorient Experimental Satellite

POSE position and orientation

AR augmented reality

VT vocabulary tree

MSS Mobile Servicing System

MDP Markov decision process

POMDPs partially observable Markov decision processes

DASC Distributed Aerospace Systems and Control

HRI human-robot interaction

IMU inertial measurement unit

IRB Institutional Review Board

HLP high-level processor

LLP low-level processor

ROS Robot Operating System

EM expectation maximization

GMM Gaussian mixture model

PID proportional-integral-derivative

FOV field of view

SUS System Usability Scale

TLX Task Load Index

LiDAR light detection and ranging

BCI brain computer interface

EEG electroencephalogram

JPL the Jet Propulsion Laboratory

SCAMP the Secondary Camera and Maneuvering Platform

Abstract

In this dissertation, the author presents a set of control, estimation, and decision making strategies to enable small unmanned aircraft systems and free-flying space robots to act as intelligent mobile wireless sensor networks. These agents are primarily tasked with gathering information from their environments in order to increase the situational awareness of both the network as well as human collaborators. This information is gathered through an abstract sensing model, a forward facing anisotropic spherical sector, which can be generalized to various sensing models through adjustment of its tuning parameters.

First, a hybrid control strategy is derived whereby a team of unmanned aerial vehicles can dynamically cover (i.e., sweep their sensing footprints through all points of a domain over time) a designated airspace. These vehicles are assumed to have finite power resources; therefore, an agent deployment and scheduling protocol is proposed that allows for agents to return periodically to a charging station while covering the environment. Rules are also prescribed with respect to *energy-aware* domain partitioning and agent waypoint selection so as to distribute the coverage load across the network with increased priority on those agents whose remaining power supply is larger. This work is extended to consider the coverage of 2D manifolds embedded in 3D space that are subject to collision by stochastic intruders. Formal guarantees are provided with respect to collision avoidance, timely convergence upon charging stations, and timely interception of intruders by friendly agents. This chapter concludes with a case study in which a human acts as a dynamic coverage supervisor, i.e., they use hand gestures so as to direct the selection of regions which ought to be surveyed by the robot.

Second, the concept of situational awareness is extended to networks consisting of humans working in close proximity with aerial or space robots. In this work, the robot acts as an assistant to a human attempting to complete a set of interdependent and spatially separated multitasking objectives. The human wears an augmented reality display and the robot must learn the human's task locations online and broadcast camera views of these tasks to the human. The locations of tasks are learned using a parallel implementation of expectation maximization of Gaussian mixture models. The selection of tasks from this

learned set is executed by a Markov Decision Process which is trained using Q-learning by the human. This method for robot task selection is compared against a supervised method in IRB approved (HUM00145810) experimental trials with 24 human subjects.

This dissertation concludes by discussing an additional case study, by the author, in Bayesian inferred path planning. In addition, open problems in dynamic coverage and human-robot interaction are discussed so as to present an avenue forward for future work.

CHAPTER 1

Introduction

1.1 Motivation

The past twenty years have seen rapid improvements in the performance of small unmanned aircraft systems (sUAS), in part through the proliferation of cheap and reliable high specific-energy lithium polymer batteries, as well as the continued miniaturization of low-cost sensors. The intersection of these technologies has given rise to a renaissance of algorithmic developments geared towards mobile wireless sensor networks (MWSNs). MWSNs are useful in a wide range of military and civilian applications including: battlefield surveillance, disaster relief, and environmental monitoring.

The Defense Advanced Research Projects Agency (DARPA) has been particularly interested in the former application since the Vietnam War. For instance, their recent development efforts have resulted in the WASP III, the first micro aerial vehicle adopted by the U.S. Armed Forces in direct support of combat operations. Soldiers on the front lines now deploy sUAS, such as WASP III, from their backpacks to supervise unmanned aerial reconnaissance and surveillance beyond the line of sight [15]. This is depicted in Fig. 1.1. Looking to the future, many authors envision teams of fully autonomous sUAS patrolling the airspace around friendly bases in search of intruding aerial and ground vehicles [16].

In the context of disaster relief, MWSNs played a pivotal role in the humanitarian response to Typhoon Haiyan. Central to this effort were large high altitude unmanned aerial vehicles (UAVs), specifically the Northrop Grumman Global Hawk, which was deployed by the U.S. Military to provide electrooptical images of damaged roads and airports. This directly informed response teams while coordinating the rescues of survivors [17, 18]. Although military efforts with the Global Hawk attracted the most media attention, the civilian use of sUAS played a role as well. One such example was a non-governmental organization named Team Rubicon who, in partnership with the private company Danoffice IT, deployed a Huginn X1 quadcopter to take aerial photos of a damaged hospital. The hospital had be-

(a) (b)

Figure 1.1: WASP platforms have already been deployed by soldiers in the field to gain situational awareness beyond the line of sight. This solider is depicted as deploying the vehicle over land [1] and sea [2].

come inaccessible by road and these photos were used assess repair material needs [19].

As stated above, sUAS have great potential to aid in environmental monitoring as well. In the context of forest fires, early detection is key to preventing the mass destruction of property and loss of life. One approach is presented in [20] where the authors develop a novel Forest Fire Detection Index whereby sUAS scanning a forest will detect the discrete event of fire through color cues determined by an onboard image processing algorithm. Although this algorithm was shown to be effective on 97% of test images, it was never demonstrated in the field. Swarms of sUAS have, however, been experimentally demonstrated in the monitoring of long-distance plant pathogen movements. A team from Virginia Tech recently conducted aerobiological sampling of spores by an autonomous network of UAVs flying synchronized trajectories through a spore plume. The authors demonstrated that coordinated simultaneous sampling by multiple autonomous UAVs within a single plume reduces sample variations in plant pathogen measurements in the lower atmosphere [21].

sUAS also offer great capability to aid in the structural inspection of bridges. Traditional bridge inspection techniques often require the construction of scaffolding and the operation of cherry pickers, cranes, etc. In addition, the necessary lane closures and traffic control measures have, at times, resulted in the deaths of inspectors [22]. The expenses and risks of these approaches can be mitigated through remote inspection via sUAS. In [23], the authors consider semi-autonomous bridge inspection by UAVs. In this work, the inspection path of the vehicle is entirely pre-planned by the human supervisor as a set of GPS waypoints. An onboard low level controller is used to guide the vehicle along this path while it records video images of the structure for post flight analysis. The authors of [24] consider a Traveling Salesman approach to sequencing an unordered set of waypoints for the generation of time optimal bridge inspection trajectories. Their work places specific

emphasis on path planning in the face of a variety of wind disturbances. In [3], the authors substantially reduce the number of required waypoints by partitioning a bridge into a set of surfaces with entry and exit waypoints. The surfaces are observed via a local coverage controller. This partitioning is depicted in Fig. 1.2. In all of these works, the environment is highly structured and substantial planning is required offline in order to select points of visual interest or surface partitioning schemes. These approaches are unable to react in real time to their observations.

Figure 1.2: The authors of [3] reduce the number of waypoints required for bridge structural inspection by partitioning the bridge into a set of surfaces with entry and exit nodes. The high level sequencing of surfaces is accomplished with a Traveling Salesman approach while the low level observation of surfaces is accomplished with a local coverage controller.

The common theme in all of these sUAS applications is that of sustaining situational awareness, i.e., "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" [25]. This theme has surfaced repeatedly and has become the central tenet of this dissertation. Although the examples cited until this point have been terrestrial and aerial, many parties are interested in situational awareness in outer space as well.

Situational awareness in space dates back to at least 1957 when the U.S. military began to actively track artificial satellites, i.e., Sputnik 1. As of 2007, the Space Surveillance Network tracks over 10,000 objects using a network of ground-based electro-optical and

radar sensors distributed around the world (see Fig. 1.3). Operators of this military network routinely inform the National Aeronautics and Space Administration (NASA) of imminent threats to their assets including the ISS [4]. Unfortunately, the vast majority of orbital debris and micrometeoroid particles are beyond the detection capabilities of this network. Such objects pose a persistent threat to space assets. The significance of these threats became apparent during a number of space shuttle flights between 1984 and 1990. Most notably, the Long Duration Exposure Facility spent 69 months in low Earth orbit during this period and was found to have incurred over 35,000 surface impacts that were greater than 0.5 mm. This experiment drew greater attention to the importance of routine exterior structural inspection during long duration crewed spaceflight missions [26].

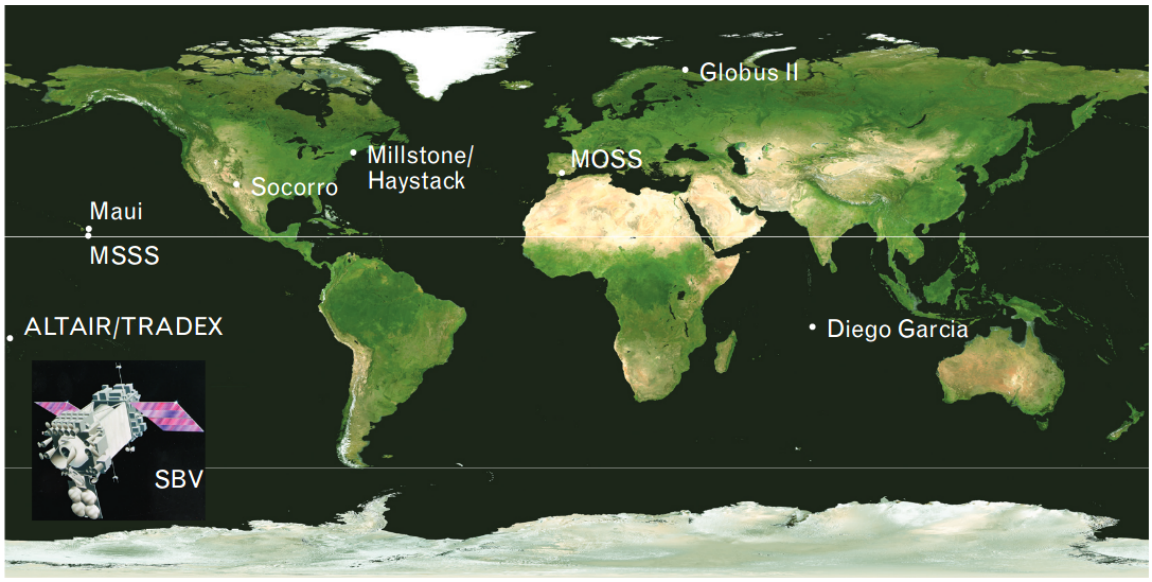
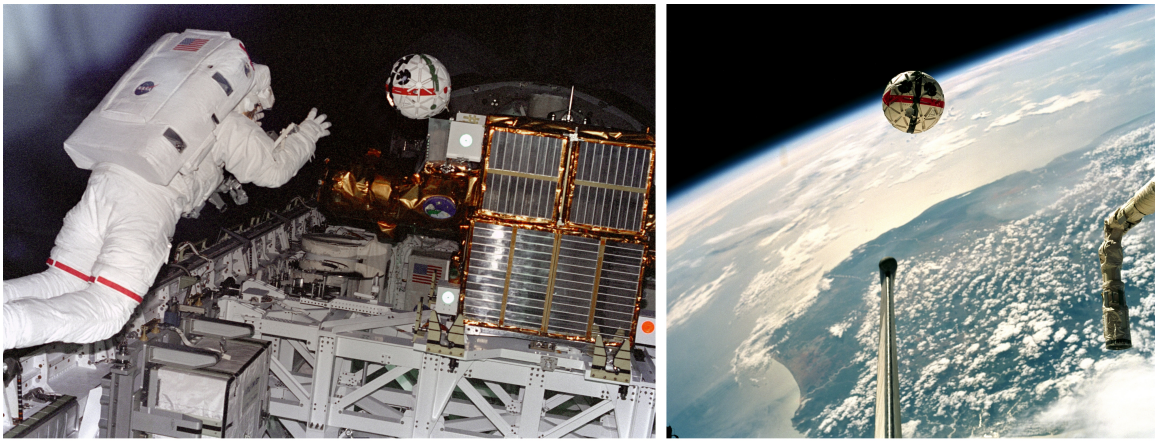


Figure 1.3: The Space Surveillance Network consists of ground-based electro-optical and radar sensors which are operated by the U.S. Army, Navy, and Air Force at 25 sites worldwide [4].

Throughout the 1990's, NASA worked on a number of experimental platforms to enable robotic inspection of the conceptual Space Station Freedom (which later evolved into the ISS). Among the first of these was the Remote Surface Inspection System developed by engineers at the Jet Propulsion Laboratory (JPL). This was essentially a robotic manipulator with a camera attached to the end which was teleoperated to view a 1/3 scale mockup of the Space Station Freedom truss. Engineers at the time proposed that this system could eventually be used in a fully autonomous mode in which automatic scans of the truss could be compared in flight against a reference scan to detect anomalies via computer vision [27]. Though the autonomous mode was never implemented in flight, this project was a precur-

sor to the type of inspection that is now carried out using cameras attached to the Mobile Servicing System (MSS) (a large robotic platform containing manipulator arms on board ISS).

Aside from manipulator arms, NASA has also considered the use of a special class of MWSNs known as free-flying space co-robots—henceforth referred to as free flyers. The first demonstration of this technology was the autonomous extravehicular robotic camera (AERCam) Sprint which was remotely piloted outside of the Space Shuttle during the STS-87 mission in December of 1997. During the demonstration, the 35 pound vehicle flew alongside a spacewalking astronaut and was manually controlled by another astronaut inside of the Orbiter. Although the pilot maintained constant visual contact with the AERCam through the Shuttle windows, he also viewed a real-time video stream provided by the robot to verify the feasibility of free flyer inspection via remote viewing [5]. Photos of this demonstration are presented in Fig. 1.4.



(a) (b)

Figure 1.4: In December of 1997 the AERCam was released from the Space Shuttle Payload Bay by astronaut Winston Scott (a) to fly away from the spacecraft. It is the first and only free flyer to demonstrate extravehicular capabilities [5].

In the decade following this demonstration, NASA worked to develop a miniaturized successor known as Mini AERCam and envisioned its use in supervised autonomous inspection of spacecraft exteriors [5]. This project unfortunately never made it off the ground with NASA shifting their focus towards intravehicular free flyers. Thus, the primary tools presently available for structural inspection aboard ISS are a set of standard definition television cameras fixed to the exterior structure. These cameras have substantial gaps in coverage as illustrated in Fig. 1.5. As stated above, there are additional cameras attached to the MSS; however, inspection with this system requires substantial time and planning as every joint angle must be specifically planned out and commanded by a human [6]. The

current state of structural inspection in space is far from autonomous.

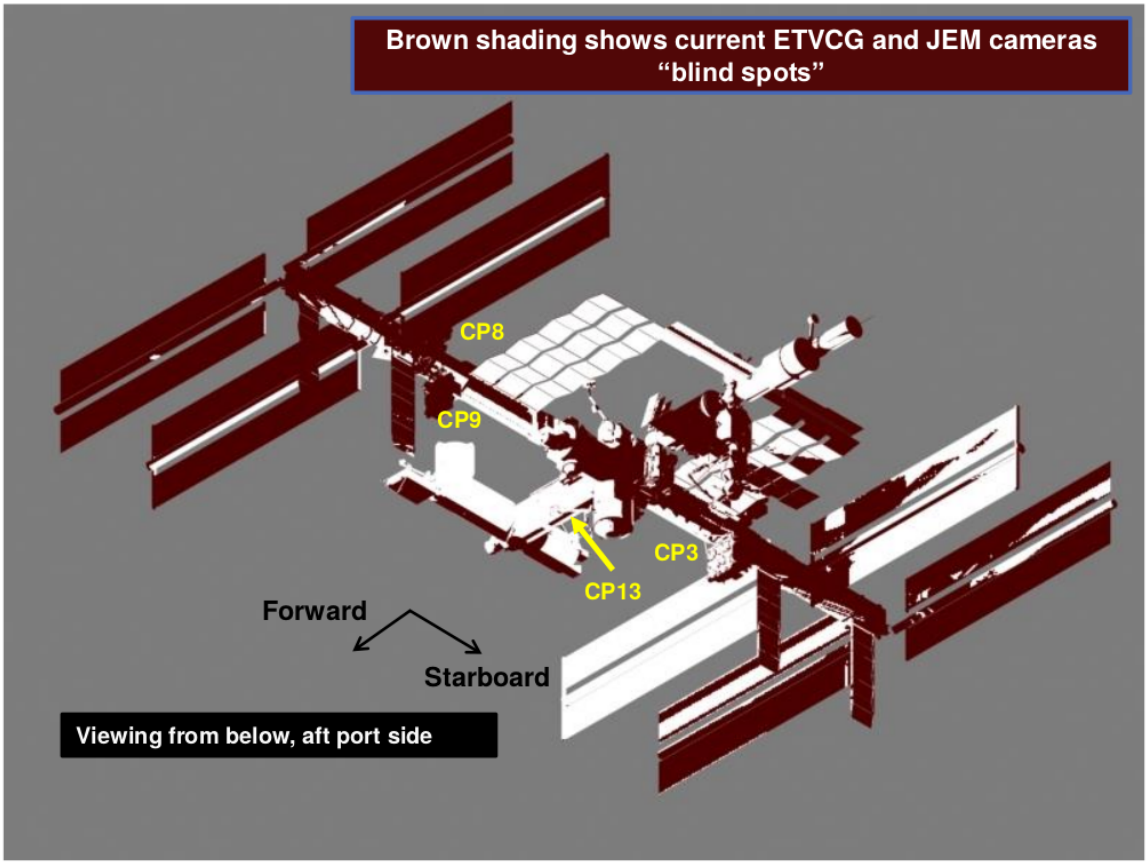


Figure 1.5: Standard definition television cameras mounted on the exterior of the ISS have substantial gaps in coverage [6].

Recall that NASA's recent free flyer projects have primarily been intravehicular. The first of these, the Synchronized Position Hold Engage and Reorient Experimental Satellite (SPHERES), was delivered to the ISS in 2006 and was developed in an effort to sustain situational awareness inside of the cabin. This platform is depicted in Fig. 1.6. The vehicles use cold-gas thrusters and an external ultrasonic local positioning system for the control and estimation of position and orientation (POSE) respectively. Through the Smart SPHERES project, NASA sought to test the feasibility of using free flyers, teleoperated by ground control, to conduct routine environmental surveys, e.g., sound and radiation levels, of the cabin. These surveys are normally gathered by astronauts at the expense of their most precious resource, i.e, time. Between 2011 and 2012, operators in Mission Control conducted test surveys of the ISS using a combination of direct manual control and supervisory waypoint commands. The primary lessons learned from these experiments were the need for increased flight time, autonomous recharging, and onboard localization [28]. All

three of these limitations have been lifted in the successor program Astrobee which arrived on the ISS in April of 2019 [29]. Astrobee is capable of autonomous rendezvous and docking with its recharging station, uses impellers for propulsion, and runs localization onboard via fusion of sparse map, optical flow, and inertial measurement unit (IMU) data [30].

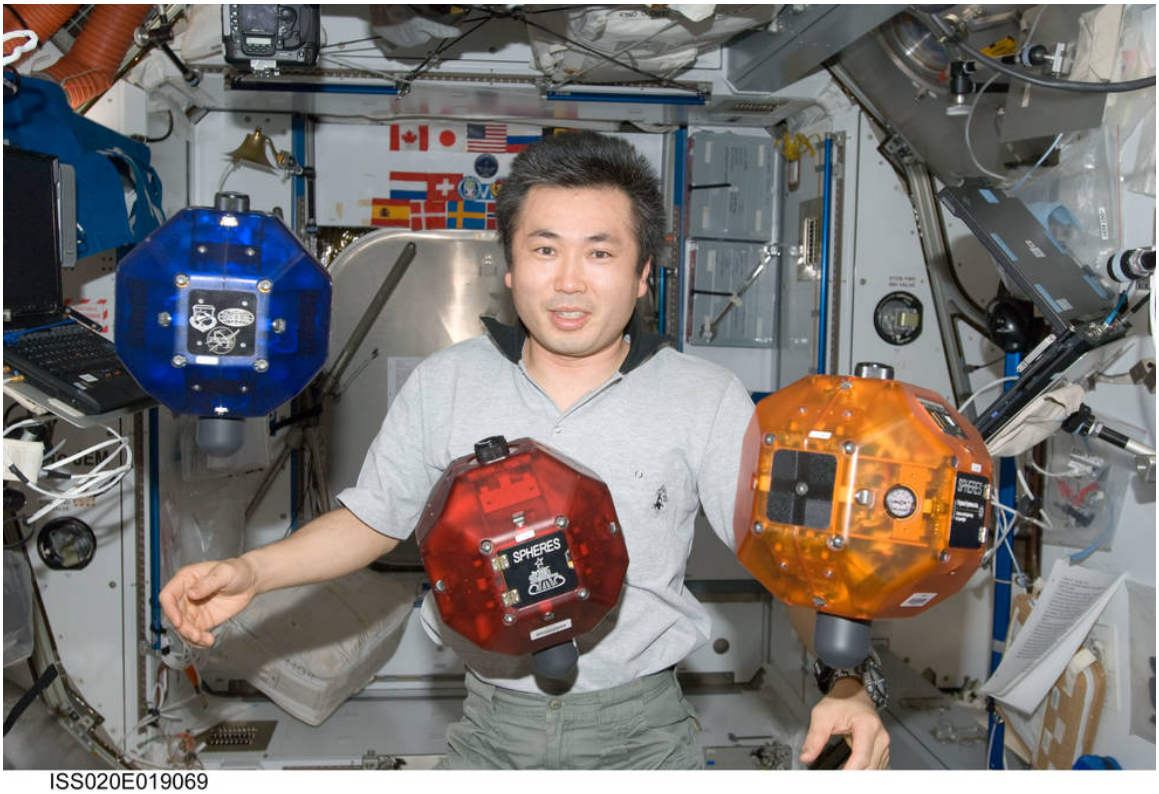


Figure 1.6: Three SPHERES were launched to the ISS in 2006 and have subsequently had their computational and sensing capabilities upgraded through the Smart SPHERES project. Their appearance was inspired by the lightsaber training droid in *Star Wars* [7].

While Astrobee will greatly enhance situational awareness inside of ISS, it is unfortunate that extravehicular free flyer developments halted after the flight of the AERCam Sprint—a recent mission comes to mind in which they would have been particularly valuable. On the eighth day of the STS-120 mission, the deployment of a newly installed solar panel array aboard the ISS was abruptly halted when astronauts observed a tear between two panels via the MSS. With the station now underpowered and unfit to withstand external loads, e.g., docking and undocking, crew and ground controllers worked tirelessly for over 72 hours to plan and execute a spacewalk to repair the array. The improvised procedure was greatly dependent on MSS to transport an astronaut to the damage site, and then downlink external video of the repair to those assisting on the ground [31]. One can argue that these repairs could have been expedited had free flyers been available to extend the range of ac-

cessible camera views and to gather visual information more rapidly than possible through the MSS. Furthermore, the successes of STS-120 and MSS would not have translated to incidents with faster dynamics such as that of the Progress M-34 collision with space station Mir that resulted in the loss of the Spektr module in 1997 [32]. Let these incidents illustrate the importance of effective human-robot collaboration, in part through assistive camera views, in sustaining situational awareness during spaceflight.

As NASA begins to integrate augmented reality (AR) into spacesuit helmets [33], [34] robotic cameras will be able to directly inform spacewalking astronauts without the need for real-time ground communication—a necessity for deep space missions beyond the Earth-Moon system. These missions will substantially benefit from some level of autonomy in the generation of assistive camera views as the aforementioned contemporary camera systems depend upon teleoperation—a task that is less than trivial when considering the difficulty in establishing a common frame of reference during space walks. Astronauts routinely change their frame of reference when communicating with ground controllers—often multiple times in a single sentence [35], [36]. A simple request such as "pan the camera up" can become incredibly difficult in an environment where directions are somewhat obscure. Thus, a primary motivation for free flyer control is to develop autonomous algorithms by which free flyers can learn and stream assistive camera views to astronauts without the need for constant verbal feedback.

The motivation for this dissertation at large is to develop a series of hybrid control strategies which, having been deployed to sUAS and free flyers, will increase the situational awareness of both autonomous robot teams as well as human-robot collaborative teams across the domains of air and space. In the latter team structure, the collaboration is facilitated primarily via AR. The theoretical contributions of this work are presented in as general a manner as possible so as to maximize the range of problems to which they may be applied. In much of the contemporary controls literature, teams of autonomous mobile sensors tasked with surveying and patrolling a domain are typically referred to as dynamic coverage networks. Thus, much of the literature review presented in the following section will be devoted to that topic. The remainder of the literature review will be devoted to contemporary work on human-robot interaction (HRI) and AR.

1.2 Literature Review

1.2.1 Multi-agent Coverage

At the highest level, coverage refers to the problem of deploying a sensor network within a domain such as to guarantee some quality of surveillance throughout the environment. Typical measures of this quality include the proportion of the environment that is presently or has previously been sensed at any given time as well as the efficacy of the network for the detection of intruders [37]. Coverage is often partitioned into three classes of problems: static, dynamic, and persistent.

Numerous authors have addressed 2D static coverage problems [8, 38]. These are often referred to as area coverage, k-coverage, or point coverage and the solutions typically concern directing the agents towards the centroids of Voronoi tesselations [39,40] (see Fig. 1.7. A common theme among these variations of the static coverage problem is that each agent seeks to converge upon a fixed position in space to yield satisfactory steady-state coverage.

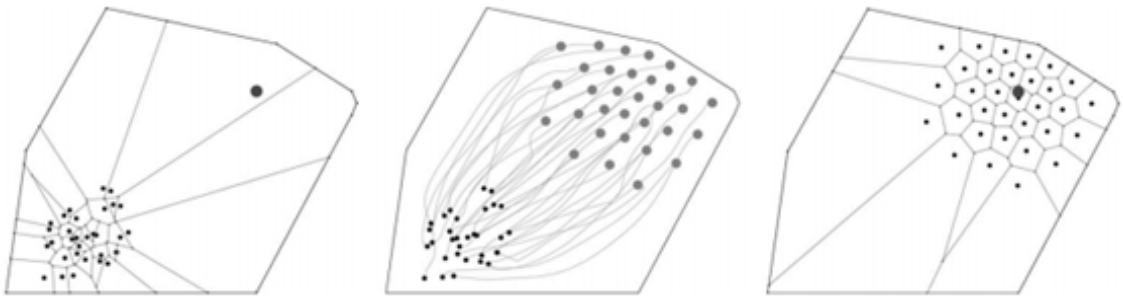


Figure 1.7: Early problems in static coverage were solved by directing sensing agents to the centroids of Voronoi tesselations. The image above is taken from [8].

Static coverage of 3D environments is presented in [41], where the authors extend the classic problems of node scheduling and area coverage to a rough terrain modeled after Encanto Park, Phoenix. Furthermore, they present a novel airdrop-based deployment strategy designed around potential aircraft flight paths. Additional examples of 3D static coverage occur in [42] and [43]. The authors of [42] derive the optimal configurations of Pan-Tilt-Zoom cameras to maximize coverage, while the authors of [43] utilize isotropic conical sensing models and maximize the network's power lifespan subject to minimum coverage rate constraints.

Dynamic coverage problems involve the active exploration of a domain. The agents are tasked to move ("sweep") their sensing footprints over all points until some desired level of

coverage has been achieved [9, 44, 45]. Illustrations of dynamic coverage are presented in Fig. 1.8 and Fig. 1.9. Persistent coverage is more or less a subset of dynamic coverage that considers information decay within the environment: i.e., agents are required to continually return to areas of interest.

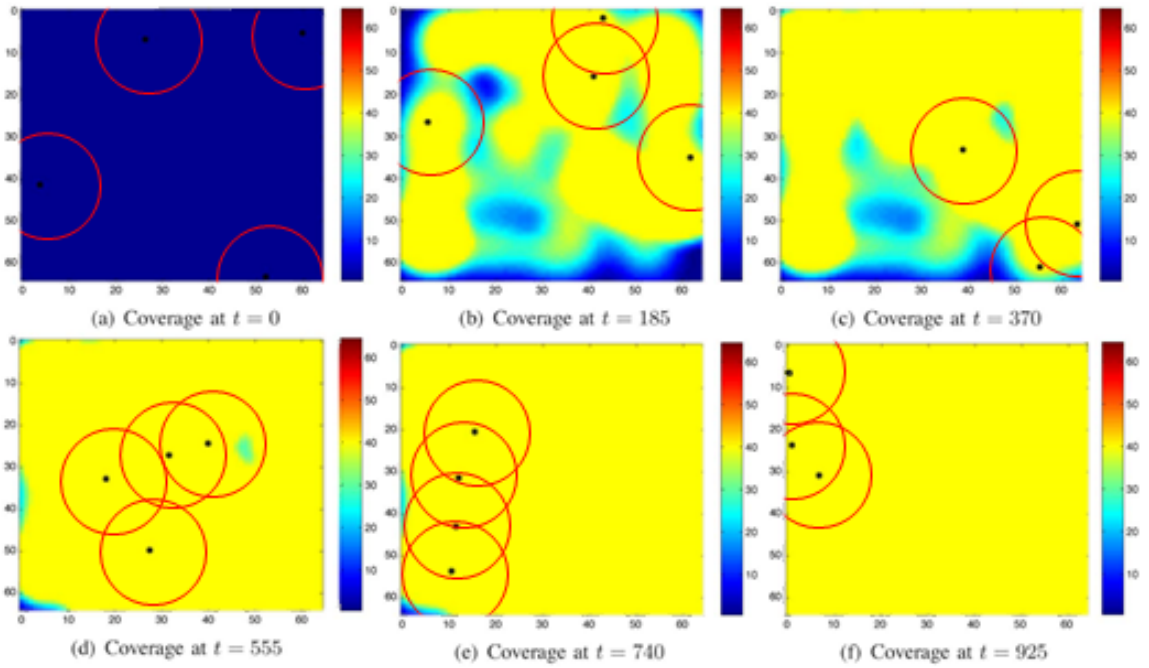


Figure 1.8: A 2D dynamic coverage scenario is illustrated above. A team of four agents are dispatched to sweep their sensing footprints across a domain until it reaches a satisfactory level of coverage. The range of the sensing footprints are indicated by red circles. Satisfactory coverage is indicated by yellow while zero coverage is indicated by blue. The image above is taken from [9]. Note that the red circles were added by the author of this dissertation in the interest of clarity.

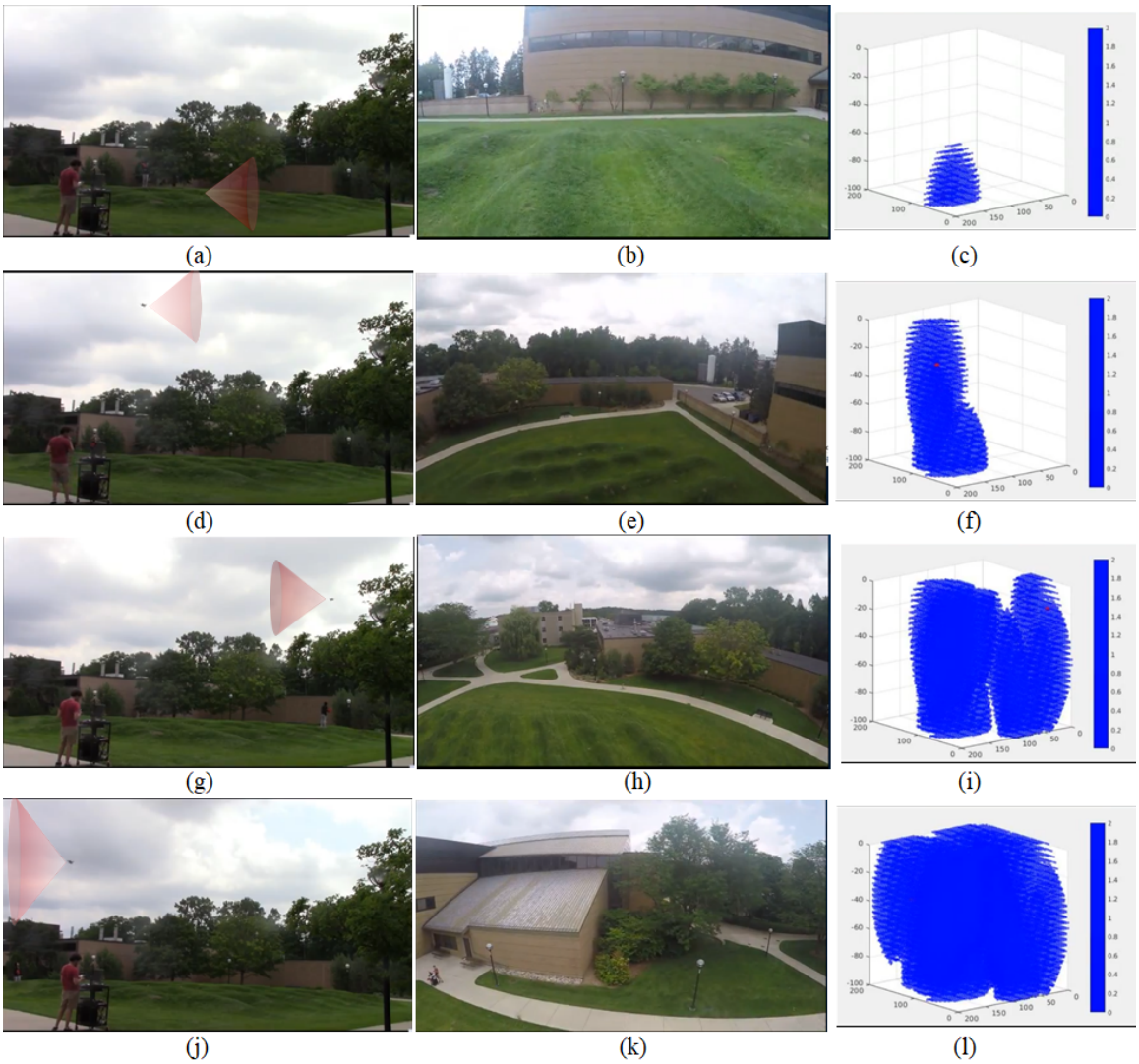


Figure 1.9: Illustrated above: a demonstration of 3D dynamic coverage conducted by the author at the University of Michigan. A UAV is dispatched to sweep its sensing footprint across the airspace above the Wave Field until all points within this airspace are covered up to some satisfactory level. The four rows correspond to unique instances in time. The left column depicts a fixed exterior view and includes an illustration of the sensing footprint. The center column depicts the UAV's camera view which is approximately that of the sensing footprint. The right column depicts the coverage level of the domain as it increases in time. A video of this demonstration is available online: https://www.youtube.com/watch?v=uZI4cfDnLGc&feature=youtu.be.

One of the earliest works in persistent coverage was [10], where agents were required to cover all points in a 2-D convex polygonal domain every $T^\star$ time units. This was accomplished with the design of concentric polygonal trajectories with agents following closed paths in steady state. An illustration of this is presented in Fig. 1.10. The work in [46] is

similar but also introduces a linear coverage decay rate for specific points of interest. In that work, as well as [47], controller design is akin to regulating the velocity along paths generated offline to increase observation time at select points of interest. As the decay rates are known and time invariant, optimal speed control is computed via linear programming.

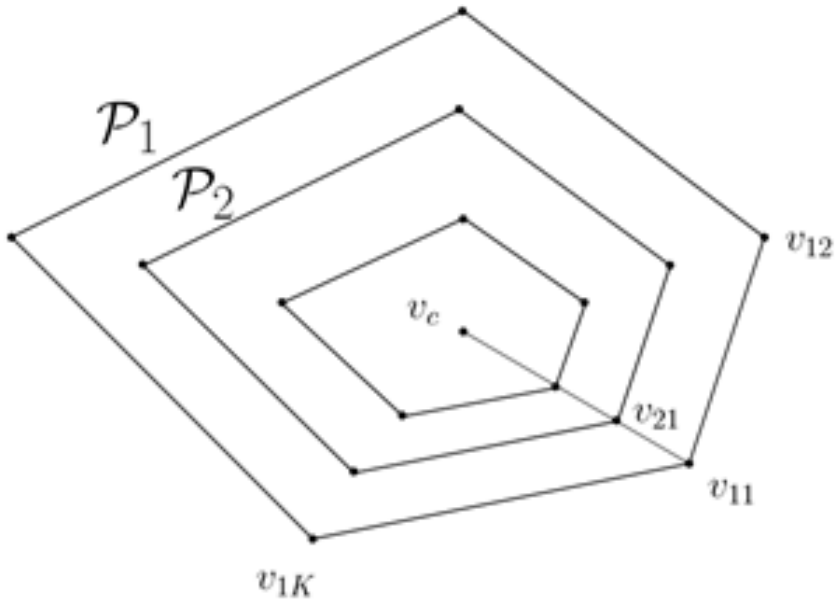


Figure 1.10: In this simple persistent coverage protocol, agents follow closed concentric polygonal trajectories so as to repeatedly service all points in the domain. This image is taken from [10].

Palacios-Gasós et. al have developed further results on persistent coverage [48] that build upon [47]. While the earlier work assumed both the existence and knowledge of an optimal path to cover all points of interest, [48] uses techniques from discrete optimization and linear programming to iteratively compute this path. The effect is that if the coverage decay rate of a specific point of interest is found to be insufficient to justify the transit time required to service it, then the point may be removed online from the path of the robot. Prior works, i.e., [46, 47], would have instead driven the robot to quickly pass through the point.

In [11] and [49], the desired coverage level of the domain is maintained with density maps that yield additional observation time at select areas of interest. In [49] the maps are time-invariant, while [11] considers time-varying density maps that may be designed around moving points of interest (e.g., aerial surveillance targets). However, the latter work only uses density maps in the derivation of control laws and not in the differential equations governing the coverage level evolution. There is no memory of the target, or of its effect on the environment, encoded into the coverage level; therefore, any implications on the

environment due to the presence of the target are not considered after the target exits the domain. This is illustrated in Fig. 1.11.

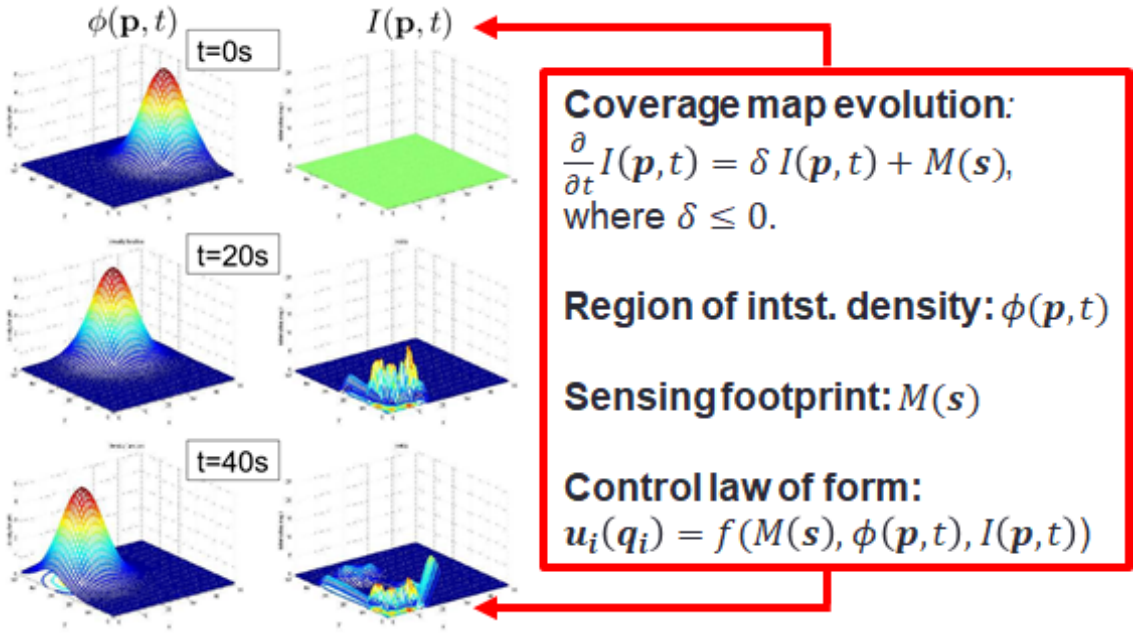


Figure 1.11: In this persistent coverage protocol, a target flies in circular trajectories (left column) with it's location coincident with the mean of an associated region of interest density map. The control laws are functions of both the coverage level as well as the target's density map. The coverage level is independent of the target's density map and thus the system is memoryless with respect to targets. This image is taken from [11].

Common assumptions among all of these persistent coverage approaches include convex 2-D domains, predictable environments, and simplified sensing and dynamic models for agents. Persistent coverage of surfaces embedded in $\mathbb{R}^3$ is considered in [50]; however, this work is closer to that of [10] in that agents also follow preplanned trajectories without considering spatially-dependent coverage decay maps.

Coverage of stochastic environments is presented in [12, 51]. In [12], the authors consider that agents must observe events at multiple points of interest, and that the precise arrival times of events are unknown *a priori*. Arrival time statistics are used to inform a multi-objective scheduling protocol that results in fixed cyclic servicing policies. This is illustrated in Fig. 1.12. In [51], the environment contains smart intruders that actively attempt to evade a camera surveillance network. Camera motion is restricted to a single pan axis and thus the system model is essentially that of a 1-D pursuit evasion problem.

Until very recently, approaches to the dynamic coverage problem typically did not consider the power constraints of the vehicles nor attempt to develop any protocols for vehicle

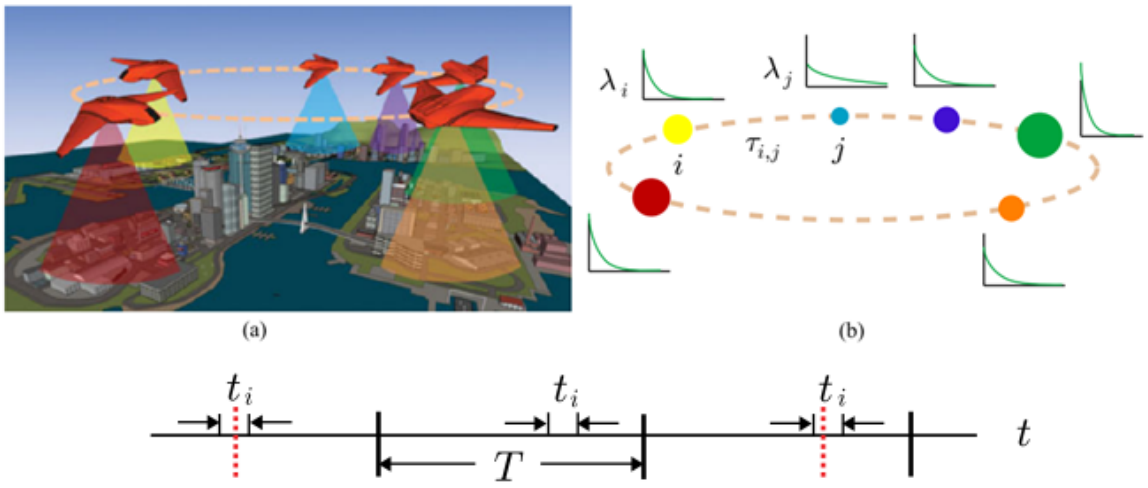


Figure 1.12: In this persistent coverage protocol, agents follow cyclic trajectories (a) to observe events arriving at random times at fixed locations. The relative event frequency is illustrated by disc size in (b). The authors optimize the waiting time at specific stations (c) so as to maximize the likelihood of observing all events. This image is taken from [12].

servicing and redeployment. A number of authors have begun to explore this theme of *energy-awareness*.

The authors of [52] present experiments validating an energy-aware surveillance protocol for multiple quadrotors. Trajectories are generated by dividing the domain into a grid of cubes, each of which contains a vector field. The field may flow entirely inward to a cube thus directing the agent to hover or it may flow outward through one surface to direct the agent to an adjacent cube. This work only allows for one quadrotor to fly at a time and the agents are provided three separate charging stations. This work does not model the sensing capability of the agents and surveillance is described in terms of persistently routing the agents between a small set of regions of interest.

In [53] and [54], the authors consider dynamic coverage by agents with homogeneous and fixed battery lifespans. To continue covering, the agents must periodically direct themselves to charging stations. The coverage task is to service a set of discrete points of interest within a 2-D plane. Neither paper presents realistic models of agent kinematics, sensing, or control. [53] considers a centralized charging station and attempts to determine the minimum number of agents and their associate paths that will allow for a sufficient number of battery swaps while providing adequate service. [54] considers a set of distributed charging stations and provides analysis on the suboptimal Hamiltonian paths between target points.

With few exceptions, the majority of contemporary literature on coverage makes the following assumptions: 2D convex domains, omnidirectional sensing models, and unlim-

ited power resources. In this work, the dynamic and persistent coverage protocols: are defined for complex 3D environments containing stochastic intruders (whose estimated trajectories, unlike in [11], are recorded in the coverage level), collect data from anisotropic sensors of limited field-of-view, and prescribe energy-aware deployment, scheduling, and domain partitioning. These protocols enable situational awareness in both autonomous and human-robot collaborative teams across the domains of air and space. The efficacy of interaction in human-robot networks can be enhanced in part through AR and accurate inference of human intent. Thus, a literature review of these related topics is the subject of the following subsection.

1.2.2 Human-Robot Interaction and Augmented Reality

Much of the existing literature on HRI has focused on human supervisory control, i.e., an operator is required to process information provided by the robot, usually through a computer interface, and then direct the robot's behavior [55]. In these systems, the human is often required to possess detailed knowledge of the robot's operating modes and transition characteristics in order to properly predict future actions. Automation surprises, due to human fatigue or lack of mode awareness, may have potentially fatal consequences [56,57].

Human-robot networks of the future need to operate in a manner that is both intuitive and natural for the human, i.e., free of the information transfer overhead present in supervisory protocols. This will enable a truly collaborative "peer-to-peer" network where humans and robots can operate independently of one another to service various sets of tasks within a given domain [35]. Central to this idea is the requirement that both the robot and the human must be capable of inferring one another's intentions. Human inference of the robot's intent has been studied at length in [13,58–60], where a typical solution is to design legible motion planning, i.e. visually communicative robot trajectories that are arced or exaggerated to aid in human perception. These approaches are based on tenets of film and computer animation and are thus predictable to humans. User studies have demonstrated that humans are better able to predict the future path of a robot while it follows an exaggerated trajectory than if it were to follow a purely functional one [13]. This concept is illustrated in Fig. 1.13.

In [61], the authors consider how visually displaying an autonomous robot's intent, i.e., waypoints and trajectories, can inform a human teammate in servicing spatially distributed tasks. This group also studied AR in the context of providing UAV pilots engaged in aerial inspection tasks with a visual display of the robot's current sensing field [62]. In these works, experimental results are presented through statistical analysis of both objective task

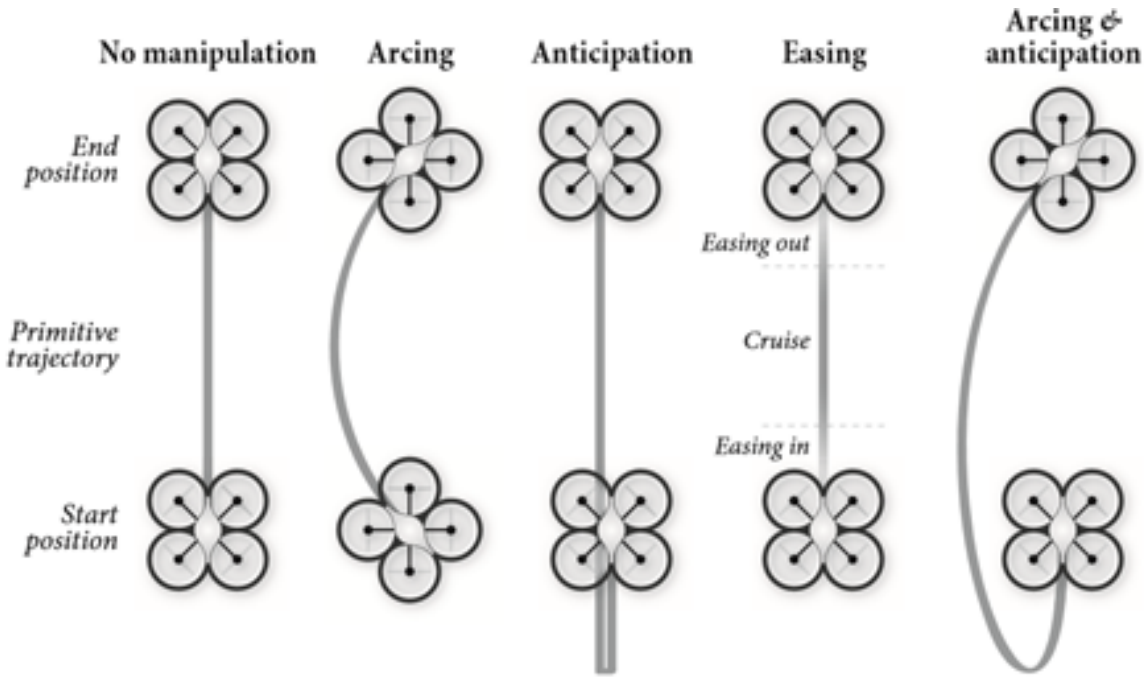


Figure 1.13: Many authors have studied methods for the effective communication of intent by the robot to the human. One strategy is to design exaggerated trajectories which can help the human to predict the path of the robot. The techniques illustrated above, based upon film and computer animation principles, have been demonstrated in user studies to aid humans in the prediction of animated bouncing ball trajectories. This image is taken from [13].

performance metrics as well as survey-based subjective measurements of user satisfaction.

In [63], the authors study the problem of a human and a robotic manipulator completing independent tasks in a closely shared workspace. The authors demonstrate that human productivity is only minimally reduced so long as the manipulator is proactive in not obstructing the human's present task at any given time. This work does not address the inference of human intent and instead provides ideal measurements to the robot.

The robot's inference of human intent has been approached using tools from Bayesian statistics [64–68] in the context of single-task collaboration. In [64], the authors consider a human and robot collaborating within a factory setting on the construction of a cart. The robot is required to fetch, return, and use various tools to assemble the cart at a workbench while the human stocks cart components. The authors model the workbench state as a first-order Markov process, i.e., each state is a conjunction of the partially assembled cart along with the component that shall be attached. This state can be mapped to a unique tool which the robot ought to fetch and use. The authors demonstrate experimentally that Bayesian inference of the next Markov state (i.e., the next cart component selected by the human)

offers a more fluid and natural assembly process than strictly reactive trials. Essentially, the robot is able to predict which tool it ought to retrieve before the human returns with the next component thus expediting the assembly process. In [65], the authors use the related technique, Intention-Driven Dynamics, in games such as table tennis. This approach is an extension of Gaussian process dynamical models that incorporates human intention (e.g., the intended return point of table tennis ball) into the estimation of latent variable transition functions.

Visual gaze and head tracking data can be useful in the communication of human intent to free flyers (e.g., Astrobee) that NASA is actively developing, in part, to act as intelligent mobile cameras. Online learning using such data, as well as first-person video streams, are studied in [69–71]. In [69], the authors present an assistive co-robot that can aid quadriplegic individuals by retrieving desired objects. Gaze information directs the robot towards a search area where object recognition is achieved via a vocabulary tree (VT) based recognition algorithm [72]. In [70] and [71], the authors record egocentric video streams from humans participating in daily activities, e.g., preparing coffee, during a learning stage. Their algorithm detects task relevant objects and will display video clips of proper usage, obtained during the learning stage, to the headsets of future users interacting with the same objects. Instructional videos displayed via AR are also considered in [73].

The concept of using an aerial/space co-robot to extend a human's visual field of view (FOV) is explored in [74], where the authors conducted a small-scale user study on aerial self-observation during outdoor maze navigation. Drone-augmented human vision is also studied in [75], [76], [77], and [14]. In [75], aerial images of a construction site are captured from a static location and fused with holographic elements to aid in site management. Semi-autonomous structural inspection is considered in [76] where human head motions are used to perturb the UAV's automated reference trajectory online. In [14], occluded regions of interest are made visible to users via virtual x-ray vision (see also [78]). This is illustrated in Fig. 1.14. In these studies, UAVs require supervision from their human teammates and must be controlled via gesture, gaze, or joystick. The process becomes inefficient in that the humans primarily focus upon their interaction with the drone. It is therefore beneficial to the process efficacy if the robotic assistive camera has increased levels of autonomy. The agent (i.e., robotic camera) should be capable not only of learning sets of task-relevant views but also the temporal context of these views, i.e., where and when to observe. This will result directly from some model of human task sequences, e.g., captured via a MDP.

MDPs have been used to model human task sequences in shared environments in [79], [80], and [81]. In [79], partially observable Markov decision processes (POMDPs) are used to model the probability that any one task, from a finite task set, is the current intended task

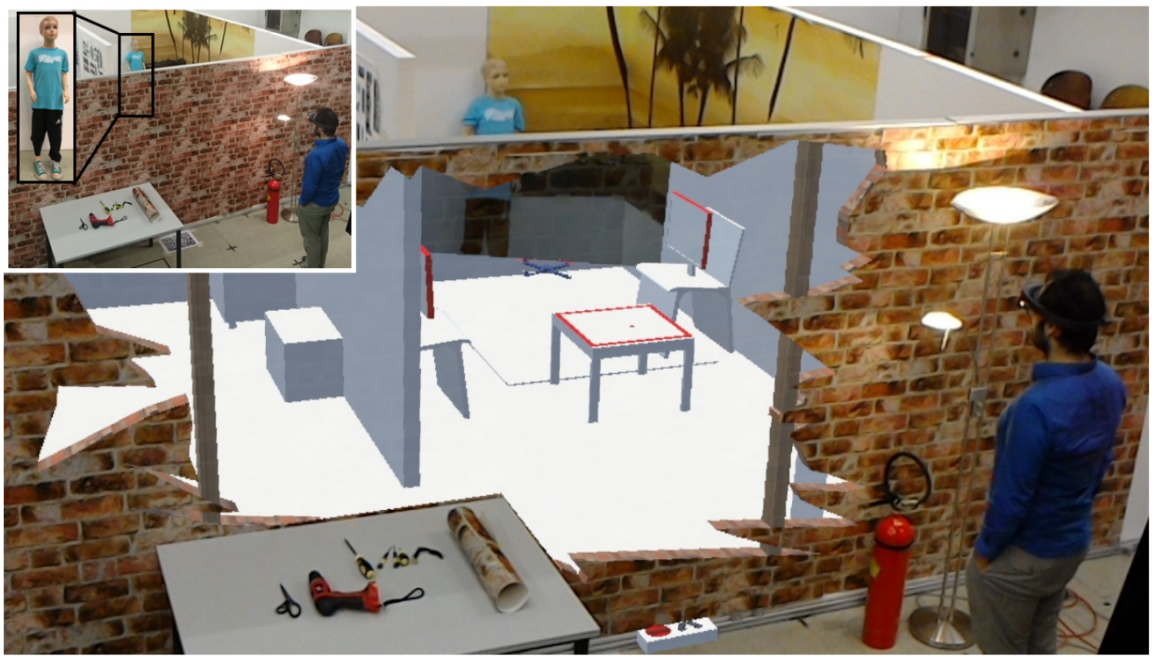


Figure 1.14: In [14], a human was tasked with piloting a UAV, via gaze inputs, through an occluded environment. This was done by projecting exocentric views of the environment onto the obstructing wall with AR. Thus, the human was given virtual x-ray vision.

of the human. These "robot belief" states are updated through human action observations. However, the number of states quickly explode and the benefits of this approach are limited for highly trained individuals in structured task environments. Such environments, specifically the International Space Station, are the focus of [80], and [81]. The authors of [81] assume that human tasks are known *a priori* and propose that robot actions should not interfere with the human. This dissertation (chapter 3) adopts a similar context in that both agents maintain a safe proximity and the robot allows for the human to choose at any given time whether they prefer to interact with it directly or to instead execute their tasks independently of the robot.

The primary contribution of this dissertation, with respect to HRI, is a set of algorithms that enable an assistive aerial robotic camera to extend the FOV of a multitasking human, via AR, to include additional task-relevant views of the environment. The robot must learn the locations and proper sequence of visual interest regions within the multitasking environment, and broadcast camera views of these regions in real time to a human collaborator wearing an AR headset. The camera views and appropriate sequencing are learned from a combination of visual gaze tracking, point cloud data, and MDP training cues. This form of collaboration is designed to improve the situational awareness of the human during multitasking.

1.3 Contributions

The following list is a summary of the contributions of this research:

- formal hybrid control strategies for 3D multi-agent dynamic coverage networks subject to anisotropic sensing and rigid body kinematics,
- energy-aware deployment and scheduling of MWSNs, which must periodically return to refueling stations, with formal guarantees provided for agent safety,
- techniques for the detection and trajectory estimation of stochastic intruders with formal guarantees provided for interception by friendly agents,
- a hybrid model for human supervisory dynamic coverage via gesture-based commands,
- an unsupervised learning algorithm by which aerial co-robots and free flyers can estimate sets of task-relevant camera views and then provide such views to their human partner,
- a MDP-based supervised learning algorithm by which a human can reinforce the sequential selection of views from the aforementioned set.

Several concepts occur appear repeatedly throughout this dissertation and their connections to major contributions are illustrated in Fig. 1.15.

1.4 Relevant Publications

This dissertation was composed, in part, from research published in the following prior works:

- W. Bentz and D. Panagou "An Energy-aware Redistribution Method for Multi-Agent Dynamic Coverage Networks", 55th IEEE Conference on Decision and Control, Las Vegas, USA, December 2016.
- W. Bentz and D. Panagou "3D Dynamic Coverage and Avoidance Control in Power-constrained UAV Surveillance Networks", 2017 International Conference on Unmanned Aircraft Systems, Miami, FL, USA, June 2017.
- W. Bentz, T. Hoang, E. Bayasgalan, and D. Panagou "Complete 3-D Dynamic Coverage in Energy-constrained Multi-UAV Sensor Networks", Autonomous Robots, April 2018, Volume 42, Issue 4, pp 825-851.

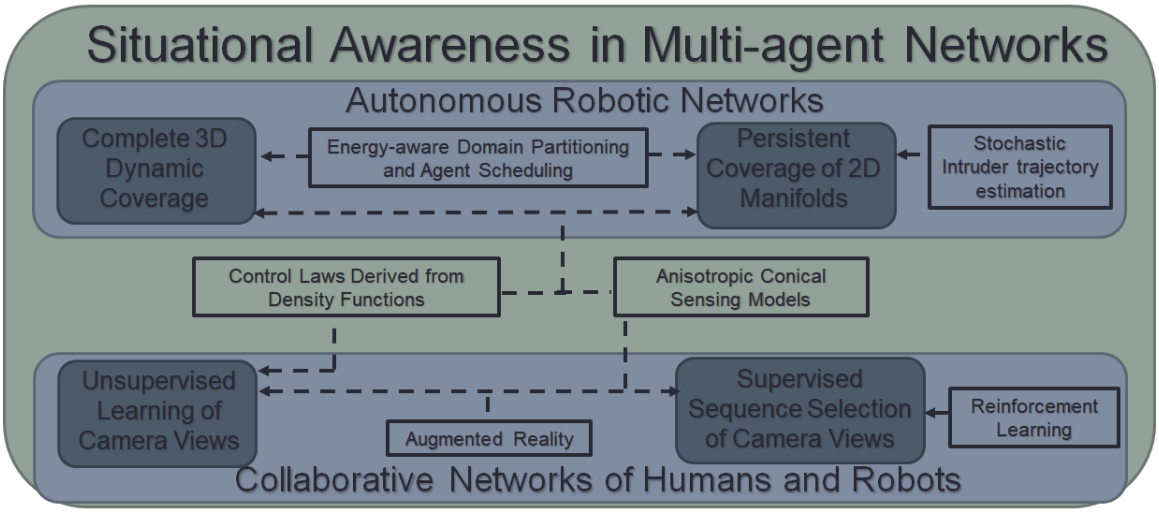


Figure 1.15: Several concepts, including sensing models and vehicle guidance along density functions, occur repeatedly throughout this dissertation. Their connections to the major contributions are illustrated above.

- W. Bentz and D. Panagou "Persistent Coverage of a Two-dimensional Manifold Subject to Time-varying Disturbances", 56th IEEE Conf. on Decision and Control, Melbourne, Australia, December 2017.
- W. Bentz and D. Panagou "Energy-aware Persistent Coverage and Intruder Interception in 3D Dynamic Environments", 2018 American Control Conference, Milwaukee, Wisconsin, June 2018.
- W. Bentz and D. Panagou "A Hybrid Approach to Persistent Coverage in Stochastic Environments", Automatica, November 2019, Volume 109, doi:10.1016/j.automatica.2019.108554.
- W. Bentz, S. Dhanjal and D. Panagou "Unsupervised Learning of Assistive Camera Views by an Aerial Co-robot in Augmented Reality Multitasking Environments", 2019 IEEE International Conference on Robotics and Automation, Montreal, Canada, May 2019.
- W. Bentz, L. Qian, and D. Panagou "Online Learning of Assistive Dynamic Camera Views by an Aerial Co-robot in Augmented Reality Multitasking Environments", submitted to the International Journal of Robotics Research, under review.

1.5 Outline

Chapter 2 includes the contributions of this dissertation to power-constrained dynamic coverage. It presents a hybrid control strategy for UAVs that guarantees complete coverage of 3D environments. This work was published in *Autonomous Robots* in 2017 (see section 1.4). This initial work lays the path for a treatment of persistent coverage in substantially more complex environment. Agents are now tasked with covering a 2D manifold embedded in $\mathbb{R}^3$ that is subject to impact by stochastically generated intruders. This work was published in *Automatica* in 2019 (see section 1.4). Chapter 2 concludes with a case study on gesture-based coverage where a human acts as a dynamic coverage supervisor to a semi-autonomous agents. This study serves to transition the work away from fully autonomous networks and towards human-robot collaborative networks that seek to sustain situational awareness in the human teammate.

Chapter 3 is devoted entirely to human-aerial/space robot collaboration. The chapter describes research into online learning and selection of relevant camera views within multitasking environments. The results of this work are data driven and derived from an Institutional Review Board (IRB) approved (HUM00145810) human experiment on subjects within the Distributed Aerospace Systems and Control (DASC) laboratory. The cumulation of this work was submitted to the *The International Journal of Robotics Research* in January of 2020 and is currently under review.

Chapter 4 will summarize the major contributions of this dissertation as well as some miscellaneous case studies and discuss the open problems in the fields of dynamic coverage and human-aerial/space robot collaboration. This will follow with appendices containing mathematical derivations too lengthy for the main body of the work.

CHAPTER 2

Power-constrained Dynamic Coverage

This chapter presents the culmination of the author's contributions to multi-agent dynamic coverage networks. As described in chapter 1, this work is motivated in part by the fact that the vast majority of coverage literature assumes unlimited power supplies for agents. Few authors have provided protocols for vehicle servicing and redeployment, which nevertheless consider simplified agent models and environment geometries. To the best of the author's knowledge, no prior authors have presented energy-aware deployment and guidance protocols for multi-agent 3D dynamic coverage networks that account for persistent battery replacements at charging stations.

The first portion of this chapter is devoted to the complete coverage of 3D environments by power-constrained agents. In this work, the agents are tasked to move until a satisfactory level of coverage has been reached at all points of the domain. The satisfactory level of coverage is achieved through the gathering of information by a forward-facing sensor footprint, modeled as an anisotropic spherical sector. Theoretical guarantees for coverage and collision avoidance are met by a hybrid controller consisting of four operating modes: *local coverage*, *global coverage*, *waypoint scan* and *subdomain transfer*. Energy-aware methods are encoded into the global coverage state in order to direct less power-constrained agents to cover further away from the charging station. Additionally, a novel domain partitioning strategy is used that directs individual agents to explore within concentric hemispherical shells around a centralized charging station. Based on the proposed partitioning, flight paths are designed that are guaranteed to terminate at the charging station in the limit that agent batteries expire. The efficacy of this algorithm is presented through experimental trials with three agents in an indoor environment. Simulations are provided for ten agents.

The second portion of this chapter is devoted to the persistent coverage of a 2D manifold that has been embedded in 3D space. The manifold is subject to continual impact by intruders that travel at constant velocities along arbitrarily oriented straight-line trajectories. The trajectories of intruders are estimated online with an extended Kalman filter and their predicted impact points contribute normally distributed decay terms to the coverage

level. A formal hybrid control strategy is presented that allows for power-constrained 3D free flyer agents to persistently monitor the domain, track and intercept intruders, and periodically deploy from and return to a single charging station on the manifold. Guarantees on intruder interception with respect to agent power lifespans are formally proven. The efficacy of the algorithm is demonstrated through simulation.

The final portion of this chapter is devoted to an initial venture into HRI in which a human can supervise an agent engaged in dynamic coverage of an environment. This hybrid system adopts the local coverage mode described in section 2.2.2. However, the global coverage mode has been replaced by a set of human gestures which can be used to command the robot's flight to any portion of the domain. In such a way, the human has direct control over which regions of space are explored with local coverage.

The work of this chapter is directly relevant to the real world problem of intruder detection by UAVs patrolling a controlled airspace. Such an environment may arise on short notice (e.g., the construction of a new remote military installation during wartime) or cover a large geographical region (e.g., national borders). In these situations it may be impractical due to time or resources to gain complete coverage of the surrounding area with static sensors. The use of mobile sensors allows one to increase the probability of intruder detection as agents scan the environment [37]. The anisotropic spherical sector sensing model under consideration is abstract enough to embody the physical characteristics of various distance detection sensors (e.g., cameras, light detection and ranging (LiDAR), and sonar) through tuning of its geometric parameters (e.g., range, field of view, quality distributions) while preserving a set of generic coverage control laws. As an example, consider this anisotropic sensor model analogous to distance sampling of data gathered by an ultra wide-angle (e.g., fisheye) camera lens. Distance sampling is a technique commonly used by ecologists in order to correct for animal population densities that appear to spike in close proximity to camera traps [82]. The fundamental concept is that the probability of a population's detection falls with the distance of the population from the sensing node [83]. The robot sensing model of this dissertation encapsulates this while providing a large FOV.

The author wishes to acknowledge his co-authors Tru Hoang and Enkhmurun Bayasgalan who contributed portions of the software used in the experimental trials of section 2.2.

2.1 Notation

The following is a list of variables which frequently appear in this chapter:

- i - Index of mobile sensing agent,

- j - Index of another agent aside from i,
- N - Number of agents in network,
- $\mathcal{D}$ - Domain in $\mathbb{R}^3$,
- $\mathcal{G}$ - Global Cartesian coordinate frame with axes $\hat{x}_{\mathcal{G}}$, $\hat{y}_{\mathcal{G}}$, and $\hat{z}_{\mathcal{G}}$,
- $\mathcal{O}$ - Origin of $\mathcal{G}$,
- $\mathcal{C}$ - Ellipsoid of revolution embedded in $\mathcal{D}$,
- $\mathcal{C}_{\mu_i}$ - Ellipsoid of revolution concentric with $\mathcal{C}$ but having semi-principal axes $(\ell + \gamma)\, R$ longer than $\mathcal{C}$ for all $\mu_i \in \{0, ..., N-1\}$,
- $A_{\mathcal{C}}$ - Surface area of $\mathcal{C}$,
- $\mathcal{B}_i$ - Agent i's body fixed coordinate frame with axes $\hat{x}_{\mathcal{B}_i}$, $\hat{y}_{\mathcal{B}_i}$, and $\hat{z}_{\mathcal{B}_i}$,
- $p_i = [x_i, y_i, z_i]^T$ - Agent i's Cartesian position vector relative to $\mathcal{O}$ resolved in $\mathcal{G}$.
- $\Omega_i = [\Phi_i, \Theta_i, \Psi_i]^T$ - Orientation of $\mathcal{B}_i$ relative to $\mathcal{G}$ resolved into a 3-2-1 Euler angle sequence,
- $u_i^{()}$ - Agent i's linear velocity along x-axis of $\mathcal{B}_i$,
- $v_i^{()}$ - Agent i's linear velocity along y-axis of $\mathcal{B}_i$,
- $w_i^{()}$ - Agent i's linear velocity along z-axis of $\mathcal{B}_i$,
- $q_i^{()}$ - Agent i's angular velocity about x-axis of $\mathcal{B}_i$,
- $r_i^{()}$ - Agent i's angular velocity about y-axis of $\mathcal{B}_i$,
- $s_i^{()}$ - Agent i's angular velocity about z-axis of $\mathcal{B}_i$,
- $\tilde{q}_i$ - Agent i's state vector comprised of elements p_i and Ω_i,
- $\bar{R}_{\mathcal{D}}$ - Radius of hemispherical domain of section 2.2,
- $\mathcal{R}_1$ - Orientation matrix of agent i as presented in (2.1),
- $\mathcal{R}_2$ - Body fixed rate to Euler angle rate transformation matrix as presented in (2.2),
- $\mathcal{S}_i$ - Agent i's forward facing anisotropic sensing footprint,

- R_i - Range of $\mathcal{S}_i$ in section 2.2,
- R - Range of $\mathcal{S}_i, \forall i$ in section 2.3,
- α_i - Angle between centerline of $\mathcal{S}_i$ (i.e., $\hat{x}_{\mathcal{B}_i}$) and the periphery (i.e., half of the FOV),
- $\tilde{p}$ - Point within $\mathcal{D}$,
- $S_i\left(\tilde{q}_i, \tilde{p}\right)$ - Quality of sensing currently available at $\tilde{p}$ due to observation by agent i,
- c_{1i} - Sensing range constraint function for agent i,
- c_{2i} - Sensing angle constraint function for agent i,
- β_i - Parameter which ensures continuity in $S_i\left(\tilde{q}_i, \tilde{p}\right)$,
- η_i - Tuning Parameter of β_i which affects how close peak sensing is from the vertex of $\mathcal{S}_i$.
- $r_{\tilde{p}/p_i}$ - Position of $\tilde{p}$ relative to p_i resolved in $\mathcal{G}$,
- ϕ_i - Angle between $r_{\tilde{p}/p_i}$ and $\hat{x}_{\mathcal{B}_i}$,
- $\mathscr{z}_i$ - Radius of sphere that bounds the size of agent i,
- C_{ki} - Defined as $\max\{0, c_{ki}\}, \forall k \in \{1, 2\}$,
- $\bar{C}_i$ - The set of zero elements in C_{ki},
- ∂ - Denotes the boundary of a region,
- $Q_i\left(t, \tilde{p}\right)$ - The coverage level provided by agent i at time t,
- $Q\left(t, \tilde{p}\right)$ - The global coverage level,
- $E(t)$ - The global coverage error,
- $C^\star$ - The desired coverage level for any point $\tilde{p}$,
- t_f - Final time at which coverage is complete,
- $V_i\left(t\right)$ - Battery voltage of agent i,
- V^{co} - Cut off voltage, i.e., minimum safe voltage for agent flight,

- V_0 - Battery voltage at agent deployment,
- t_i^{co} - Cut off time, i.e., $V_i(t_i^{co}) = V^{co}$,
- t_i^d - Deployment time of agent i,
- T_i - Lifespan of agent i's battery, i.e., $T_i = t_i^{co} - t_i^d$,
- T_i^w - Waiting period of agent i before redeployment,
- $\bar{T}$ - Upper bound on maximum battery lifespan of any agent in network,
- $\hat{e}_i\left(t\right)$ - Agent i's contribution to $\dot{E}\left(t\right)$,
- $a_{ik}\left(t, Q\left(t, \tilde{p}\right)\right), \forall k \in \{0, 1, 2, 3, 4, 5\}$ - Terms which may be summed in linear combination with agent velocity components to yield $\dot{\hat{e}}_i(t)$,
- $k_u^{()}$ - Proportional gain for input $u_i^{()}$,
- $k_v^{()}$ - Proportional gain for input $v_i^{()}$,
- $k_w^{()}$ - Proportional gain for input $w_i^{()}$,
- $k_r^{()}$ - Proportional gain for input $r_i^{()}$,
- $k_s^{()}$ - Proportional gain for input $s_i^{()}$,
- $R_{b,ij}$ - Buffer distance of agent i from j,
- ϱ - An arbitrarily small positive real number,
- $\acute{Q}_{ij}\left(t, \tilde{p}\right)$ - Coverage map avoidance term,
- $M_{1i}\left(p_j, \tilde{p}\right)$ - Map augmentation term 1,
- $\bar{B}_{R_{b,ij}}(p_j)$ - Closed ball of radius $R_{b,ij}$ centered at p_j,
- $\acute{Q}_i\left(r, \tilde{p}\right)$ - Avoidance augmented global coverage map for agent i,
- $\mathcal{D}_\ell$ - Set of concentric hemispherical shell domains $\forall \ell \in \{1, ..., N-1\}$,
- $\mathcal{D}_{i_{cr}}$ - Central hemispherical subdomain,
- $\bar{R}_{cr}$ - Radius of $\mathcal{D}_{i_{cr}}$,
- i_{cr} - Index of power critical agent,

- i_ℓ - Battery voltage sorted agent indices,
- $\acute{Q}_{i_{cr}}(t, \tilde{p})$ - Energy-aware augmented global coverage level for agent i_{cr},
- $M_{2i_{cr}}(t, \tilde{p})$ - Map augmentation term 2,
- κ_{icr} - Tuning parameter for rate at which $\bar{R}_{cr} \to 0$,
- U_{max} - Maximum linear velocity of slowest agent in network,
- V_{rem} - The volume of $\mathcal{D} \setminus \mathcal{D}_{i_{cr}}$,
- $\bar{R}_\rho$ - The inner radius of hemispherical shell $\mathcal{D}_\rho, \forall \rho \in \{2, ..., N-2\}$,
- $\acute{Q}_\ell(t, \tilde{p})$ - Agent ℓ's augmented global coverage level,
- $M_{3\ell}(t, \tilde{p})$ - Map augmentation term 3,
- ε_1 - Threshold for triggering global coverage mode,
- ε_2 - Threshold of acceptable position convergence,
- ε_3 - Threshold of acceptable orientation convergence,
- ε_4 - Threshold of acceptable distance for agent from assigned particle at time of interception,
- ε_5 - Altitude convergence threshold for surface transfer mode,
- δ - The Dirac delta function,
- $\mathcal{D}^j$ - Cube shaped partition of $\mathcal{D}$,
- N_c - Number of elements $\mathcal{D}^j$,
- W^j - Centroid of $\mathcal{D}^j$,
- $V_{\mathcal{D}^j}$ - Volume of $\mathcal{D}^j$,
- $\hat{C}_i^j$ - Anticipated local coverage error for agent i in cube $\mathcal{D}^j$,
- t_{gc} - Time at which global coverage initiates,
- $\gamma_{ik} \forall k \in \{1, ..., 4\}$ - Cost weights to penalize time history of actuation effort,
- Γ_i - Actuation accumulation function,

- Λ_{i0} - Static weight to penalize waypoint distance,
- Λ_{i1} - Static weight to penalize actuation effort in waypoint selection,
- $\tilde{j}$ - Index of optimal global coverage cube for agent i,
- $p_{\star,i} = [x_{\star,i}, y_{\star,i}, z_{\star_i}]^T$ - Agent i's destination position,
- $\Psi_{\star,i}$ - Agent i's destination yaw angle,
- $\tilde{q}_{\star,i}$ - Agent i's desired state vector,
- $\mathcal{V}_{i0}$ - Nominal Lyapunov-like function,
- c_{4ij} - Global avoidance constraint,
- $b_{ij}\left(p_i, p_j\right)$ - Logarithmic barrier function,
- $q_{ij}\left(p_i, p_j\right)$ - Recentered barrier function,
- $\mathcal{V}'_{ij}(p_i, p_j)$ - Avoidance Lyapunov-like function,
- σ_{ij} - Avoidance activation bump function,
- d_{ij} - Euclidean distance between agents i and j,
- ν_i - Augmented nominal Lyapunov-like function,
- $\mathcal{V}_i$ - Scaled candidate Lyapunov-like function,
- ζ_{ik} - Discrete states of hybrid automaton,
- $G\left(\zeta_{ik}, \zeta_{ik}\right)$ - Guard conditions of hybrid automaton,
- $x_{\mathcal{C},r}$ - Semi-major axis of $\mathcal{C}$,
- $z_{\mathcal{C},r}$ - Semi-minor axis of $\mathcal{C}$,
- $Q_i^{\mathcal{C}}\left(t, \tilde{p}\right)$ - The coverage level provided by agent i at time t along $\mathcal{C}$,
- $C\left(\tilde{p}\right)$ - Takes non-zero value only along surface of $\mathcal{C}$,
- N_p - Number of particle intruders,
- $\Lambda_k\left(\tau, \tilde{p}\right)$ - Map decay term of particle k,
- $Q^{\mathcal{C}}\left(t, \tilde{p}\right)$ - The global coverage level along $\mathcal{C}$,

- n_i - Vector normal to $\mathcal{C}$ which intersects position of agent i and has length equal to the altitude of i above $\mathcal{C}$,
- $T^\star$ - Battery lifespan of agents in section 2.3,
- $\hat{T}$ - A moving time window in which a maximum of $N-1$ particle impacts on $\mathcal{C}$ are serviceable,
- $\mathcal{F}$ - Agent fueling station in section 2.3,
- R_{det} - Lower bound on distance from intruder detection to impact,
- $\tilde{q}_k = [x_k, y_k, z_k, \dot{x}_k, \dot{y}_k, \dot{z}_k]^T$ - Cartesian state vector of particle k resolved in $\mathcal{G}$,
- $\tilde{z}_k = [\rho_k, \theta_k, \psi_k]^T$ - Spherical coordinate measurement vector of particle k resolved in $\mathcal{G}$ which consists of range, azimuthal angle, and polar angle respectively,
- ϵ - Zero-mean Gaussian measurement noise,
- U_{max}^{int} - Upper bound on particle speed,
- $\tilde{h}\left(x_k, y_k, z_k\right)$ - Refers to vector in (2.62),
- $\mathbf{R} = diag\left(\sigma_\rho^2, \sigma_\theta^2, \sigma_\psi^2\right)$ - Measurement covariance matrix,
- $\hat{\tilde{q}}_k = \left[\hat{x}_k, \hat{y}_k, \hat{z}_k, \dot{\hat{x}}_k, \dot{\hat{y}}_k, \dot{\hat{z}}_k\right]^T$ - Particle state estimate,
- $\mathbf{P}_k$ - Particle state estimate covariance matrix,
- t_{dk} - Detection time of particle k,
- t_{ck} - Estimated impact time of particle k,
- t_{ik} - Time required to travel from $p_i(t_{dk})$ to $proj_{\mathcal{C}_0}\tilde{p}_k'(t_{ck})$,
- $t_{i\mathcal{F}}$ - The amount of time since agent i's last deployment from $\mathcal{F}$,
- $\tilde{p}_k'\left(t+\tau\right) = \left[\tilde{x}_k'\left(t+\tau\right), \tilde{y}_k'\left(t+\tau\right), \tilde{z}_k'\left(t+\tau\right)\right]^T$ - Particle k's predicted position at time $t+\tau$,
- $\mathbf{P}_k\left(t+\tau\right)$ - Particle k's predicted state estimate covariance at time $t+\tau$,
- $T_{H,k}\left(t\right)$ - Time horizon lower-bounded by estimate for remaining time until particle impact,

- λ_k - Tuning parameter for Map decay term,
- $\mathcal{N}$ - Denotes a Gaussian distribution,
- $G(\tau)$ - Matrix to propagate particle state estimate forward in time,
- $E^{\mathcal{C}}(t)$ - The global coverage error along $\mathcal{C}$,
- $\hat{e}_i^{\mathcal{C}}$ - Agent i's contribution to $E^{\mathcal{C}}(t)$,
- $\tilde{e}_k^{\mathcal{C}}$ - Particle k's contribution to $E^{\mathcal{C}}(t)$,
- $a_{ik}^{\mathcal{C}}(t, Q(t, \tilde{p})), \forall k \in \{0, 1, 2, 3, 4, 5\}$ - Terms which may be summed in linear combination with agent velocity components to yield $\dot{\hat{e}}_i^{\mathcal{C}}(t)$,
- $\rho_{l,i}$ - Term to encode collision avoidance with respect to surface of interest,
- $\rho_{a,i}$ - Term to align $\hat{x}_{\mathcal{B}_i}$ with $-\hat{n}_i$,
- $\bar{r}_i$ - Saturation limit for $r_i^{\mathcal{C},loc}$,
- $\bar{s}_i$ - Saturation limit for $s_i^{\mathcal{C},loc}$,
- γ - Tunes the altitude at which agent flies relative to $\mathcal{C}$,
- ξ - Proportional gain on rate at which $\hat{x}_{\mathcal{B}_i}$ tends to align with $-\hat{n}_i$,
- U_{max}^{agt} - An upper bound to agent velocity tangential to $\mathcal{C}$,
- i_k - The index of the agent assigned to intercept particle k,
- f_i - Flag to denote whether agent i is currently assigned to a particle,
- $i_p(t)$ - The power index of agent i described in section 2.3.4,
- μ_i - Surface assignment index for agent i,
- $proj_{\mathcal{C}_{\mu_i}} \tilde{p}_k'(t_{ck})$ - The projection point of $\tilde{p}_k'(t_{ck})$ onto $\mathcal{C}_{\mu_i}$,
- χ - Heading angle output from Vincenty's algorithm,
- $\hat{\iota}_i$ - Heading unit vector computed from χ,
- $g_{\mathcal{C}_{\mu_i}}$ - Function of the semi-principal axes of surface $\mathcal{C}_{\mu_i}$ defined in (2.81),
- $\mathcal{P}_{max}$ - Upper bound to maximum path length agent shall take to intercept any particle,

- $\mathcal{P}_{geo}$ - Geodesic path length between two points on $\mathcal{C}_{\mu_i}$,
- $\mathcal{P}_{lat}$ - Path of constant latitude along $\mathcal{C}_{\mu_i}$,
- $\mathcal{P}_{long}$ - Path of constant longitude along $\mathcal{C}_{\mu_i}$,
- $\bar{z}_{i_p-2}$ - Upper bound to the latitude partition for agent i_p,
- $\bar{z}_{i_p-1}$ - Lower bound to the latitude partition for agent i_p,
- $\bar{j}$ - The set of agents in $i \cup j$ such that $\|p_i - p_j\| \leq R$ and $i \neq j$,
- i_{pr} - The agent with the largest value for $t_{\bar{j}\mathcal{F}}$.

2.2 Complete Coverage of 3D Environments

2.2.1 Problem Formulation

2.2.1.1 System Model

In this work, a set of kinematic guidance laws are derived that can be issued to velocity-tracking controllers on many commercially available quadrotors. The inputs to this controller are the three body-fixed Cartesian linear velocities and a body-fixed z-axis angular velocity.

Assume a network of mobile agents indexed $i \in \{1, ..., N\}$ that are subject to 3-D rigid body kinematics [84]:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} = \begin{bmatrix} \cos\Theta_i \cos\Psi_i & \sin\Phi_i \sin\Theta_i \cos\Psi_i - \cos\Phi_i \sin\Psi_i & \cos\Phi_i \sin\Theta_i \cos\Psi_i + \sin\Phi_i \sin\Psi_i \\ \cos\Theta_i \sin\Psi_i & \sin\Phi_i \sin\Theta_i \sin\Psi_i + \cos\Phi_i \cos\Psi_i & \cos\Phi_i \sin\Theta_i \sin\Psi_i - \sin\Phi_i \cos\Psi_i \\ -\sin\Theta_i & \sin\Phi_i \cos\Theta_i & \cos\Phi_i \cos\Theta_i \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix}, \quad (2.1)$$

$$\begin{bmatrix} \dot{\Phi}_i \\ \dot{\Theta}_i \\ \dot{\Psi}_i \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi_i \tan\Theta_i & \cos\Phi_i \tan\Theta_i \\ 0 & \cos\Phi_i & -\sin\Phi_i \\ 0 & \sin\Phi_i \sec\Theta_i & \cos\Phi_i \sec\Theta_i \end{bmatrix} \begin{bmatrix} q_i \\ r_i \\ s_i \end{bmatrix}, \quad (2.2)$$

where $p_i = [x_i \;\; y_i \;\; z_i]^T$ is the position vector and $\Omega_i = [\Phi_i \;\; \Theta_i \;\; \Psi_i]^T$ is the vector of 3-2-1 Euler angles taken with respect to a global Cartesian coordinate frame $\mathcal{G}$ with origin $\mathcal{O}$. The linear velocities $[u_i \;\; v_i \;\; w_i]^T$ and angular velocities $[q_i \;\; r_i \;\; s_i]^T$ are both expressed in the body fixed frame $\mathcal{B}_i$ with origin p_i. The state vector of agent i is defined as $\tilde{q}_i = [p_i^T \;\; \Omega_i^T]^T$ and the control inputs are u_i, v_i, w_i and s_i respectively. The rotation matrices in (2.1) and (2.2) shall be denoted $\mathcal{R}_1$ and $\mathcal{R}_2$ respectively for the remainder of this chapter. The agents

are confined to stationary hemispherical domain, $\mathcal{D} \subset \mathbb{R}^3$ of radius $\bar{R}_{\mathcal{D}}$, which must be fully surveyed. This is illustrated in Fig. 2.1

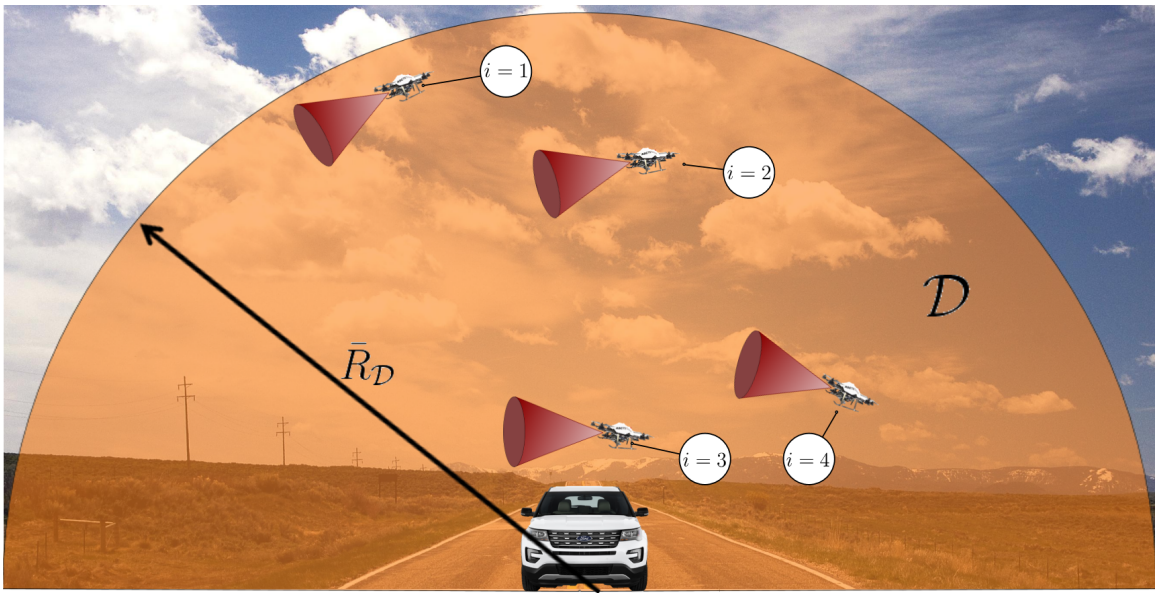


Figure 2.1: In the illustrated application, a team of quadrotors patrols the airspace surrounding a parked ground vehicle. The agents are confined to stationary hemispherical domain, $\mathcal{D} \subset \mathbb{R}^3$ of radius $\bar{R}_{\mathcal{D}}$, which must be fully surveyed.

Each agent, i, is equipped with a forward facing sensor whose spherical sector footprint shall be referred to as $\mathcal{S}_i$. This model, though intended to be generic, is similar to conical camera models presented in other works on dynamic coverage (see [43]). The model under consideration differs with respect to earlier approaches in terms of its heterogeneity, i.e., $\mathcal{S}_i$ provides anisotropic sensing data that degrade in quality towards the periphery of the footprint and changes with respect to distance from the sensor. Degradation over distance is not monotonically decreasing but instead contains a peak located near the vertex of $\mathcal{S}_i$ as in [85]. This is motivated by the fact that the probability of target detection by a camera decreases when the target is either very far from or very close to the lens. Anisotropic sensing is encoded through the definition of the sensing constraint functions for each agent i:

$$c_{1i} = \beta_i R_i^2 - (\tilde{x} - x_i)^2 - (\tilde{y} - y_i)^2 - (\tilde{z} - z_i)^2, \tag{2.3a}$$

$$c_{2i} = \alpha_i - \phi_i, \tag{2.3b}$$

for $\beta_i = \min\{1, \eta_i \left((\tilde{x} - x_i)^2 + (\tilde{y} - y_i)^2 + (\tilde{z} - z_i)^2\right)\}$ with real constant η_i constrained such that $\eta_i > \frac{1}{R_i^2}$ and $\eta_i > 1$. R_i is the sensing range, $\tilde{p} = [\tilde{x} \;\; \tilde{y} \;\; \tilde{z}]^T$ is the position

of a point in $\mathcal{D}$ relative to $\mathcal{G}$, α_i is the angle between the periphery and centerline of the spherical sector (the $\hat{x}_{\mathcal{B}_i}$ axis), and ϕ_i is the angle between $r_{\tilde{p}/p_i} = \tilde{p} - p_i$ (resolved in $\mathcal{G}$ by construction) and the $\hat{x}_{\mathcal{B}_i}$ axis given as the inverse cosine of the dot product of $\hat{r}_{\tilde{p}/p_i}$ and $\hat{x}_{\mathcal{B}_i}$ resolved in $\mathcal{G}$:

$$\phi_i = \arccos\left(\hat{r}_{\tilde{p}/p_i} \cdot \hat{x}_{\mathcal{B}_i}|_{\mathcal{G}}\right). \tag{2.4}$$

Note that:

$$\hat{r}_{\tilde{p}/p_i} = \frac{1}{\sqrt{(\tilde{x}-x_i)^2 + (\tilde{y}-y_i)^2 + (\tilde{z}-z_i)^2}} \begin{bmatrix} (\tilde{x}-x_i) \\ (\tilde{y}-y_i) \\ (\tilde{z}-z_i) \end{bmatrix},$$

and $\hat{x}_{\mathcal{B}_i}|_{\mathcal{G}}$ is determined by multiplying $\mathcal{R}_1$ by $[1 \;\; 0 \;\; 0]^T$:

$$\hat{x}_{\mathcal{B}_i}|_{\mathcal{G}} = \begin{bmatrix} \cos\Psi_i \cos\Theta_i \\ \sin\Psi_i \cos\Theta_i \\ -\sin\Theta_i \end{bmatrix}.$$

Agent i is thus capable of detecting objects that lie within an angle of $2\alpha_i > 0$ about the $\hat{x}_{\mathcal{B}_i}$ axis and a range of $R_i > 0$. Assume that the body of agent i may be bounded with a sphere of radius r_i centered at p_i. This model for agent i is depicted in Fig. 2.2. Let us denote $\max\{0, c_{ki}\} = C_{ki}$. One can define the sensing function that represents the quality of information available at each point $\tilde{p}$ over the sensing domain as:

$$S_i(\tilde{q}_i, \tilde{p}) = \begin{cases} \frac{C_{1i}C_{2i}}{C_{1i}+C_{2i}}, & \text{if } card\left(\bar{C}_i\right) < 2 \wedge \|r_{\tilde{p}/p_i}\| > 0; \\ 0, & \text{otherwise,} \end{cases} \tag{2.5}$$

where $\bar{C}_i$ is the set of zero elements in C_{ki}. $S_i(\tilde{q}_i, \tilde{p})$ takes a value of zero outside of $\mathcal{S}_i$. Note that $S_i(\tilde{q}_i, \tilde{p})$ is defined over all of $\mathcal{D}$ and thus has static bounds. $S_i(\tilde{q}_i, \tilde{p})$ is continuous in $\tilde{p}$ while taking a value of zero along $\partial\mathcal{S}_i$. In order to verify continuity at $\partial\mathcal{S}_i$, one must demonstrate that $\frac{C_{1i}C_{2i}}{C_{1i}+C_{2i}}$ tends to zero in the limit that C_{1i} and C_{2i} approach zero. Furthermore, one must also demonstrate that $\frac{C_{1i}C_{2i}}{C_{1i}+C_{2i}}$ tends to zero in the limit that $\|r_{\tilde{p}/p_i}\|$ approaches zero, i.e., near the vertex of $\mathcal{S}_i$. This property arises from the definitions of β_i and η_i which encode that $S_i(\tilde{q}_i, \tilde{p})$ drops off rapidly when in very close proximity to the vertex of $\mathcal{S}_i$. Increasing $\eta_i >> 1$ has the effect of shifting the sensing drop off point closer to p_i.

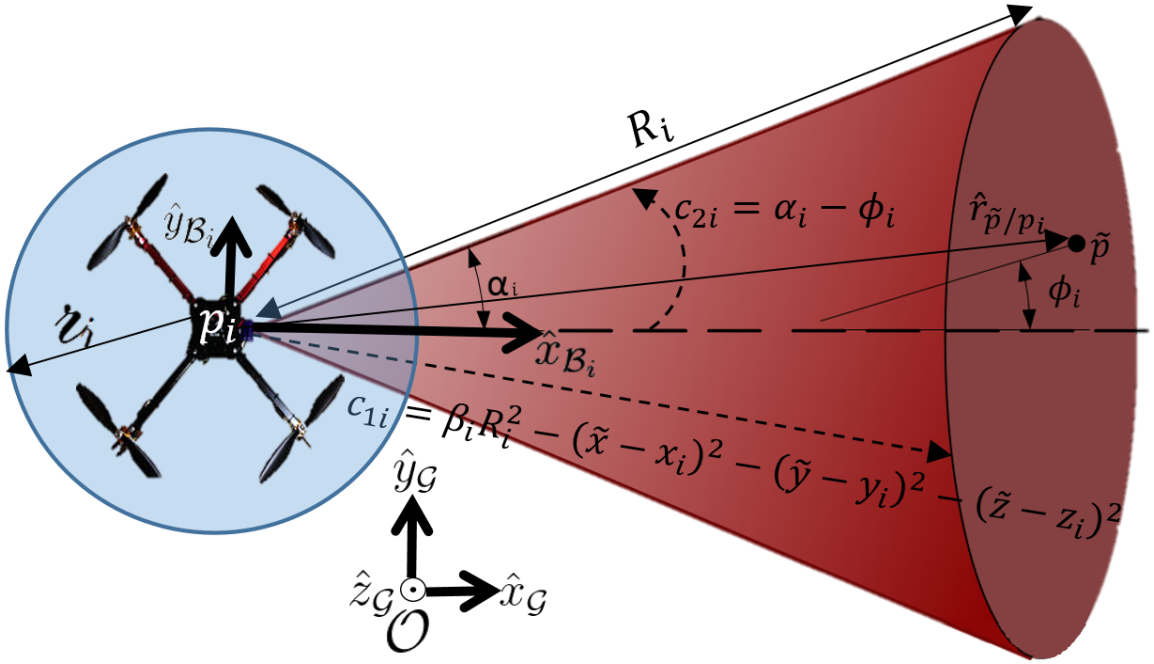


Figure 2.2: The body of agent i is bounded by a sphere of radius $\mathcal{V}_i$ and possesses $\mathcal{S}_i$.

Proposition 1. $\lim_{\substack{C_{1i}\to 0^+ \\ C_{2i}\to 0^+}} \frac{C_{1i}C_{2i}}{C_{1i}+C_{2i}} = 0.$

Proof. Let us consider the fact that:

$$0 < \frac{C_{1i}}{C_{1i} + C_{2i}} \leq 1, \ \forall C_{1i} > 0, C_{2i} > 0.$$

Multiplying by C_{2i} provides:

$$0 < \frac{C_{1i}C_{2i}}{C_{1i} + C_{2i}} \leq C_{2i}, \ \forall C_{1i} > 0, C_{2i} > 0.$$

Noting that $\lim_{\substack{C_{1i}\to 0^+ \\ C_{2i}\to 0^+}} 0 = 0$ and $\lim_{\substack{C_{1i}\to 0^+ \\ C_{2i}\to 0^+}} C_{2i} = 0$, one arrives at $\lim_{\substack{C_{1i}\to 0^+ \\ C_{2i}\to 0^+}} \frac{C_{1i}C_{2i}}{C_{1i}+C_{2i}} = 0$ via the squeeze theorem. This concludes the proof. □

Proposition 2. $\lim_{\|r_{\tilde{p}/p_i}\|\to 0} \frac{C_{1i}C_{2i}}{C_{1i}+C_{2i}} = 0.$

Proof. Note throughout this proposition that $C_{1i} > 0$ for $\|r_{\tilde{p}/p_i}\|$ arbitrarily close to zero and that $C_{2i} \geq 0$ by definition. First, assume that $C_{2i} = 0$. This physically implies that $\tilde{p}$ is approaching p_i from a trajectory that lies either outside of or on the boundary of $\mathcal{S}_i$, i.e.,

$\phi_i \geq \alpha_i$. It follows from this assumption that:

$$\lim_{\|r_{\tilde{p}/p_i}\|\to 0} \frac{C_{1i}C_{2i}}{C_{1i}+C_{2i}} = \lim_{\|r_{\tilde{p}/p_i}\|\to 0} 0 = 0 \text{ for } C_{2i} = 0.$$

Now consider the alternative case that $C_{2i} > 0$, i.e., $\tilde{p}$ is approaching p_i from a trajectory inside of $\mathcal{S}_i$. Under this assumption, substituting for C_{1i} in the argument of the limit yields:

$$\frac{\max\{0, c_{1i}\}C_{2i}}{\max\{0, c_{1i}\}+C_{2i}},$$

which is equivalent to:

$$\frac{\max\{0, \beta_i R_i^2 - (\tilde{x}-x_i)^2 - (\tilde{y}-y_i)^2 - (\tilde{z}-z_i)^2\}C_{2i}}{\max\{0, \beta_i R_i^2 - (\tilde{x}-x_i)^2 - (\tilde{y}-y_i)^2 - (\tilde{z}-z_i)^2\}+C_{2i}}.$$

Substituting in expressions for β_i and $\|r_{\tilde{p}/p_i}\|$ yields:

$$\frac{\max\{0, \min\{1, \eta_i\|r_{\tilde{p}/p_i}\|^2\}R_i^2 - \|r_{\tilde{p}/p_i}\|^2\}C_{2i}}{\max\{0, \min\{1, \eta_i\|r_{\tilde{p}/p_i}\|^2\}R_i^2 - \|r_{\tilde{p}/p_i}\|^2\}+C_{2i}}.$$

The limit of this expression can be stated as:

$$\lim_{\|r_{\tilde{p}/p_i}\|\to 0} \frac{C_{1i}C_{2i}}{C_{1i}+C_{2i}} = \lim_{\|r_{\tilde{p}/p_i}\|\to 0} \frac{\max\{0, \min\{1, \eta_i\|r_{\tilde{p}/p_i}\|^2\}R_i^2 - \|r_{\tilde{p}/p_i}\|^2\}C_{2i}}{\max\{0, \min\{1, \eta_i\|r_{\tilde{p}/p_i}\|^2\}R_i^2 - \|r_{\tilde{p}/p_i}\|^2\}+C_{2i}},$$

whose solution follows from direct substitution of $\|r_{\tilde{p}/p_i}\| = 0$ under the assumption that $C_{2i} > 0$:

$$\lim_{\|r_{\tilde{p}/p_i}\|\to 0} \frac{C_{1i}C_{2i}}{C_{1i}+C_{2i}} = 0, \ \forall C_{2i} > 0.$$

This concludes the proof. □

Define the coverage level provided by agent i at time t as:

$$Q_i(t, \tilde{p}) = \int_0^t S_i\left(\tilde{q}_i(\tau), \tilde{p}\right) \, d\tau. \tag{2.6}$$

The global coverage level is defined as:

$$Q(t, \tilde{p}) = \sum_{i=1}^{N} Q_i(t, \tilde{p}). \tag{2.7}$$

Remark 1. To summarize the problem formulation, dynamic coverage refers to the accumulation of sensing data over time. The effect of agent i's motion is to vary the points $\tilde{p} \in \mathcal{D}$ for which $S_i(\tilde{q}_i, \tilde{p})$ is nonzero. The domain $\mathcal{D}$ is said to be completely covered at a final time t_f when $Q(t_f, \tilde{p}) \geq C^\star, \forall \tilde{p} \in \mathcal{D}$ where $C^\star$ is a predefined desired coverage level. Complete coverage is equivalent to driving the global coverage error, defined in section 2.2.2.1, to zero.

2.2.1.2 Network Safety Through Energy-aware Deployment

Assumption 1. A communication network is employed such that agent i has access to $Q(t, \tilde{p})$ as well as $\tilde{q}_j(t)$ and $V_j(t), \forall j \in \{1, ..., N\}$ where $V_j(t)$ is the battery voltage of agent j.

While previous works, such as [86] and [9], have considered network safety in terms of collision avoidance, the few that have incorporated power limitations, such as [52–54], do not model the sensing capabilities of the agents. The mobile sensing agents in the proposed approach are assumed to have a constrained on-board power source (e.g., a battery) which must be periodically replaced at a centralized station. Failure to do so will result in agents becoming stranded and potentially compromised. Consider that each agent $i \in \{1, ..., N\}$ is powered by a battery with voltage $V_i(t)$, whose initial value at deployment time t_i^d is $V_i(t_i^d) = V_0$. Define the cutoff voltage, V^{co} as the minimum voltage for which an agent may fly reliably. Then, $V_i(t_i^{co}) = V^{co}$, where t_i^{co} is the cutoff time. Define agent i's nominal battery lifespan $T_i = t_i^{co} - t_i^d$. Consider also that if agent i is deployed at time t_i^d and lands at time t_i^{co}, then it shall hold for a waiting period T_i^w such that $T_i + T_i^w = \bar{T}$, where $\bar{T}$ is an upper bound on the maximum battery lifespan of any agent in the network.

Assumption 2. If $p_i\left(t_i^{co}\right) = \mathcal{O}$, i.e., if agent i reaches the origin at the cutoff time, then the voltage of agent i is fully restored within time T_i^w, i.e., $V_i(t_i^{co} + T_i^w) = V_0$.

This assumption models the scenario in which a ground servicing station is located at the origin. The agents are required to land periodically at this station to have their batteries switched out for fully charged ones. This scenario is presented in Fig. 2.3.

Definition 1. Two agents i and j are said to be collision-free so long as $\|p_i(t) - p_j(t)\| > \mathscr{r}_i + \mathscr{r}_j, \forall t$.

Remark 2. Safety is guaranteed $\forall\, i$ as long as 1) $V_i(t) \geq V^{co}, \ \forall t$ and $V_i = V^{co} \implies p_i = \mathcal{O}$ and 2) i is collision-free with all agents $j \neq i$.

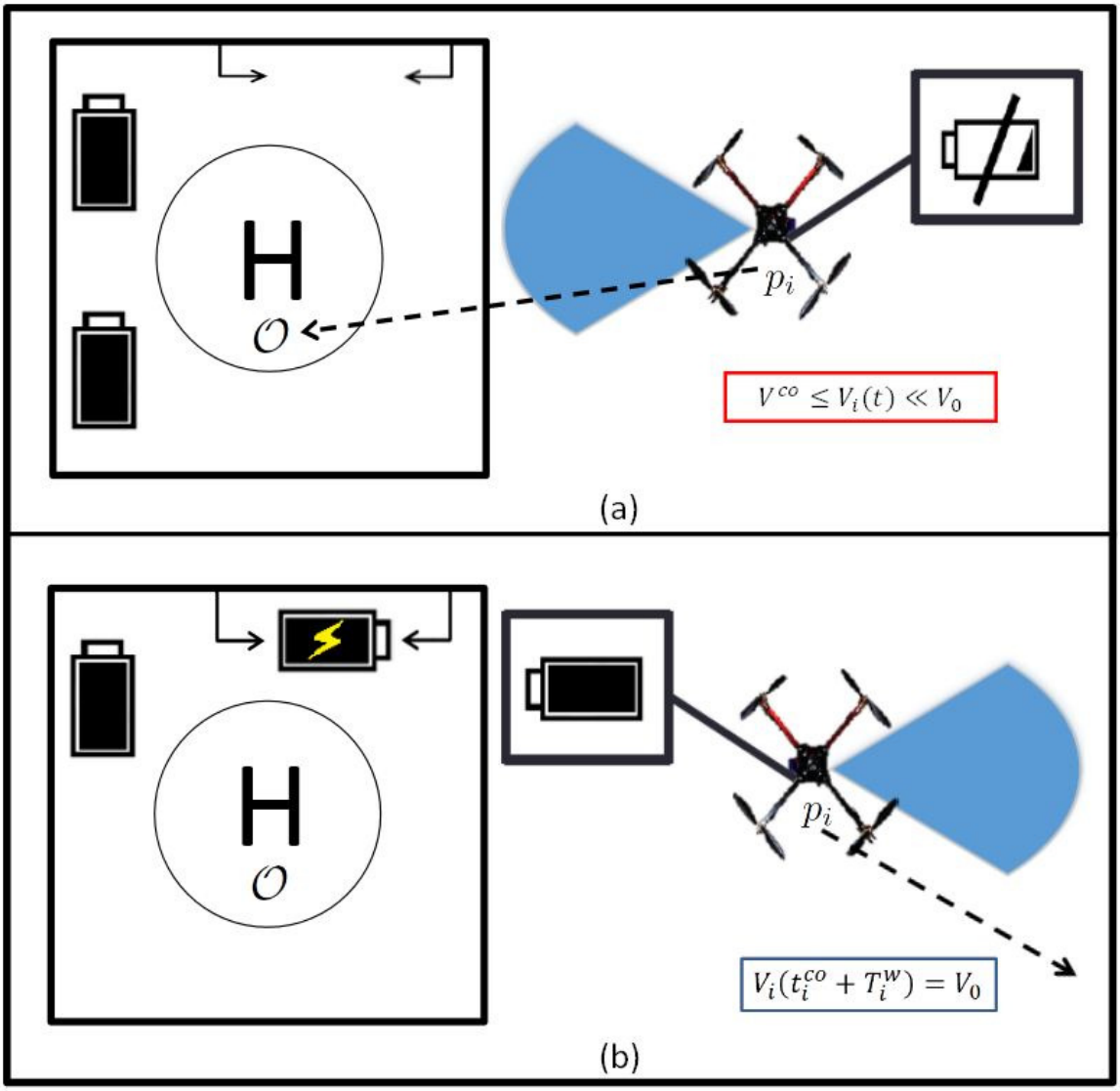


Figure 2.3: Agent i lands upon a helipad at $\mathcal{O}$ as $V_i(t) \to V^{co}$. After a waiting period, T_i^w, the voltage is instantaneously restored to V_0.

The purpose of section 2.2 is to develop and demonstrate techniques that will achieve the goals of Remark 1 and Remark 2. This is accomplished via four unique modes of operation, termed local coverage mode, global coverage mode, subdomain transfer mode, and waypoint scan mode, which shall be elaborated upon in the following section. These four modes are represented by five hybrid states.

2.2.1.3 Overview

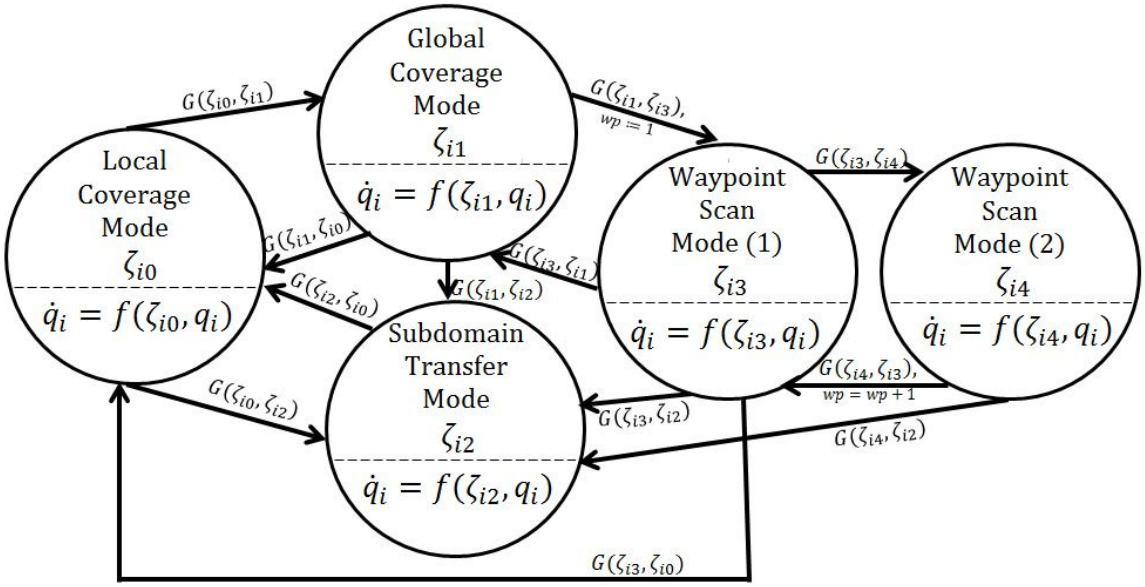


Figure 2.4: Agent i operates in accordance with this automaton.

The coverage strategy for agent i is represented by the hybrid automaton in Fig. 2.4. Note that each agent operates in accordance to its own automaton and thus an arbitrary number of agents may be in any operating mode at any given time. Before proceeding, it is prudent to provide a brief overview of the purpose of each mode.

Local coverage refers to and realizes the active exploration of $\mathcal{D}$ by agent i. In this mode, the agent continuously seeks to orient and translate its sensing footprint $\mathcal{S}_i$ such that the volume of uncovered space within $\mathcal{S}_i$ is increased. This is conceptually similar to following the gradient of the global coverage level's deviation from $C^\star$, i.e., the global coverage error. This mode becomes ineffective if and when the space surrounding the agent has been completely covered. In this situation, no infinitesimal translation or rotation of $\mathcal{S}_i$ will increase the volume of uncovered space within $\mathcal{S}_i$. Thus, when the rate of change of the global coverage error by agent i drops below some threshold, global coverage mode is activated.

Global coverage selects a waypoint for agent i to travel to before resuming local coverage. The waypoint is selected by a cost function that considers the coverage level in the local vicinity of the waypoint, the distance of the waypoint from the agent, and the agent's time-history of energy consumption. If the threshold described in the previous paragraph is still violated at the waypoint, the agent will switch into waypoint scan mode in order to perform one final sweep of the waypoint's local area.

Waypoint scan mode (1) guides agents between a set of 13 sub-waypoints that surround the immediate vicinity of the global coverage waypoint. Waypoint scan mode (2) controls a 360° yaw revolution of $\mathcal{S}_i$ at each sub-waypoint. Upon completion of this scan the agent typically uses global coverage to select a new waypoint at which it may attempt to resume local coverage.

Subdomain transfer mode is used to transfer agents between subdomain partitions of $\mathcal{D}$. These subdomains are concentric hemispherical shells that surround a battery charging station. Throughout its operation, the agent is continually transfered to interior subdomains until it occupies the innermost shell, whose boundaries converge upon the charging station in the limit that the battery expires. Subdomain transfer is activated from any other mode if the agent exceeds some distance from the charging station.

A rigorous definition of all entities of the automaton, including the guard conditions present in Fig. 2.4, is presented in Appendix A.1.

2.2.2 Local Coverage Mode

2.2.2.1 Nominal Control Strategy

Local coverage is the nominal mode that directs active exploration by the agents. Define the global coverage error with respect to $C^\star$ as:

$$E(t) = \int_{\mathcal{D}} h(C^\star - Q(t, \tilde{p}))\, d\tilde{p}. \tag{2.8}$$

where $h(w) = (\max\{0, w\})^3$ with first derivative $h' = \frac{dh}{dw} = 3(\max\{0, w\})^2$ and second derivative $h'' = \frac{d^2h}{dw^2} = 6(\max\{0, w\})$. The nominal control strategy will be derived via differentiation of (2.8) (i.e., a volume integral), so a few mathematical preliminaries are required.

Recall the generalized transport theorem [87]:

$$\frac{d}{dt}\int\limits_{R_{(s)}} f dV = \int\limits_{R_{(s)}} \frac{\partial f}{\partial t}\, dV + \int\limits_{S_{(s)}} f \mathbf{v}_{(\mathbf{s})} \cdot \mathbf{n}\, dA \tag{2.9}$$

where f is any scalar-, vector-, or tensor-valued function of position and time, $S_{(s)}$ is the boundary of the volume $R_{(s)}$ over which f is integrated, $\mathbf{n}$ is the unit vector normal to the boundary, and $\mathbf{v}_{(\mathbf{s})}$ is the velocity of the boundary. V and A refer to volume and area respectively. Invoking (2.9) allows for differentiation of (2.8) with respect to time:

$$\dot{E}(t) = \int\limits_{\mathcal{D}} h'(C^\star - Q(t,\tilde{p}))\left(\frac{-\partial Q(t,\tilde{p})}{\partial t}\right) d\tilde{p} + \int\limits_{\partial\mathcal{D}} \big(h(C^\star - Q(t,\tilde{p}))\big)\mathbf{v}_{(\mathbf{s})} \cdot \mathbf{n}\, dA, \tag{2.10}$$

where $\partial\mathcal{D}$ is the boundary of $\mathcal{D}$. The control volume $\mathcal{D}$ is time invariant, and thus $\mathbf{v}_{(\mathbf{s})} = 0$. Further justification for differentiation under the integral sign of (2.8) is presented in Appendix A.2. (2.10) reduces to:

$$\dot{E}(t) = \int\limits_{\mathcal{D}} h'(C^\star - Q(t,\tilde{p}))\left(\frac{-\partial Q(t,\tilde{p})}{\partial t}\right) d\tilde{p}, \tag{2.11}$$

which expands to:

$$\begin{aligned}
\dot{E}(t) &= -\int\limits_{\mathcal{D}} h'(C^\star - Q(t,\tilde{p})) \sum_{i=1}^{N} S_i(\tilde{q}_i(t),\tilde{p}) d\tilde{p} \\
&= -\sum_{i=1}^{N} \int\limits_{\mathcal{D}} h'(C^\star - Q(t,\tilde{p})) S_i(\tilde{q}_i(t),\tilde{p}) d\tilde{p} \\
&= \sum_{i=1}^{N} \hat{e}_i(t),
\end{aligned} \tag{2.12}$$

where the definition of $\hat{e}_i(t)$ is implicit. (2.12) is less than or equal to zero. Therefore, (2.8) is non-increasing. In fact, $\dot{E}$ can only take zero value at some time $t^\star$ if $\forall i \in N$, $Q(t^\star,\tilde{p}) \geq C^\star\ \forall \tilde{p} \in \mathcal{D} \mid S_i(\tilde{q}_i(t^\star),\tilde{p}) > 0$. The proposed strategy is to design control laws that drive (2.12) to be increasingly negative. Taking the derivative of (2.12) with respect to time yields:

$$\ddot{E}(t) = \sum_{i=1}^{N} \dot{\hat{e}}_i(t) \tag{2.13}$$

where $\dot{\hat{e}}_i(t)$ is:

$$\dot{\hat{e}}_i(t) = -\int_{\mathcal{D}} \left(-h''(C^\star - Q(t,\tilde{p}))S_i(\tilde{q}_i(t),\tilde{p})(S_i(\tilde{q}_i(t),\tilde{p}) + h'(C^\star - Q(t,\tilde{p}))\frac{d}{dt}(S_i(\tilde{q}_i(t),\tilde{p}))\right) d\tilde{p}. \tag{2.14}$$

The accumulation of the coverage level is independent of the angle Φ_i of the sensing footprint $\mathcal{S}_i$ assuming that the centerline of the spherical sector is aligned with the $\hat{x}_{\mathcal{B}_i}$ axis. Expanding $\frac{d}{dt}(S_i(\tilde{q}_i(t),\tilde{p}))$ yields:

$$\begin{aligned}
&\frac{d}{dt}(S_i(\tilde{q}_i(t),\tilde{p})) = \frac{\partial S_i}{\partial x_i}\dot{x}_i(t) + \frac{\partial S_i}{\partial y_i}\dot{y}_i(t) + \frac{\partial S_i}{\partial z_i}\dot{z}_i(t) + \frac{\partial S_i}{\partial \Psi_i}\dot{\Psi}_i(t) + \frac{\partial S_i}{\partial \Theta_i}\dot{\Theta}_i(t) \\
&= \left(\frac{\partial S_i}{\partial x_i}\cos\Theta\cos\Psi + \frac{\partial S_i}{\partial y_i}\cos\Theta\sin\Psi - \frac{\partial S_i}{\partial z_i}\sin\Theta\right)u_i(t) \\
&+ \left(\frac{\partial S_i}{\partial x_i}(\sin\Phi\sin\Theta\cos\Psi - \cos\Phi\sin\Psi) + \frac{\partial S_i}{\partial y_i}(\sin\Phi\sin\Theta\sin\Psi + \cos\Phi\cos\Psi) + \frac{\partial S_i}{\partial z_i}\sin\Phi\cos\Theta\right)v_i(t) \\
&+ \left(\frac{\partial S_i}{\partial x_i}(\cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi) + \frac{\partial S_i}{\partial y_i}(\cos\Phi\sin\Theta\sin\Psi - \sin\Phi\cos\Psi) + \frac{\partial S_i}{\partial z_i}\cos\Phi\cos\Theta\right)w_i(t) \\
&+ \left(\frac{\partial S_i}{\partial \Psi_i}\sin\Phi\sec\Theta + \frac{\partial S_i}{\partial \Theta_i}\cos\Phi\right)r_i(t) + \left(\frac{\partial S_i}{\partial \Psi_i}\cos\Phi\sec\Theta - \frac{\partial S_i}{\partial \Theta_i}\sin\Phi\right)s_i(t).
\end{aligned} \tag{2.15}$$

Now consider the following definitions:

$$a_{i0}(t,Q(t,\tilde{p})) = \int_{\mathcal{D}} h''(C^\star - Q(t,\tilde{p}))S_i(\tilde{q}_i(t),\tilde{p})(S_i(\tilde{q}_i(t),\tilde{p})d\tilde{p}, \tag{2.16}$$

$$a_{i1}(t,Q(t,\tilde{p})) = \int_{\mathcal{D}} h'(C^\star - Q(t,\tilde{p}))\left(\frac{\partial S_i}{\partial x_i}\cos\Theta\cos\Psi + \frac{\partial S_i}{\partial y_i}\cos\Theta\sin\Psi - \frac{\partial S_i}{\partial z_i}\sin\Theta\right)d\tilde{p}, \tag{2.17}$$

$$\begin{aligned}
&a_{i2}(t,Q(t,\tilde{p})) = \int_{\mathcal{D}} h'(C^\star - Q(t,\tilde{p})) \\
&\left(\frac{\partial S_i}{\partial x_i}(\sin\Phi\sin\Theta\cos\Psi - \cos\Phi\sin\Psi) + \frac{\partial S_i}{\partial y_i}(\sin\Phi\sin\Theta\sin\Psi + \cos\Phi\cos\Psi) + \frac{\partial S_i}{\partial z_i}\sin\Phi\cos\Theta\right)d\tilde{p},
\end{aligned} \tag{2.18}$$

$$\begin{aligned}
&a_{i3}(t,Q(t,\tilde{p})) = \int_{\mathcal{D}} h'(C^\star - Q(t,\tilde{p})) \\
&\left(\frac{\partial S_i}{\partial x_i}(\cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi) + \frac{\partial S_i}{\partial y_i}(\cos\Phi\sin\Theta\sin\Psi - \sin\Phi\cos\Psi) + \frac{\partial S_i}{\partial z_i}\cos\Phi\cos\Theta\right)d\tilde{p},
\end{aligned} \tag{2.19}$$

$$a_{i4}(t,Q(t,\tilde{p})) = \int_{\mathcal{D}} h'(C^\star - Q(t,\tilde{p}))\left(\frac{\partial S_i}{\partial \Psi_i}\sin\Phi\sec\Theta + \frac{\partial S_i}{\partial \Theta_i}\cos\Phi\right)d\tilde{p}, \tag{2.20}$$

$$a_{i5}(t,Q(t,\tilde{p})) = \int_{\mathcal{D}} h'(C^\star - Q(t,\tilde{p}))\left(\frac{\partial S_i}{\partial \Psi_i}\cos\Phi\sec\Theta - \frac{\partial S_i}{\partial \Theta_i}\sin\Phi\right)d\tilde{p}. \tag{2.21}$$

One can then rewrite (2.14) as:

$$\dot{\hat{e}}_i(t) = a_{i0}(t, Q(t,\tilde{p})) - u_i(t)a_{i1}(t, Q(t,\tilde{p})) \\ - v_i(t)a_{i2}(t, Q(t,\tilde{p})) - w_i(t)a_{i3}(t, Q(t,\tilde{p})) - r_i(t)a_{i4}(t, Q(t,\tilde{p})) - s_i(t)a_{i5}(t, Q(t,\tilde{p})). \tag{2.22}$$

Assuming zero control inputs (linear and angular velocities) in (2.22), one can see that (2.16) can be physically interpreted as the rate at which the coverage rate is reducing due to information saturation at any fixed position and orientation of the sensing footprint, $\mathcal{S}_i$. As the footprint remains stationary, there are diminishing returns on the value of newly acquired information. Thus, the additional terms in (2.22) allow for the coverage rate to be increased by mobilizing the sensor. One strategy is to define the following control law:

$$u_i{}^{loc} = k_u^{loc}\ a_{i1}(t, Q(t,\tilde{p})), \tag{2.23a}$$
$$v_i{}^{loc} = k_v^{loc}\ a_{i2}(t, Q(t,\tilde{p})), \tag{2.23b}$$
$$w_i{}^{loc} = k_w^{loc}\ a_{i3}(t, Q(t,\tilde{p})), \tag{2.23c}$$
$$s_i{}^{loc} = k_s^{loc}\ a_{i5}(t, Q(t,\tilde{p})), \tag{2.23d}$$

which reduces (2.22) with the intent of driving $\dot{E}\,(t)$ towards increasingly negative values. (2.23a)—(2.23d) tends to drive i to increase the volume of uncovered space intersecting $\mathcal{S}_i$ at any given time. This strategy alone cannot guarantee that the global coverage error, $E(t)$, will converge to zero. This guarantee is met with the introduction of additional hybrid modes and proven in section 2.2.6. Note that superscripts associated with control inputs denote the applicable operating mode: e.g., loc for local coverage, glo for global coverage etc.

2.2.2.2 Collision Avoidance Augmentation

Collision avoidance can be encoded directly into the local coverage control strategy (2.23) through a modification to the global coverage level (2.7)–a technique the authors refer to as map augmentation. Further map augmentation of (2.7) will be presented in section 2.2.2.3 to produce additional desired effects.

Consider agent i that must avoid agent j. Define the buffer distance of i from j as $R_{b,ij} = R_i + \textbf{\textit{r}}_i + \textbf{\textit{r}}_j + \varrho$, where $\varrho > 0$ can be arbitrarily small. Define the coverage map

avoidance term:

$$\acute{Q}_{ij}(t,\tilde{p}) = \begin{cases} M_{1i}(p_j,\tilde{p}), & \text{if } \|p_i - p_j\| \leq R_i; \\ 0, & \text{otherwise,} \end{cases} \tag{2.24}$$

where:

$$M_{1i}(p_j,\tilde{p}) = \begin{cases} C^\star, & \text{if } \tilde{p} \in \bar{B}_{R_{b,ij}}(p_j); \\ 0, & \text{otherwise,} \end{cases} \tag{2.25}$$

and $\bar{B}_{R_{b,ij}}(p_j)$ is the closed ball of radius $R_{b,ij}$ centered at p_j. $\acute{Q}_{ij}(t,\tilde{p})$ augments the global coverage map for i as follows:

$$\acute{Q}_i(t,\tilde{p}) = Q(t,\tilde{p}) + \sum_{j=1}^{N-1} \acute{Q}_{ij}(t,\tilde{p}). \tag{2.26}$$

$\acute{Q}_i(t,\tilde{p})$ is the avoidance augmented global coverage map for agent i. This term allows agent i to perceive a closed ball of space around j as completely covered if the agents come into close proximity. This effect is realized by substituting $\acute{Q}_i(t,\tilde{p})$ for $Q(t,\tilde{p})$ in the proposed local coverage control law (2.23):

$$\acute{u_i}^{loc} = k_u^{loc}\, a_{i1}(t,\acute{Q}_i(t,\tilde{p})), \tag{2.27a}$$
$$\acute{v_i}^{loc} = k_v^{loc}\, a_{i2}(t,\acute{Q}_i(t,\tilde{p})), \tag{2.27b}$$
$$\acute{w_i}^{loc} = k_w^{loc}\, a_{i3}(t,\acute{Q}_i(t,\tilde{p})), \tag{2.27c}$$
$$\acute{s}_i^{loc} = k_s^{loc}\, a_{i5}(t,\acute{Q}_i(t,\tilde{p})), \tag{2.27d}$$

where $a_{ik}(t,\acute{Q}_i(t,\tilde{p}))$, $\forall k \in \{1,2,3,5\}$, are functions of $\acute{Q}_i(t,\tilde{p})$ rather than $Q(t,\tilde{p})$.

Theorem 1. If both agents are operating under control law (2.27), agent i can not collide with agent j.

Proof. Assume that $\|p_i - p_j\|$ approaches $(r_i + r_j)$. After the time at which $\|p_i - p_j\| \leq R_i$, we have that:

$$\acute{Q}_i(t,\tilde{p}) \geq C^\star,\ \forall \tilde{p} \in \bar{B}_{R_{b,ij}}(p_j),$$

due to the effect of (2.24)—(2.26). In the limit that $\|p_i - p_j\| \to (\text{r}_i + \text{r}_j)$, it follows that:

$$\acute{Q}_i(t, \tilde{p}) \geq C^\star, \ \forall \tilde{p} \in \bar{B}_{R_i}(p_i),$$

due to the fact that $\bar{B}_{R_i}(p_i) \subset \bar{B}_{R_{b,ij}}(p_j)$ in this limit. Noting that the partial derivatives of $S_i\left(\tilde{q}_i(t), \tilde{p}\right)$ in (2.17)—(2.19) and (2.21) take nonzero values only within $\bar{B}_{R_i}(p_i)$, it follows that $h'\left(C^\star - \acute{Q}_i\left(t, \tilde{p}\right)\right) = 0$ for all points $\tilde{p}$ such that the partial derivatives of $S_i\left(\tilde{q}_i(t), \tilde{p}\right)$ are nonzero. As all additive terms in (2.17)—(2.19) and (2.21) are scaled by partial derivatives of $S_i\left(\tilde{q}_i(t), \tilde{p}\right)$, it follows that (2.17)—(2.19) and (2.21), with $\acute{Q}_i(t, \tilde{p})$ having been substituted for $Q(t, \tilde{p})$, tend to zero as $\|p_i - p_j\| \to (\text{r}_i + \text{r}_j)$. Thus, (2.27) tend to zero in this limit implying that agent i comes to rest (as does agent j following the same logic). Hence, agent i and j avoid collision. This concludes the proof. □

The above result implies that the agents come to rest adjacent to one another. Staying indefinitely at rest can only occur if the agents' local space, i.e., $\tilde{p} \in \mathcal{S}_i$, has reached complete coverage. In that case, the agents would switch into global coverage mode and make use of the global coverage collision avoidance technique presented in section 2.2.3.

2.2.2.3 Energy-aware Domain Partitioning

$\acute{Q}_i(t, \tilde{p})$ may be further augmented to produce a local coverage strategy consistent with the power-constrained requirements of section 2.2.1.2. This is accomplished through a novel partitioning of the hemisphere $\mathcal{D}$ with radius $\bar{R}_\mathcal{D}$ into concentric hemispherical shell subdomains, $\mathcal{D}_\ell$, $\forall \ell \in \{1, ..., N-1\}$, around a centralized hemispherical subdomain, $\mathcal{D}_{i_{cr}}$ of radius $\bar{R}_{cr}$. Each of the N agents shall be assigned to monitor one of the N subdomains. This is illustrated in Fig. 2.5. This choice of partitioning lends itself to the construction of energy-aware safety guarantees in that agents can be sorted by their remaining power supplies and subsequently assigned to cover regions of the domain sorted by distance from the charging station.

Recall the cutoff voltage, V^{co}, as the minimum voltage for which an agent can fly reliably. In practice, this should be chosen as the minimum voltage that allows for the vehicle to transition from a hovering state to the charging station through a stable landing sequence. This may be determined experimentally (or predicted online using techniques such as Bayesian inference in [88] and machine learning in [89]) and adjusted depending upon how large of a buffer the user desires.

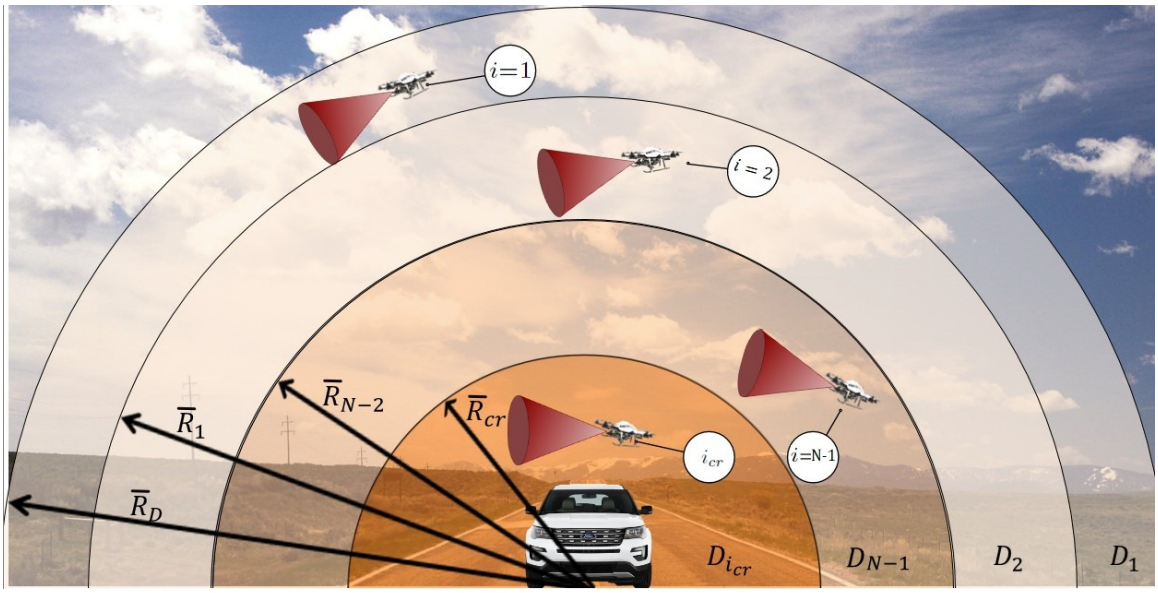


Figure 2.5: In the illustrated application, a team of quadrotors patrols the airspace surrounding a ground vehicle charging station. The most power constrained agent, i_{cr} occupies an inner hemisphere $\mathcal{D}_{i_{cr}}$ while the remaining $N-1$ agents occupy subdomains $\mathcal{D}_\ell$, $\forall \ell \in \{1, ..., N-1\}$.

At the time instant t, define the power-critical agent with index i_{cr} where:

$$i_{cr} = \arg\min_i \left(V_i(t) - V^{co}\right). \tag{2.28}$$

The remaining indices i_ℓ, $\forall \ell \in \{1, ..., N-1\}$ are chosen to satisfy:

$$V_{i_1}(t) > V_{i_2}(t) > ... > V_{i_{N-1}}(t).$$

Define the energy-aware augmented global coverage level for agent i_{cr} as:

$$\acute{Q}_{i_{cr}}(t, \tilde{p}) = \acute{Q}_{i_{cr}}(t, \tilde{p}) + M_{2i_{cr}}(t, \tilde{p}) \tag{2.29}$$

where:

$$M_{2i_{cr}}(t, \tilde{p}) = \begin{cases} 0, & \text{if } \tilde{p} \in \bar{B}_{\bar{R}_{cr}}(0); \\ C^\star, & \text{otherwise,} \end{cases} \tag{2.30}$$

and $\bar{R}_{cr}$, as illustrated in Fig. 2.5, is the operating range of i_{cr} defined as:

$$\bar{R}_{cr} = \min\left(\left(\frac{1}{N}\bar{R}_{\mathcal{D}}^3\right)^{\frac{1}{3}}, \kappa_{i_{cr}}\left(V_{i_{cr}} - V^{co}\right)\right), \tag{2.31}$$

where $\kappa_{i_{cr}}$ is a tuning parameter. Increasing $\kappa_{i_{cr}}$ has the effect of prolonging the time which i_{cr} spends away from $\mathcal{O}$ which in turn requires a faster terminal rate of convergence as $V_{i_{cr}}(t) \to V^{co}$. Note that (2.31) is upper-bounded such that a hemisphere of radius $\bar{R}_{cr}$ would have a maximum possible volume of $\frac{1}{N}$ the volume of $\mathcal{D}$.

The effect above (2.29) is to encode that agent i_{cr} shall only explore uncovered space within $\mathcal{D}_{i_{cr}}$. As $V_{i_{cr}}$ converges to V^{co}, the domain of interest of i_{cr} reduces to the origin. Thus, so long as i_{cr} is capable of entering $\mathcal{D}_{i_{cr}}$ at $t^\star \leq t_{i_{cr}}^{co}$ where $V_{i_{cr}}(t_{i_{cr}}^{co}) = V^{co}$ and remaining within $\mathcal{D}_{i_{cr}}$ until $t_{i_{cr}}^{co}$, then $p_{i_{cr}} \to \mathcal{O}$ as $t \to t_{i_{cr}}^{co}$. The agent then holds at $\mathcal{O}$ until redeployment at $t = t_{i_{cr}}^{co} + T_{i_{cr}}^{w}$ when the agent's voltage is restored to V_0. The ability to remain within $\mathcal{D}_{i_{cr}}$ is guaranteed by Theorem 2 while the ability to enter $\mathcal{D}_{i_{cr}}$ at or before $t_{i_{cr}}^{co}$ is guaranteed through Theorem 4 in section 2.2.2.4.

Theorem 2. Assume that $|\frac{dV_{i_{cr}}(t)}{dt}| \leq |\frac{dV_{i_{cr}}(t)}{dt}|_{t=t_{i_{cr}}^{co}}$, $\forall t \in [t^\star, t_{i_{cr}}^{co}]$ and that $p_{i_{cr}}(t^\star) \in \mathcal{D}_{i_{cr}}$. If $\kappa_{i_{cr}}$ is chosen such that $\kappa_{i_{cr}} \leq \frac{U_{max}}{|\frac{dV_{i_{cr}}(t)}{dt}|_{t=t_{i_{cr}}^{co}}}$, where U_{max} is the maximum achievable linear velocity of the slowest agent in the network, then there exists a control strategy such that $p_{i_{cr}}(t) \in \mathcal{D}_{i_{cr}}$ for $t \in \left[t^\star, t_{i_{cr}}^{co}\right]$.

Proof. Given that $p_{i_{cr}}(t^\star) \in \mathcal{D}_{i_{cr}}$, one must demonstrate that $\bar{R}_{cr}$ shrinks at a rate less than or equal to U_{max}. Differentiate $\bar{R}_{cr}$:

$$\dot{\bar{R}}_{cr} = \begin{cases} 0, & \text{if } \kappa_{i_{cr}}\left(V_{i_{cr}} - V^{co}\right) > \left(\frac{1}{N}\bar{R}_D^3\right)^{\frac{1}{3}}; \\ \kappa_{i_{cr}}\frac{dV_{i_{cr}}(t)}{dt}, & \text{otherwise,} \end{cases}, \tag{2.32}$$

and note that this expression conveys the rate at which $\partial\mathcal{D}_{i_{cr}} \to \mathcal{O}$. For the model obtained through experimental trials, the magnitude of $\dot{V}_{i_{cr}}(t)$ tends towards a global maximum when $V_{i_{cr}}(t) \to V^{co}$. Under this assumption, one may place an upper bound on the magnitude of (2.32): $|\dot{\bar{R}}_{cr}| \leq \kappa_{i_{cr}}|\frac{dV_{i_{cr}}(t)}{dt}| \leq \kappa_{i_{cr}}|\frac{dV_{i_{cr}}(t)}{dt}|_{t=t_{i_{cr}}^{co}}$ in terms of $\kappa_{i_{cr}}$. Choosing:

$$\kappa_{i_{cr}} \leq \frac{U_{max}}{|\frac{dV_{i_{cr}}(t)}{dt}|_{t=t_{i_{cr}}^{co}}}, \tag{2.33}$$

implies that $U_{max} \geq |\dot{\bar{R}}_{cr}|$, i.e., U_{max} is greater than or equal to the rate at which $\partial\mathcal{D}_{i_{cr}} \to \mathcal{O}$. Thus, assuming that $p_{i_{cr}}(t^\star) \in \mathcal{D}_{i_{cr}}$, there exists a control strategy to guarantee that

$p_{i_{cr}}(t) \in \mathcal{D}_{i_{cr}}$, $\forall t \in \left[t^{\star}, t^{co}_{i_{cr}}\right]$. This concludes the proof. □

For the remaining $N-1$ agents, the goal is to isolate each to their own concentric hemispherical shell subdomain, denoted $\mathcal{D}_\ell$ where $\ell \in \{1, ..., N-1\}$. This guarantees that, at any given time, each $\tilde{p} \in \mathcal{D}$ is within the subdomain of one agent. This is accomplished by assigning $\mathcal{D}_1$ to have outer radius $\bar{R}_{\mathcal{D}}$ and inner radius $\bar{R}_1$, $\mathcal{D}_2$ to have outer radius $\bar{R}_1$ and inner radius $\bar{R}_2$ and similarly up to $\mathcal{D}_{N-1}$ with outer radius $\bar{R}_{N-2}$ and inner radius $\bar{R}_{cr}$.

To this end, divide the volume of $\mathcal{D} \setminus \mathcal{D}_{i_{cr}}$, that is equal to:

$$V_{rem.} = \frac{1}{2}\left(\frac{4}{3}\pi\left(\bar{R}^3_{\mathcal{D}} - \bar{R}^3_{cr}\right)\right), \tag{2.34}$$

equally among each element of $\mathcal{D}_\ell$. This requires $\{\bar{R}_1, \bar{R}_2, ...\bar{R}_{N-2}\}$ to satisfy the following constraint:

$$\frac{V_{rem.}}{(N-1)} = \frac{1}{2}\left(\frac{4}{3}\pi\left(\bar{R}^3_D - \bar{R}^3_1\right)\right) = \frac{1}{2}\left(\frac{4}{3}\pi\left(\bar{R}^3_1 - \bar{R}^3_2\right)\right) = \ldots \frac{1}{2}\left(\frac{4}{3}\pi\left(\bar{R}^3_{N-2} - \bar{R}^3_{cr}\right)\right). \tag{2.35}$$

This computation is relatively straightforward as a closed form expression exists for $\bar{R}_1$:

$$\bar{R}_1 = \left(\bar{R}^3_D - \frac{\left(\bar{R}^3_D - \bar{R}^3_{cr}\right)}{(N-1)}\right)^{\frac{1}{3}} \tag{2.36}$$

Then, $\{\bar{R}_1, \bar{R}_2, ...\bar{R}_{N-2}\}$ can be found through a sequential chain of computations. The set $\bar{R}_\rho$, $\forall \rho \in \{2, ..., N-2\}$ is defined as:

$$\bar{R}_\rho = \left(\bar{R}^3_{\rho-1} - \frac{\left(\bar{R}^3_{\rho-1} - \bar{R}^3_{cr}\right)}{(N-1)}\right)^{\frac{1}{3}}. \tag{2.37}$$

The energy-aware augmented global coverage level for agent ℓ is defined similarly to (2.29):

$$\acute{Q}_\ell(t, \tilde{p}) = \acute{Q}_\ell(t, \tilde{p}) + M_{3\ell}(t, \tilde{p}) \tag{2.38}$$

where:

$$M_{3\ell}(t, \tilde{p}) = \begin{cases} 0, & \text{if } \tilde{p} \in \mathcal{D}_\ell; \\ C^{\star}, & \text{otherwise.} \end{cases} \tag{2.39}$$

One can substitute $\acute{Q}_\ell(t, \tilde{p})$ for (2.7) in (2.17)—(2.19) and (2.21) for agents $\ell \in \{1, ..., N-$

$1\}$ and similarly $\acute{Q}_{i_{cr}}(t,\tilde{p})$ for agent i_{cr} to yield the elected local coverage control strategy:

$$\acute{u_i}^{loc} = k_u^{loc}\, a_{i1}(t, \acute{Q}_i(t,\tilde{p})), \tag{2.40a}$$

$$\acute{v_i}^{loc} = k_v^{loc}\, a_{i2}(t, \acute{Q}_i(t,\tilde{p})), \tag{2.40b}$$

$$\acute{w_i}^{loc} = k_w^{loc}\, a_{i3}(t, \acute{Q}_i(t,\tilde{p})), \tag{2.40c}$$

$$\acute{s_i}^{loc} = k_s^{loc}\, a_{i5}(t, \acute{Q}_i(t,\tilde{p})), \tag{2.40d}$$

The effect of this strategy is that each agent shall only perceive space as uncovered if such space lies within the agent's assigned subdomain. When agents operate in close proximity to the boundary of their assigned subdomain, (2.40) shall tend to direct their sensing footprints towards the interior of the assigned subdomain.

2.2.2.4 An Upper Bound on N for Agent Servicing at $\mathcal{O}$

The ability of agent i_{cr} to enter $\mathcal{D}_{i_{cr}}$ at $t^\star \leq t_{i_{cr}}^{co}$ is dependent upon the number of agents in the network and the scheduling used to temporally space agent deployment. Specifically, there exists an upper bound on N for which agent $i = N-1$ is still within a safe proximity of $\mathcal{O}$ at the instant before selection as i_{cr}. Beyond this bound, the incumbent critical agent cannot transfer to $\mathcal{D}_{i_{cr}}$ before battery expiration. The purpose of this subsection is to define this limit on N and the proper deployment schedule.

Remark 3. With assumed knowledge of the lifespan of each individual agent's battery upon deployment, T_i as defined in section 2.2.1.2, as well as the waiting period T_i^w satisfying $\bar{T} = T_i + T_i^w$, the network deployment schedule operates such that landing times at $\mathcal{O}$ occur with a period of $\frac{\bar{T}}{N}$. With this schedule, it is guaranteed that any agent will have a remaining flight time of $\frac{\bar{T}}{N}$ at the instant before selection as i_{cr}. This guarantee is provided by Theorem 3.

Theorem 3. Assume that $N \geq 2$ agents are deployed one at a time with the deployment schedule and assumptions of Remark 3. Then, agent i will have a remaining flight time of $\frac{\bar{T}}{N}$ at the instant before selection as i_{cr}.

Proof. Consider that N agents are deployed at the following times:

$$\begin{aligned} t_N^d &= T_N^w \\ t_{N-1}^d &= \frac{\bar{T}}{N} + T_{N-1}^w \\ t_{N-2}^d &= \frac{2\bar{T}}{N} + T_{N-2}^w \\ &\vdots \\ t_1^d &= \frac{(N-1)\,\bar{T}}{N} + T_1^w \end{aligned}$$

From the definition $T_i = t_i^{co} - t_i^d$, it follows that $t_i^{co} = t_i^d + T_i$. Thus the cutoff times of the N agents are:

$$\begin{aligned} t_N^{co} &= T_N^w + T_N = \bar{T} \\ t_{N-1}^{co} &= \frac{\bar{T}}{N} + T_{N-1}^w + T_{N-1} = \frac{(N+1)\,\bar{T}}{N} \\ t_{N-2}^{co} &= \frac{2\bar{T}}{N} + T_{N-2}^w + T_{N-2} = \frac{(N+2)\,\bar{T}}{N} \\ &\vdots \\ t_1^{co} &= \frac{(N-1)\,\bar{T}}{N} + T_1^w + T_1 = \frac{(2N-1)\,\bar{T}}{N} \end{aligned}$$

where the right hand simplification results from the fact that $T_i + T_i^w = \bar{T}$. Thus, it follows that $t_i^{co} - t_{i-1}^{co} = \frac{\bar{T}}{N}, \forall i \in \{1, ..., N-1\}$. It is straightforward to show that this holds for repeated deployments upon index wrap-around. This concludes the proof. □

To simplify notation, define $f = \frac{U_{max}\bar{T}}{R_D}$. There exists an analytic expression for the bound on N in accordance with the conditions of this subsection.

Theorem 4. Assume that $N \geq 2$ agents are deployed one at a time with the deployment schedule and assumptions of Remark 3. Assume also that i_{cr} is in $\mathcal{D}_{N-1}$ at the instant of selection as the power-critical agent. Then:

$$N \leq \frac{\left(\frac{2}{3}\right)^{\frac{1}{3}} f^3}{\left(3^{\frac{1}{2}}\left(27f^6 - 4f^9\right)^{\frac{1}{2}} - 9f^3\right)^{\frac{1}{3}}} + \frac{\left(3^{\frac{1}{2}}\left(27f^6 - 4f^9\right)^{\frac{1}{2}} - 9f^3\right)^{\frac{1}{3}}}{2^{\frac{1}{3}}3^{\frac{2}{3}}} \tag{2.41}$$

is a sufficient condition such that i_{cr} enters $\mathcal{D}_{i_{cr}}$ at $t^\star \leq t_{i_{cr}}^{co}$.

Proof. Denote $g = \left(3^{\frac{1}{2}}\left(27f^6 - 4f^9\right)^{\frac{1}{2}} - 9f^3\right)^{\frac{1}{3}}$ and substitute into (2.41) to obtain:

$$N \leq \frac{\left(\frac{2}{3}\right)^{\frac{1}{3}} f^3}{g} + \frac{g}{2^{\frac{1}{3}}3^{\frac{2}{3}}}.$$

Now find a common denominator and combine terms:

$$N \leq \frac{\left(2f^3 3^{\frac{1}{3}} + g^2 2^{\frac{1}{3}}\right) 6^{\frac{1}{3}}}{6g}. \tag{2.42}$$

Cubing this expression yields:

$$N^3 \leq \frac{\left(2f^3 3^{\frac{1}{3}} + g^2 2^{\frac{1}{3}}\right)^3}{36g^3} \tag{2.43}$$

and subtracting 1 from each side of (2.42) yields:

$$N - 1 \leq \frac{\left(2f^3 3^{\frac{1}{3}} + g^2 2^{\frac{1}{3}}\right) 6^{\frac{1}{3}} - 6g}{6g}. \tag{2.44}$$

The assumption that $N \geq 2$ produces a special case for which one may draw the following conclusion from (2.43) and (2.44):

$$\frac{N^3}{N-1} \leq \frac{\left(2f^3 3^{\frac{1}{3}} + g^2 2^{\frac{1}{3}}\right)^3}{36g^3\left(\frac{\left(2f^3 3^{\frac{1}{3}} + g^2 2^{\frac{1}{3}}\right) 6^{\frac{1}{3}} - 6g}{6g}\right)}.$$

A proof of this special case is presented in Appendix A.3. Expanding the numerator and denominator yields:

$$\frac{N^3}{N-1} \leq \frac{24f^9 + 12f^6 g^2 3^{\frac{2}{3}} 2^{\frac{1}{3}} + 6f^3 g^4 3^{\frac{1}{3}} 2^{\frac{2}{3}} + 2g^6}{12f^3 g^2 3^{\frac{2}{3}} 2^{\frac{1}{3}} + 6g^4 3^{\frac{1}{3}} 2^{\frac{2}{3}} - 36g^3}$$

$$= f^3\left(\frac{24f^6 + 12f^3 g^2 3^{\frac{2}{3}} 2^{\frac{1}{3}} + 6g^4 3^{\frac{1}{3}} 2^{\frac{2}{3}} + \frac{2g^6}{f^3}}{12f^3 g^2 3^{\frac{2}{3}} 2^{\frac{1}{3}} + 6g^4 3^{\frac{1}{3}} 2^{\frac{2}{3}} - 36g^3}\right) \tag{2.45}$$

Noting common terms in the numerator and denominator, one may show that the term in parenthesis in (2.45) is equal to 1 if $24f^6 + \frac{2g^6}{f^3} = -36g^3$. Expanding either the left hand

or the right hand side of this equation and substituting in the definition for g yields:

$$324f^3 - 36\,(3)^{\frac{1}{2}}\left(27 - 4f^3\right)^{\frac{1}{2}} f^3.$$

Therefore, (2.45) reduces to:

$$\frac{N^3}{N-1} \leq f^3$$

whose cubic root is:

$$\frac{N}{(N-1)^{\frac{1}{3}}} \leq \frac{U_{max}\bar{T}}{\bar{R}_{\mathcal{D}}}. \tag{2.46}$$

$\bar{R}_{\mathcal{D}}$ may be expressed in terms of $\bar{R}_{N-2}$ in the limit that $\partial\mathcal{D}_{i_{cr}} \to \mathcal{O}$. Substituting $\bar{R}_{cr} = 0$ into (2.34) yields:

$$V_{rem} = \frac{2}{3}\pi\bar{R}_{\mathcal{D}}^3,$$

which may be substituted into (2.35) to yield:

$$\frac{\frac{2}{3}\pi\bar{R}_{\mathcal{D}}^3}{N-1} = \frac{2}{3}\pi\bar{R}_{N-2}^3.$$

Removing common terms and taking the cubic root of each side yields:

$$\frac{\bar{R}_{\mathcal{D}}}{(N-1)^{\frac{1}{3}}} = \bar{R}_{N-2}.$$

which may be rearranged in terms of $\bar{R}_{\mathcal{D}}$ and substituted into (2.46):

$$N \leq \frac{U_{max}\bar{T}}{\bar{R}_{N-2}}$$

which is equivalent to:

$$\bar{R}_{N-2} \leq \frac{U_{max}\bar{T}}{N}. \tag{2.47}$$

As $\bar{R}_{N-2}$ is an upper bound on the agent's distance from $\mathcal{O}$ while in $\mathcal{D}_{N-1}$, it follows that the agent is no further away from the origin than the maximum velocity of the slowest agent multiplied by the deployment period. It is intuitive that (2.47) is a sufficient condition that i_{cr} can enter $\mathcal{D}_{i_{cr}}$ before $t_{i_{cr}}^{co}$. Upon selection as i_{cr}, the agent will enter Subdomain

Transfer Mode (defined in section 2.2.4) and fly directly towards the origin with a velocity of U_{max} until it enters $\mathcal{D}_{i_{cr}}$. Under the deployment period assumption, the agent will have a minimum remaining flight time of $\frac{\bar{T}}{N}$. In the worst case scenario, the agent is at a distance $\bar{R}_{N-2}$ from the origin at the instant before selection and enters $\mathcal{D}_{i_{cr}}$ at the instant that $V_{i_{cr}} = V^{co}$. This concludes the proof. □

Note that Theorem 4 is built upon the assumption that the power-critical agent is inside of $\mathcal{D}_{N-1}$ when selected by (2.28). Subdomain Transfer Mode (defined in section 2.2.4) commands each agent to transfer to its inner adjacent shell once per $\frac{\bar{T}}{N}$ time units. Considering that each shell is thinner than its adjacent inner shell ($\bar{R}_{\mathcal{D}} > \bar{R}_1 > ... > \bar{R}_{cr} \implies \bar{R}_{N-2} - \bar{R}_{cr} > \bar{R}_{N-3} - \bar{R}_{N-2} > ... > \bar{R}_{\mathcal{D}} - \bar{R}_1$ under the constraint of (2.35)), one can see that the assumption is met recursively so long as $i = 1$ is within $\mathcal{D}$.

To illustrate the bound of Theorem 4, consider that $U_{max} = 1$ m/s, $\bar{T} = 1000$ seconds, $\bar{R}_{\mathcal{D}} = 100$ meters, and thus $f = 10$. A numerical evaluation of (2.41) yields an upper theoretical limit of 31 agents.

2.2.3 Energy-aware Global Coverage Mode

The convergence of the global coverage error (2.8) to zero is accomplished via the Global Coverage Mode. This mode is initiated for agent i when:

$$|\hat{e}_i(t)| < \varepsilon_1. \tag{2.48}$$

that is, agent i's contribution to the rate of change of (2.8) has dropped below a predefined threshold ε_1. To apply the global coverage strategy, $\mathcal{D}$ is divided into a grid of N_c cubes $\mathcal{D}^j$, $\forall j \in \{1, ..., N_c\}$. Each $\mathcal{D}^j$ is of dimension $\sqrt{2}R_i \times \sqrt{2}R_i \times \sqrt{2}R_i$, and the possible waypoints W^j are placed at the centroid of each $\mathcal{D}^j$.These dimensions are chosen such $\mathcal{D}^j$ is inscribed within the sphere of possible orientations for $\mathcal{S}_i$. N_c shall be large enough such that the grid over-approximates the hemispherical $\mathcal{D}$. Elements of $\mathcal{D}^j$ that intersect $\partial\mathcal{D}$ shall be truncated along $\partial\mathcal{D}$ with their associated elements of W_j shifted to the new centroid.

Define the following function to encode anticipated local coverage error for agent i in each cube $\mathcal{D}^j$:

$$\hat{C_i^j}(t_{gc}) = \int_{\mathcal{D}^j} \min\{C^\star, \acute{Q}_i(t_{gc}, \tilde{p})\} d\tilde{p} + \delta \left(C^\star V_{\mathcal{D}^j} - \int_{\mathcal{D}^j} \min\{C^\star, \acute{Q}_i(t_{gc}, \tilde{p})\} d\tilde{p} \right), \tag{2.49}$$

where $V_{\mathcal{D}^j}$ is the volume of each $\mathcal{D}^j$ and t_{gc} is the time at which global coverage mode was initiated. (2.49) is the coverage weight of each cube $\mathcal{D}^j$. Note that $\hat{C_i^j}(t_{gc}) \to \infty$ as $\acute{Q}_i(t_{gc}, \tilde{p}) \geq C^\star, \forall \tilde{p} \in \mathcal{D}^j$. This is accomplished via the Dirac delta function, in the second term of the right-hand side of (2.49), which ensures that completely covered cubes shall not be selected as waypoints. The arguments of the minimum functions prevent discrete points covered well beyond $C^\star$ from carrying an unwarranted weight. Such a scenario may result in patches of gross coverage levels producing cube coverage levels in excess of $C^\star V_{\mathcal{D}^j}$ while discrete points $\tilde{p} \in \mathcal{D}^j$ remain uncovered.

Power conservation is encoded into the global coverage strategy by considering the relative distances of the waypoints W^j from the current location of the agent i. Define the actuation accumulation function:

$$\Gamma_i(t_{gc}) = \int_0^{t_{gc}} \gamma_{i1}|u_i(\tau)| + \gamma_{i2}|v_i(\tau)| + \gamma_{i3}|w_i(\tau)| + \gamma_{i4}|s_i(\tau)|d\tau \tag{2.50}$$

which represents the net accumulated actuator effort since deployment. $\gamma_{ik} > 0$, where $k \in \{1, ..., 4\}$ are cost weights that can be assigned based upon a particular system's rate of energy usage. (2.49) and (2.50) can be combined into the overall cost definition:

$$\tilde{j} = \arg\min_j \left((\Lambda_{i0} + \Gamma_i(t_{gc})) \|p_i - W^j\| + \Lambda_{i1} \hat{C_i^j} \right), \tag{2.51a}$$

$$p_{\star,i} = W^{\tilde{j}}, \tag{2.51b}$$

where the chosen waypoint, $p_{\star,i}$, for agent i is assigned to be the centroid of the cube indexed by $\tilde{j}$, that minimizes the cost function (2.51a). $\Lambda_{i0} > 0$ and $\Lambda_{i1} > 0$ are cost weights.

Note that the cost function utilizes a static weight, Λ_{i0}, and a dynamic weight, $\Gamma_i(t_{gc})$, to penalize the selection of waypoints that are spatially distant from the agent's current position. $\Gamma_i(t_{gc})$ grows in time as a function of agent i's actuation effort. This cost is weighted against the relative coverage level of the cube. In real world implementation, the value of Λ_{i0} would be assigned upon deployment as a function of agent i's remaining battery life. This offers energy conserving flexibility to scenarios in which a multi-agent sensing network may be deployed with short notice and disparate battery charge levels. Additionally, γ_{ik} can be increased for individual agents that consume energy at a faster rate. These weights allow for agents with a history of less actuation effort to select more distant waypoints, thus reducing the need for persistent spatial relocation of the more power-constrained agents.

Upon selecting the waypoint, the agent must determine the optimal orientation to as-

sume at the destination. This orientation will be the one for which $\mathcal{S}_i$ is exposed to the least covered subspace of $\mathcal{D}$. This can be expressed as finding the value of Ψ_i that solves the following optimization problem:

$$\Psi_{\star,i} = \arg\max_{\Psi} \left(\int_{\mathcal{D}} h' \left(C^{\star} - \acute{Q}_i(t_{gc}, \tilde{p}) \right) \left(S_i\big(\tilde{q}_{\star,i}, \tilde{p}\big) \right) d\tilde{p} \right), \tag{2.52}$$

where $\tilde{q}_{\star,i} = \left[{p_{\star,i}}^T \, 0 \, 0 \, \Psi_{\star,i} \right]^T$. Selecting a destination orientation with the minimum coverage level maximizes (2.12), and consequently, the initial coverage rate when local coverage mode is resumed.

Convergence to the desired position, $p_{\star,i}$, as well as collision avoidance are achieved using the following control scheme that was earlier developed in [90] and later in [91]. Define a nominal Lyapunov-like function of the form:

$$\mathcal{V}_{i0} = \|p_i - p_{\star,i}\|^2, \tag{2.53}$$

which is positive definite and encodes convergence of agent i to the destination $p_{\star,i}$. Consider the global avoidance constraint for each of the $j \neq i$ friendly agents:

$$c_{4ij} = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 - (\mathscr{r}_i + \mathscr{r}_j)^2 > 0,$$

to encode that collision avoidance requires $\|p_i - p_j\| > \mathscr{r}_i + \mathscr{r}_j$. Define a logarithmic barrier function in terms of this constraint:

$$b_{ij}(p_i, p_j) = -\ln(c_{4ij}(p_i, p_j)),$$

and define, as in [92], the recentered barrier function:

$$q_{ij}(p_i, p_j) = b_{ij}(p_i, p_j) - b_{ij}(p_{\star,i}, p_j), \tag{2.54}$$

which tends to $+\infty$ as $\|p_i - p_j\| \to \mathscr{r}_i + \mathscr{r}_j$ and zero as $p_i \to p_{\star,i}$. To ensure that the Lyapunov-like function is non-negative everywhere, define:

$$\mathcal{V}'_{ij}(p_i, p_j) = (q_{ij}(p_i, p_j))^2.$$

The global avoidance constraint is activated under the condition that $\|p_i - p_j\| \leq R_i$ using a bump function σ_{ij}, which is twice continuously differentiable with respect to the inter-agent

distance:

$$\sigma_{ij} = \begin{cases} 1 & \text{if } \textit{r}_i + \textit{r}_j \leq d_{ij} \leq R_z \\ Ad_{ij}^3 + Bd_{ij}^2 + Cd_{ij} + D & \text{if } R_z < d_{ij} < R_i \\ 0 & \text{if } d_{ij} \geq R_i \end{cases}$$

where $d_{ij} = \|p_i - p_j\|$. R_z determines the steepness of the bump and is chosen such that $\textit{r}_i + \textit{r}_j < R_z < R_i$. The coefficients are defined as follows: $A = -(2/(R_z - R_i)^3)$, $B = (3(R_z+R_i)/(R_z-R_i)^3)$, $C = (-6R_zR_i/(R_z-R_i)^3)$, and $D = (R_i^2(3R_z-R_i)/(R_z-R_i)^3)$. Therefore, the global avoidance constituent Lyapunov-like function is:

$$\mathcal{V}_{ij}(p_i, p_j) = \sigma_{ij}\mathcal{V}'_{ij}(p_i, p_j),$$

which augments the nominal Lyapunov-like function as:

$$\nu_i = \mathcal{V}_{i0} + \sum_{j=1}^{N-1} \mathcal{V}_{ij}$$

and can be scaled between zero and one to form a candidate Lyapunov-like function for the system governing the position trajectories $p_i(t)$ of agent i:

$$\mathcal{V}_i = \frac{\nu_i}{1 + \nu_i}. \tag{2.55}$$

As discussed in [91], trajectories that follow the negative gradient of $\mathcal{V}_i$ are almost globally asymptotically stabilizing, i.e., asymptotically reach the destination from almost all initial conditions, except for a set of initial conditions of measure zero. The controller is defined as:

$$\begin{bmatrix} u_i^{glo} \\ v_i^{glo} \\ w_i^{glo} \end{bmatrix} = \mathcal{R}_1^{-1} \begin{bmatrix} \frac{-\partial \mathcal{V}_i}{\partial x_i} \left(\frac{U_{max}}{\sqrt{\left(\frac{\partial \mathcal{V}_i}{\partial x_i}\right)^2 + \left(\frac{\partial \mathcal{V}_i}{\partial y_i}\right)^2 + \left(\frac{\partial \mathcal{V}_i}{\partial z_i}\right)^2}} \right) \\ \frac{-\partial \mathcal{V}_i}{\partial y_i} \left(\frac{U_{max}}{\sqrt{\left(\frac{\partial \mathcal{V}_i}{\partial x_i}\right)^2 + \left(\frac{\partial \mathcal{V}_i}{\partial y_i}\right)^2 + \left(\frac{\partial \mathcal{V}_i}{\partial z_i}\right)^2}} \right) \\ \frac{-\partial \mathcal{V}_i}{\partial z_i} \left(\frac{U_{max}}{\sqrt{\left(\frac{\partial \mathcal{V}_i}{\partial x_i}\right)^2 + \left(\frac{\partial \mathcal{V}_i}{\partial y_i}\right)^2 + \left(\frac{\partial \mathcal{V}_i}{\partial z_i}\right)^2}} \right) \end{bmatrix}. \tag{2.56}$$

With the exception of local coverage, whose avoidance strategies are presented in Theorem 1, all hybrid modes controlling linear velocity are based upon waypoints and incor-

porate controllers similar to (2.56). The anomalous scenario in which multiple agents become isolated in critical points of $\mathcal{V}_i$ can be overcome by applying the high-level prioritized path planning protocol that is presented in [91]. This technique is suited for high-density operations in cluttered environments. For further analysis on collision avoidance with logarithmic barriers, please see [91] and [90].

Convergence to the desired orientation $\Psi_{\star,i}$ is achieved via the following controller:

$$s_i^{glo} = \mathcal{R}_{2_{3,3}}^{-1} \left(\Psi_{\star,i} - \Psi_i \right). \tag{2.57}$$

Thus, upon computation of (2.51b) and (2.52), control laws (2.56) and (2.57) are simultaneously activated. Upon acquisition of the desired position and orientation, a transition occurs that is consistent with the hybrid control strategy introduced in section 2.2.1.3 and rigorously defined in Appendix A.1. This transition hands control to either local coverage mode, upon satisfaction of $G(\zeta_{i1}, \zeta_{i0})$, or Waypoint Scan Mode, upon satisfaction of $G(\zeta_{i1}, \zeta_{i3})$. If at any time the agent i lies outside of the intended operating range, as consistent with $G(\zeta_{i1}, \zeta_{i2})$, control is handed to Subdomain Transfer Mode.

2.2.4 Subdomain Transfer Mode

Subdomain transfer mode can be entered into from any of the other modes as triggered by the equivalent conditions $G(\zeta_{ik}, \zeta_{i2})$, $\forall k \in \{0, 1, 3, 4\}$. The linear velocity control law of this mode shall be denoted as: $[u_i^{sub}\, v_i^{sub}\, w_i^{sub}]^T$, and is equal to the right hand side of (2.56) with $p_{\star,i} = \mathcal{O}$. The angular velocity control law of this mode shall be denoted as s_i^{sub} and is equal to the right hand side of (2.57) with:

$$\Psi_{\star,i} = \arctan 2\left(\frac{-y_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}}, \frac{-x_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}}\right). \tag{2.58}$$

With this choice of $\Psi_{\star,i}$, agent i aims to align the $\hat{x}_{\mathcal{B}_i}$ axis, the center line of $\mathcal{S}_i$, with the xy-plane projection of $-p_i$ during subdomain transfer. This orients $\mathcal{S}_i$ to be inward looking towards the origin thus ensuring that $\mathcal{S}_i$ will be inside of the target subdomain when control is handed to local coverage mode.

2.2.5 Waypoint Scan Mode

For $G(\zeta_{i1}, \zeta_{i3})$ (as defined in Appendix A.1) to be satisfied, the cube $\mathcal{D}^j$ will contain space that has not yet been completely covered. However, $|\hat{e}_i(t)|$ (the absolute value of the derivative of the global coverage error due to the motion of agent i) is not large enough to warrant

exploration in local coverage mode. Such a situation often occurs in practice when there is uncovered space within $\mathcal{D}^j$ directly above or below the agent while $\|p_i - p_{\star,i}\| < \varepsilon_2$ is satisfied. Under these conditions, no choice of Ψ_i will bring this portion of space inside of $\mathcal{S}_i$. One solution is to design a hybrid mode in which the agents patrol a defined path within $\mathcal{D}^j$ until the cube has been fully swept over.

An intuitive solution would be for the agent to select sub-waypoints above and below $W^{\tilde{j}}$ and to yaw through one complete revolution at each sub-waypoint. However, as Fig. 2.6 (a) illustrates, this path would contain gaps in the sweep due to the spherical sector shape of $\mathcal{S}_i$. Through symmetry, it is clear that the angle of the horizontal vertices of the gaps is 2α. The sub-waypoints illustrated in Fig. 2.6 (b) are placed at these vertices to allow for the gaps to be swept over. Only the left half sweep is illustrated in order to avoid clutter. A total of 13 sub-waypoints may be positioned as shown in Fig. 2.7 relative to $W^{\tilde{j}}$ to guarantee that $\mathcal{S}_i$ sweeps over all $\tilde{p} \in \mathcal{D}^j$. This holds true even for $\tilde{p}$ outside of the cross-section of Fig. 2.6 so long as $\frac{\pi}{4} \leq \alpha < \frac{\pi}{2}$. This is verified in Theorem 5.

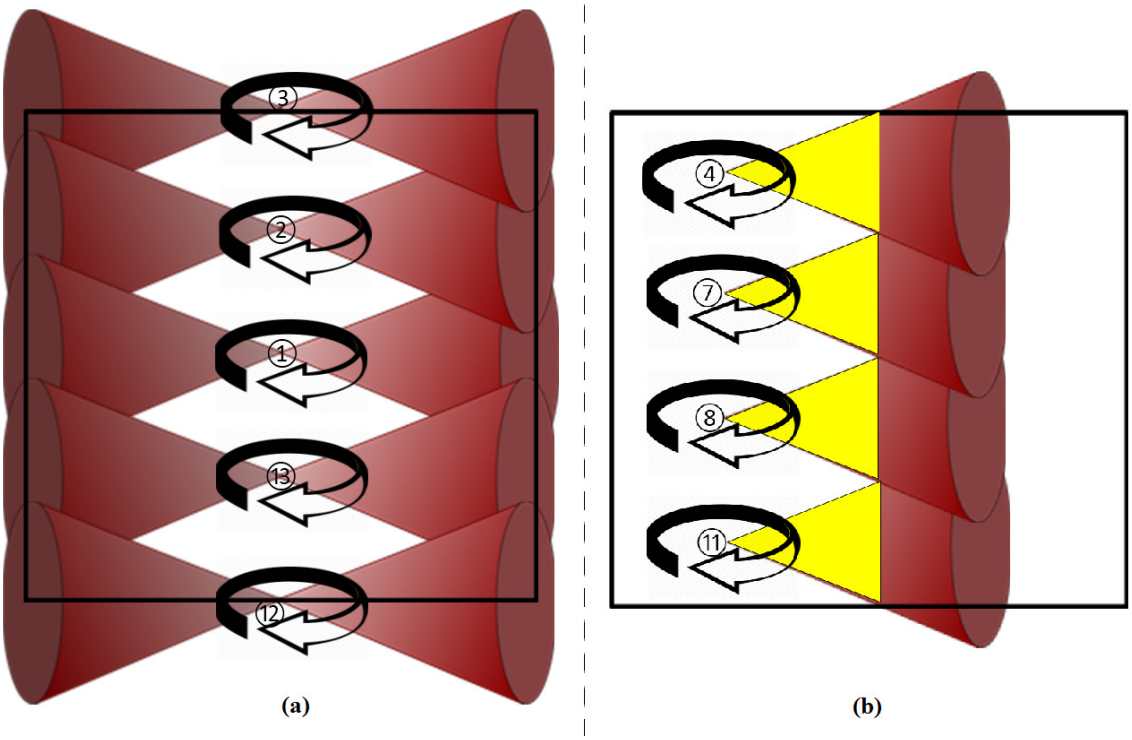


Figure 2.6: Sub-waypoints directly above or below $W^{\tilde{j}}$ will leave to coverage gaps in (a). Additional points positioned laterally guarantee coverage of the left half of the gaps in (b).

Theorem 5. If agent i yaws through a complete revolution at each of the 13 sub-waypoints illustrated in Fig. 2.7, a vertical plane bisecting $\mathcal{D}^j$, then $\mathcal{S}_i$ shall sweep over all $\tilde{p} \in \mathcal{D}^j$ so long as $\frac{\pi}{4} \leq \alpha < \frac{\pi}{2}$.

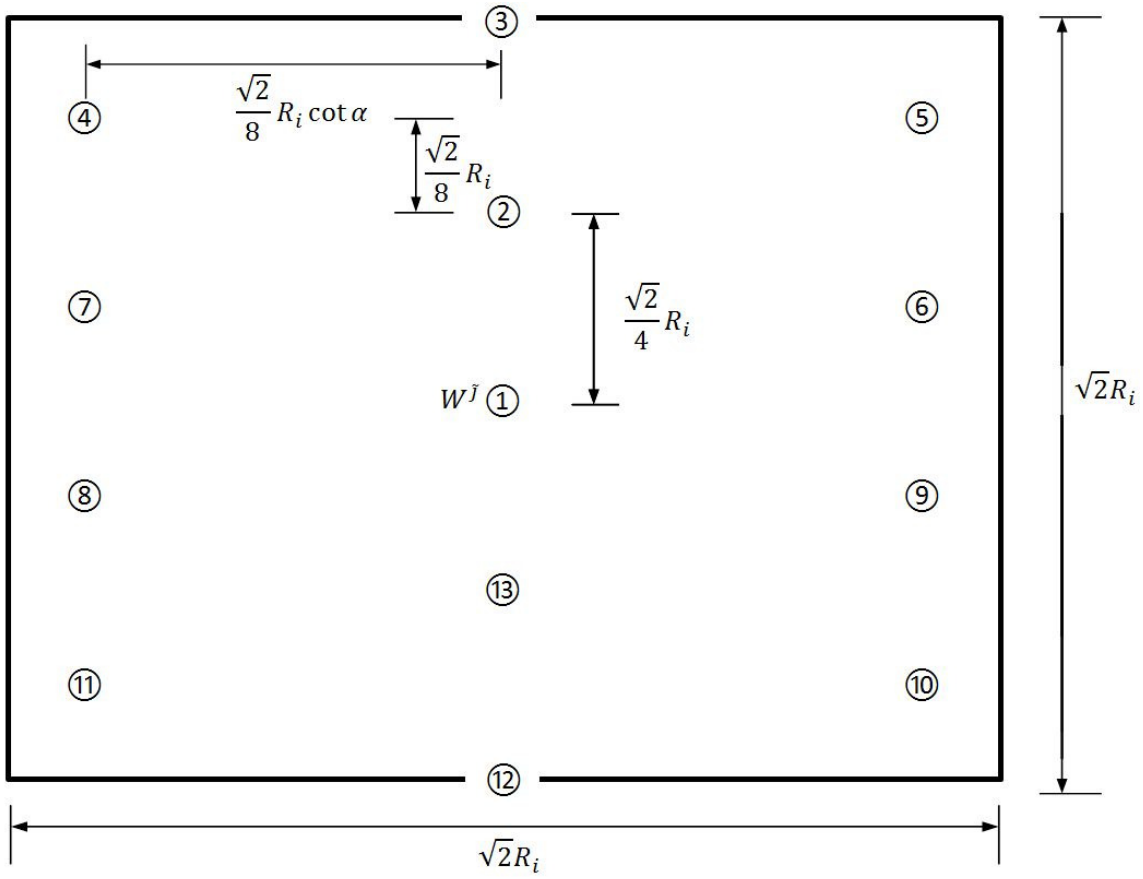


Figure 2.7: In Waypoint Scan Mode, agent i will travel between 13 sub-waypoints whose position relative to $W^{\tilde{j}}$ is defined here.

Proof. Revolutions of $\mathcal{S}_i$ about sub-waypoints above and below $W^{\tilde{j}}$ will produce four coverage gaps as illustrated in white in Fig. 2.6 (a). Each of these gaps may be represented in 3-D space by two cones, of radius $\frac{\sqrt{2}}{8}R_i \cot\alpha$ and height $\frac{\sqrt{2}}{8}R_i$, whose planar surfaces are coincident. It shall be shown that a yaw revolution of i at sub-waypoint 4 sweeps $\mathcal{S}_i$ across all $\tilde{p}$ in the left half of the top coverage gap as indicated in yellow in Fig. 2.7 (b).

This left-side gap is again indicated in yellow and projected into 3 orthogonal planes: front (a), side (b), and top (c) in Fig. 2.8. These projections shall be referenced throughout the remainder of this proof. The geometry of coverage gap projection (a), whose left-side vertex has angle 2α, may be fully contained within $\mathcal{S}_i$ at the illustrated yaw orientation so long as $\frac{\sqrt{2}}{8}R_i \cot\alpha \leq R_i$ which is equivalent to the condition that $\tan\alpha \geq \frac{\sqrt{2}}{8}$.

Gap projection (b) is obtained from the $90°$ right-handed rotation indicated in (a). The vertical height of gap projection (b) equal to that of the diameter of $\mathcal{S}_i$ within this plane by virtue of the above-stated gap vertex angle 2α. The horizontal width of gap projection (b) shall not extend outside of $\mathcal{S}_i$ so long as $\frac{\sqrt{2}}{8}R_i \cot\alpha \leq \frac{\sqrt{2}}{8}R_i$ which is equivalent to the

condition that $\tan\alpha \geq 1$.

Gap projection (c) is obtained from the 90° right-handed rotation indicated in (b) and is a semicircle. Agent i's 360° yaw motion about sub-waypoint 4 shall sweep across this semicircle so long as R_i is greater than or equal to the indicated chord length $\frac{1}{4}R_i\cot\alpha$ which is equivalent to the condition that $\tan\alpha \geq \frac{1}{4}$. The inequality constraints on $\tan\alpha$ are all satisfied when $\frac{\pi}{4} \leq \alpha < \frac{\pi}{2}$. Under this condition, a full yaw revolution of i shall sweep $\mathcal{S}_i$ over all points of the three orthonormal coverage gap projections indicated in Fig. 2.8 and thus over all points of the coverage gap. Through symmetry it may be shown that sub-waypoints 5-11 sweep over all remaining $\tilde{p}$ in coverage gaps. Thus, $\mathcal{S}_i$ sweeps over all $\tilde{p} \in \mathcal{D}^j$. This concludes the proof. □

Recall from section 2.2.1.3 that these 13 sub-waypoints are indexed by wp and note that this is to be done in a manner consistent with Fig. 2.7. The automaton defined in section 2.2.1.3 calls for an input $f(\zeta_{i3}, \tilde{q}_i)$ to transition between sub-waypoints and an input $f(\zeta_{i4}, \tilde{q}_i)$ for the 360° revolution at each sub-waypoint. $[u_i^{wps}\, v_i^{wps}\, w_i^{wps}]^T$ is equal to the right hand side of (2.56) with $p_{\star,i}$ equal to the element of the set of points defined in Fig. 2.7 with index wps. s_i^{wps} is equal to an arbitrary positive constant. $G(\zeta_{i3}, \zeta_{i4})$ provides a definition for sub-waypoint capture consistent with this work and $G(\zeta_{i4}, \zeta_{i3})$ is satisfied upon completion of one revolution–that is, an arbitrary positive s_i^{wps} command has rotated i to an orientation $\Psi_i = \Psi_i(t_{G(\zeta_{i3},\zeta_{i4})}) - d\Psi$.

2.2.6 Guarantee of Complete Coverage

Theorem 6. The control strategy defined above guarantees that $E(t) \to 0$ as $t \to t_f$.

Proof. As the protocol approaches complete coverage, $\hat{e}_i$ shall tend to maintain an absolute value less than ε_1. With $G(\zeta_{i0}, \zeta_{i1})$ and $G(\zeta_{i1}, \zeta_{i3})$ persistently satisfied for agent i operating within it's prescribed subdomain, the agent shall repeatedly trigger waypoint scan mode. The geometry of the sub-waypoints within each cube $\mathcal{D}^j$ guarantees that each $\tilde{p} \in \mathcal{D}^j$ shall remain within the interior of $\mathcal{S}_i$ for a finite interval of time (see Theorem 5) during the execution of waypoint scan mode. Thus, $\mathcal{D}^j$ shall be fully covered given a sufficient number of activations of waypoint scan mode. Global coverage mode shall continue to select new $\mathcal{D}^j$, $\forall j \in \{1, ..., N_c\}$ and hand over coverage to waypoint scan mode until the entire domain has been sufficiently covered. Thus, the coverage error shall converge to zero at a final time t_f. This concludes the proof. □

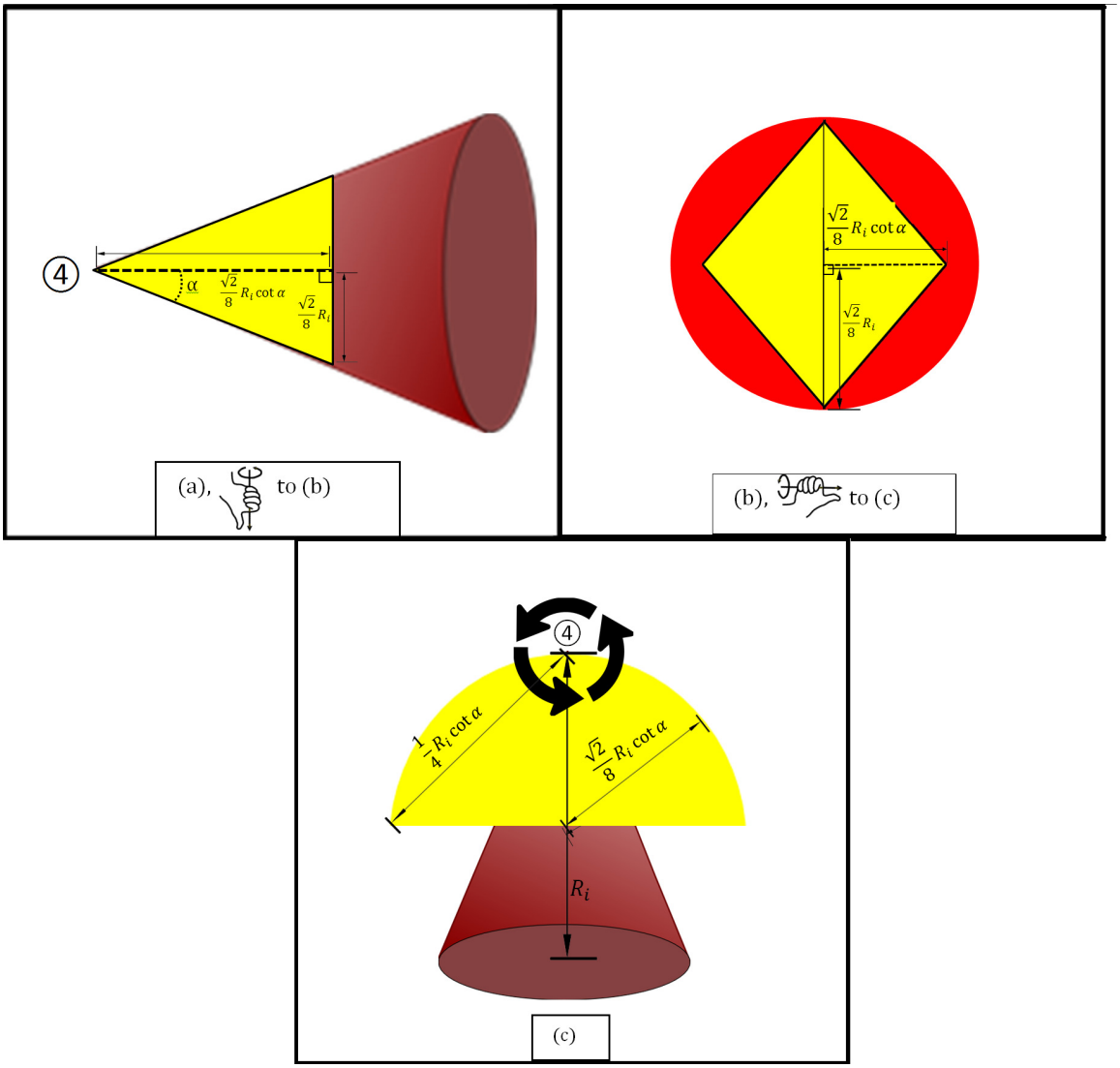


Figure 2.8: The coverage gap resulting from sweeps at sub-waypoints 2 and 3 is indicated in yellow and projected into 3 orthogonal planes in (a), (b), and (c). View (a) is the same as Fig. 2.7 and subsequent views are indicated by right hand rotations on the figure.

2.2.7 Experimental and Simulation-based Verification

2.2.7.1 Experimental Procedure

The coverage strategy was validated experimentally in an indoor test environment with the agents dispatched to explore a 2.55 m x 2.55 m x 2.3 m box. The agent team was composed of three commercially available Hummingbird quadrotors manufactured by Ascending Technologies. These vehicles are each powered by a single 11.1 Volt lithium polymer battery nominally rated at 2100 mAh. Each quadrotor features two embedded processors running at 1 kHz, denoted as the high-level processor (HLP) and the low-level processor (LLP). Additionally, the ground station in this experiment is a Dell Inspiron 3847 Desktop running Ubuntu 14.04 LTS on an Intel Core i7-4790 CPU @ 3.80 GHz x 8 with 15.6 GB of RAM. Attitude and position measurements for each quadrotor are provided by a Vicon motion capture system. The system runs at 250 Hz and provides accurate position measurements to within 1 mm within the test section.

Robot Operating System (ROS) provides a framework for exchanging data between the ground computer station, Vicon system, and the quadrotors. The control algorithm, as presented, is implemented in MATLAB. The resultant velocity inputs are transmitted via XBee serial modules from the ground station to the HLP through ROS using the Robotics System Toolbox. The HLP then issues motor commands to the LLP for execution. The desired rotor rates are achieved through the embedded proprietary control software in the LLP.

The cutoff voltage, V^{co}, was chosen as 10 Volts as this provided approximately 30 seconds or so of flight time for the Hummingbirds to transition from a hovering state through a stable landing sequence. After selecting this value, discharge trends of six batteries were recorded for a single agent trial in-order to determine an appropriate value for the parameter $\bar{T}$. These trends are presented in Fig. 2.9. The discharge times, denoted earlier as T_i, for the six batteries were 743, 758, 759, 755, 786, and 563 seconds respectively. $\bar{T}$ was chosen to be 990 seconds as this provides an upper bound on the measured discharge times and provides a sufficient window to install a fresh battery and re-establish communication with the ground station.

Within the MATLAB environment, the parameter $Q(t, \tilde{p})$ is represented by a large 3D matrix. This matrix is called each time step and augmented with the parameters $M_{1i}(p_j, \tilde{p})$, $\forall i \in \{1, ..., N\}$, $M_{2i_{cr}}(t, \tilde{p})$, and $M_{3l}(p_j, \tilde{p})$, $\forall l \in \{1, ..., N-1\}$. Thus, it is required that specific regions of $Q(t, \tilde{p})$ be accessed by their index locations and set to $C^{\star}$ each time step. Although this work nominally describes the shapes of these augmentation functions as spherical or hemispherical, an infinity norm may be equivalently used in their definitions

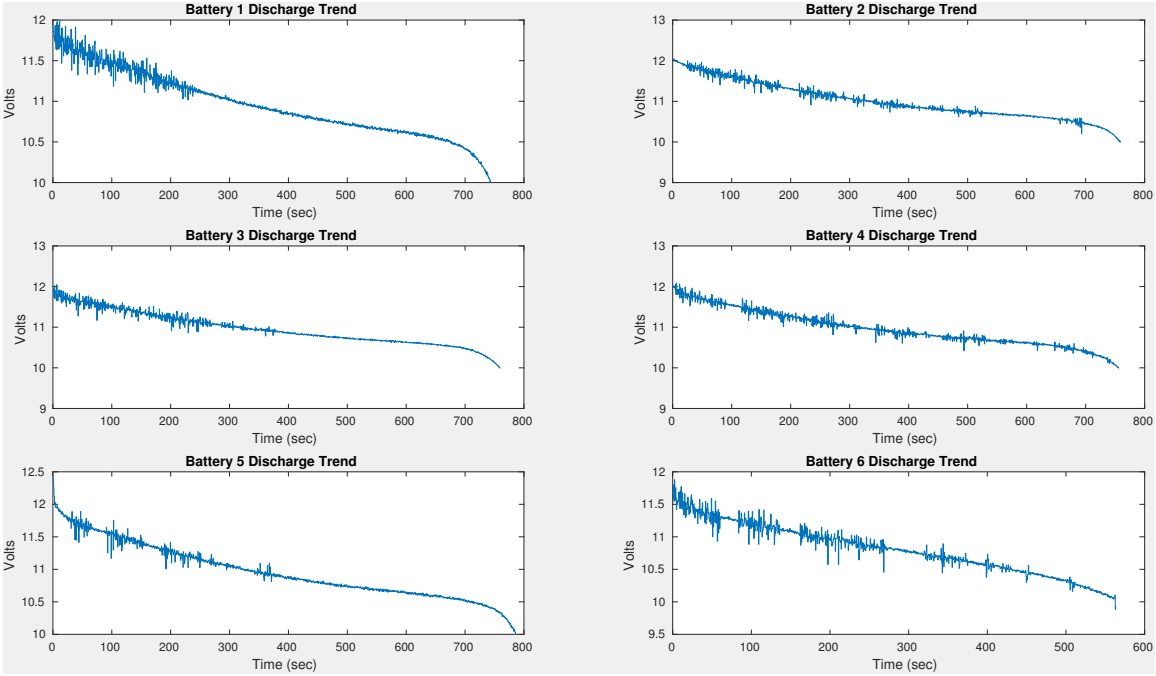


Figure 2.9: T_i values were experimentally determined for six batteries. These values are 743, 758, 759, 755, 786, and 563 seconds respectively.

rather than the implicit Euclidean norm. The infinity norm was chosen for this experimental implementation as it allowed for the augmentation functions to be defined in terms of a set of cuboid regions whose boundaries correspond in a direct manner to large index ranges. This allowed for vectorization based code which MATLAB is optimized to process. For this reason, this experimental domain is a cuboid rather than a hemisphere.

Two trials were conducted with identical parameter values. In each trial, one of the three agents was deployed from the origin every 5 minutes and 30 seconds nominally and tasked to explore the environment. The six batteries mentioned above were utilized in sequential order such that each agent should have two active exploration periods. As $V_i(t)$ of agent i approaches V^{co}, the quadrotor's position would converge upon the origin and hover a few inches above the ground. When the target position had been acquired and the battery voltage had dropped below V^{co}, an operator would cut off power to the motors thus dropping the quadrotor to the ground at it's current lateral position. The operator would then walk into the test section while avoiding the active quadrotors and replace the battery during the waiting period, T_i^w. At the next deployment point, t_i^d, the operator would return power to the agent and immediately hand control back to the ground station. With this protocol, the domain was persistently explored for upwards of 40 minutes with 2-3 quadrotors continuously in the air after the second deployment window. It should be noted

Table 2.1: The following parameter values were used in both trials

Parameter	Value	Parameter	Value
$C^\star$	10	$\bar{T}$	5 min 30 sec
k_u^{loc}	5×10^{-8}	k_v^{loc}	5×10^{-8}
k_w^{loc}	5×10^{-8}	k_s^{loc}	1×10^{-7}
ε_1	3.7 Hz	ε_2	0.2 m
ε_3	0.3 rad	U_{max}	0.1 m/s
$\kappa_{i_{cr}}$	0.13 m/V	γ_{ik}, $\forall k \in \{1, ..., 4\}$	7.8×10^{-3}
Λ_{i0}	40	Λ_{i1}	2.7×10^5
R_i	0.64 m	α	60°

here that in both trials 80% coverage was achieved within the first 10 minutes, the trials were permitted to run for 40 minutes in order to illustrate the long term efficacy of the deployment window spacing. The parameter values for both trials are presented in Table 2.1.

2.2.7.2 Experimental Results and Discussion

The coverage error evolution and inter-agent distances are presented in Figs. 2.10 and 2.11 respectively. It should be noted that the plotted value for $\mathscr{r}_i + \mathscr{r}_j$ is a physical measurement of 0.51 meters. The true value implemented in the MATLAB controller was set to 0.64 m in order to allow for some margin of stopping space on the part of the Ascending Technologies low level controller. The use of second-order modeling necessitates this margin as the vehicles cannot instantaneously change direction when new kinematic commands are uploaded.

For trial one, 50%, 80%, and 90% coverage were achieved after 236, 463, and 1241 seconds respectively. The protocol terminated with 95.6% coverage at 2350 seconds. Two anomalies occurred during this run. The first anomaly occurred at 1669 seconds. Fig. 2.10 illustrates that the proximity of agents 2 to 3 comes within a few centimeters less than $\mathscr{r}_i + \mathscr{r}_j$ at this time and yet the agents do not collide. At this moment, agent 2 was flying directly above agent 3 thus placing agent 3 into a downdraft generating additional control action and ultimately overcompensation on the part of the low level controller. This caused no issues as the vertical dimensions of the quadrotors are substantially less than the lateral dimensions. In future trials, the effect could be compensated for by increasing the margin for $\mathscr{r}_i$. Furthermore, this effect can be eliminated in future work by implementing quadrotor dynamics-based coverage control strategies rather than relying upon proprietary low level controllers such as those developed by Ascending Technologies. The second anomaly oc-

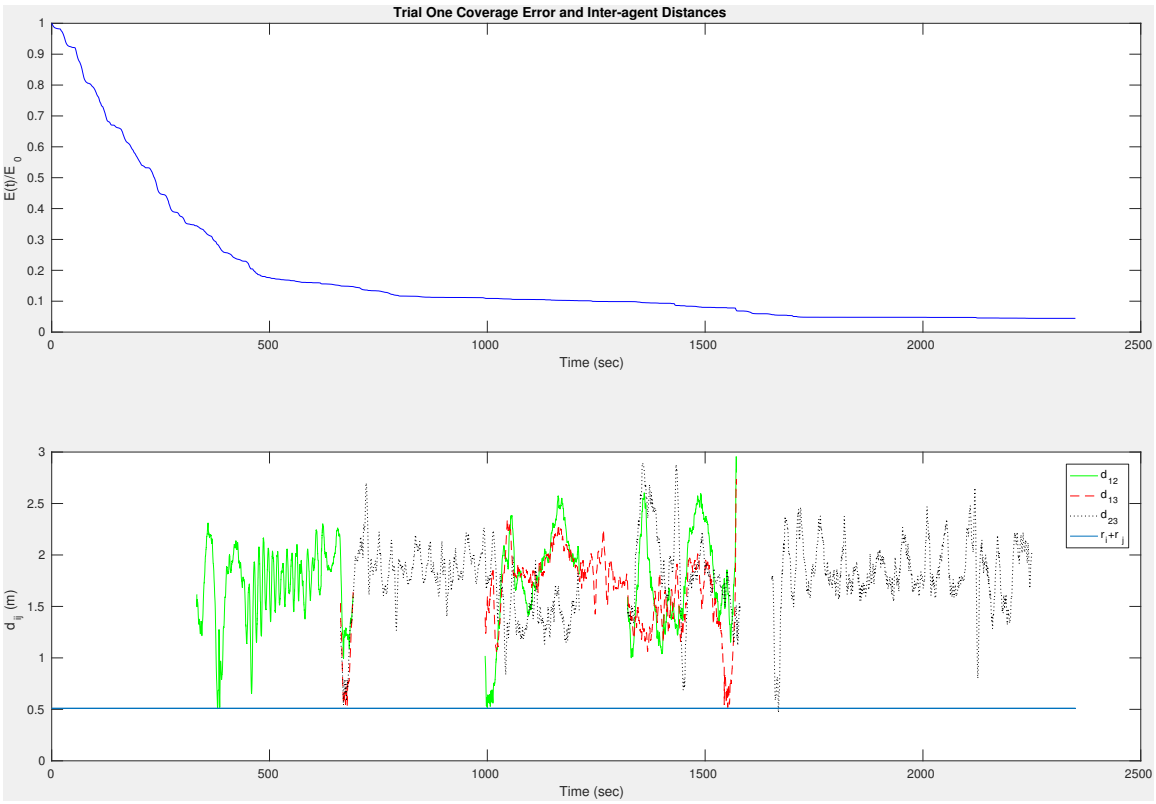


Figure 2.10: For Trial One 50%, 80%, and 90% coverage were achieved after 236, 463, and 1241 seconds respectively. The protocol terminated with 95.6% coverage at 2350 seconds. The agents successfully avoided collision although a vertical anomaly occurred at 1669 seconds between agents 2 and 3. d_{ij} is the distance of agent i from agent j, and the MATLAB typeset $r_i + r_j$ is intended as the minimum safe distance ($\mathit{r}_i + \mathit{r}_j$).

curred at 1572 seconds. After agent 1's redeployment at 995 seconds, it proceeded to operate correctly until a communication failure in the Xbee links occurred at 1572 seconds resulting in the agent flying out of control into the safety net thus ceasing coverage earlier than intended. The two remaining agents continued covering until the expiration of their batteries. As Table 2.2 illustrates, all six deployment windows occurred within 5 seconds of their nominal times and aside from the communication failure crash, all other exploration periods ended with a successfully executed landing at the origin. A video of trial one is available at https://www.youtube.com/watch?v=UjDi9PYa0zY&t=107s.

For trial two, 50%, 80%, and 90% coverage were achieved after 211, 511, and 856 seconds respectively. The protocol terminated with 97.4% coverage at 1859 seconds. Anomalies not associated with the algorithm itself occurred during this run. Specifically, another Xbee communication failure occurred at 1818 seconds leading to the crash of Agent 2 during it's second exploration protocol. Agent 3 was then landed manually 41 seconds later

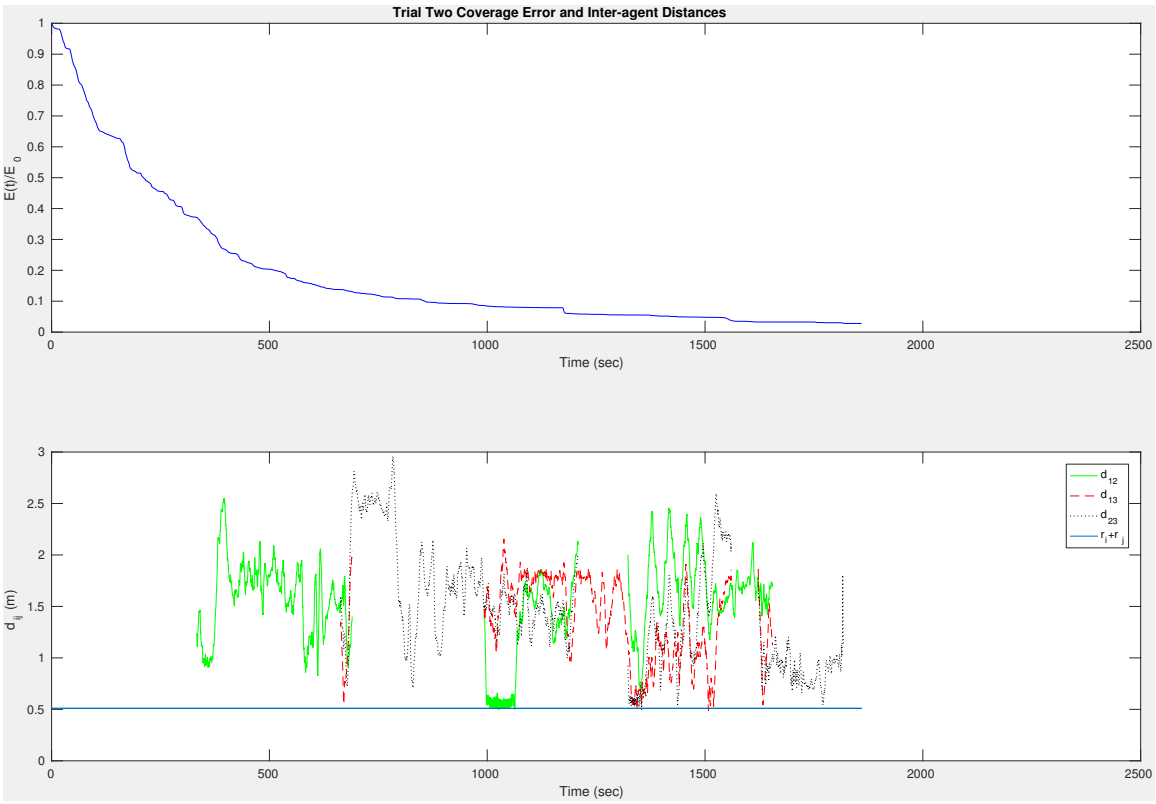


Figure 2.11: For Trial Two 50%, 80%, and 90% coverage were achieved after 211, 511, and 856 seconds respectively. The protocol terminated with 97.4% coverage at 1859 seconds The agents successfully avoided collision. d_{ij} is the distance of agent i from agent j, and the MATLAB typeset $r_i + r_j$ is intended as the minimum safe distance ($r_i + r_j$).

in order to assess damage to the crashed quadrotor. As Table 2.3 illustrates, five of the deployment windows occurred within seconds of the nominal times; however, agent 3 was relaunched 30 seconds early due to human error.

The battery voltage trends and operating modes of the hybrid automaton are presented for each trial in Figs. 2.12 and 2.13 respectively. The initial deployments for agent 1 in both trials are reflective of the fact that battery 1 has the shortest lifespan in the network. It was originally measured at 743 seconds in the days preceding the experiments and has further deteriorated through practice and tuning trials. However, these first deployment trends in Figs. 2.12 and 2.13 reflect that the 30 seconds of buffer time built into the choice of V_{co} still allow for the agent to converge upon $\mathcal{D}_{i_{cr}}$ which has now retracted to a point at the origin. This is essentially a compensation for the uncertainty in T_i as agents are actually able to maintain altitude with batteries readings as low as 9 Volts. Note in both of these figures that the tails of voltage trends always correspond with motion in subdomain transfer mode. The trajectories of the agents over the course of the trials are presented in Figs. 2.14 and

Table 2.2: For trial one, the nominal and true launch times are presented as well as the true landing times. Time is presented in seconds

Agent	Nominal Launch	True Launch	Landing
1	0	2	692
2	330	333	1211
3	660	663	1581
1	990	995	1572
2	1320	1322	2250
3	1650	1653	2360

Table 2.3: For trial two, the nominal and true launch times are presented as well as the true landing times. Time is presented in seconds

Agent	Nominal Launch	True Launch	Landing
1	0	2	690
2	330	333	1210
3	660	663	1560
1	990	993	1655
2	1320	1323	1818
3	1650	1622	1852

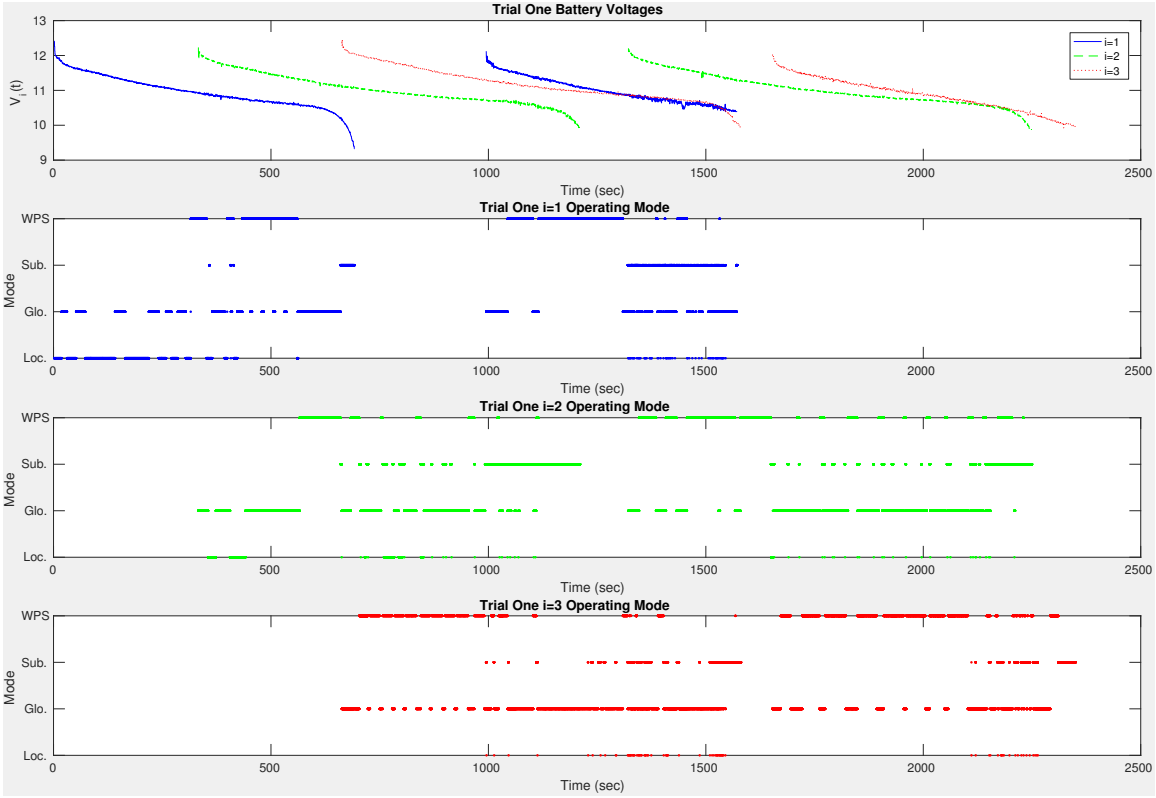


Figure 2.12: The six battery trends over the course of trial two are presented as well as the automaton modes of each of the three agents indexed $i = 1$, $i = 2$ and $i = 3$ respectively.

2.15 for reference.

Most of the anomalies reported are unrelated to the performance of the algorithm itself and result from the inevitable hardware, software and human errors that accumulate over the course of a complex 40 minute experiment. These experiments demonstrate that appropriately spaced periodic deployments of quadrotors from a single charging station can result in a long-duration surveillance operation. Additionally, the energy-aware domain partitioning technique resulted in trajectories that were globally attractive to the charging station in the limits that batteries expired. Aside from the anomalies discussed, each agent's flight path was directed to the origin as the battery expired allowing for a human to swap out the battery before the next redeployment window arrived.

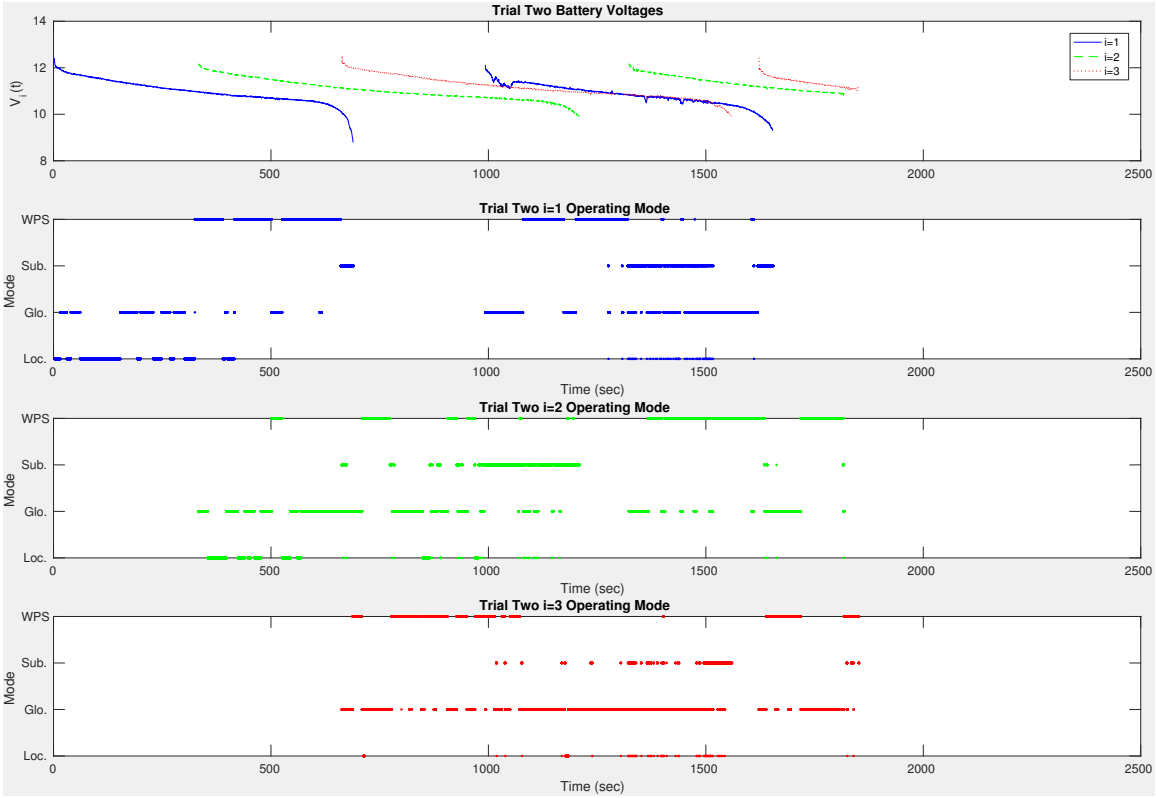


Figure 2.13: The six battery trends over the course of trial two are presented as well as the automaton modes of each of the three agents indexed $i = 1$, $i = 2$ and $i = 3$ respectively.

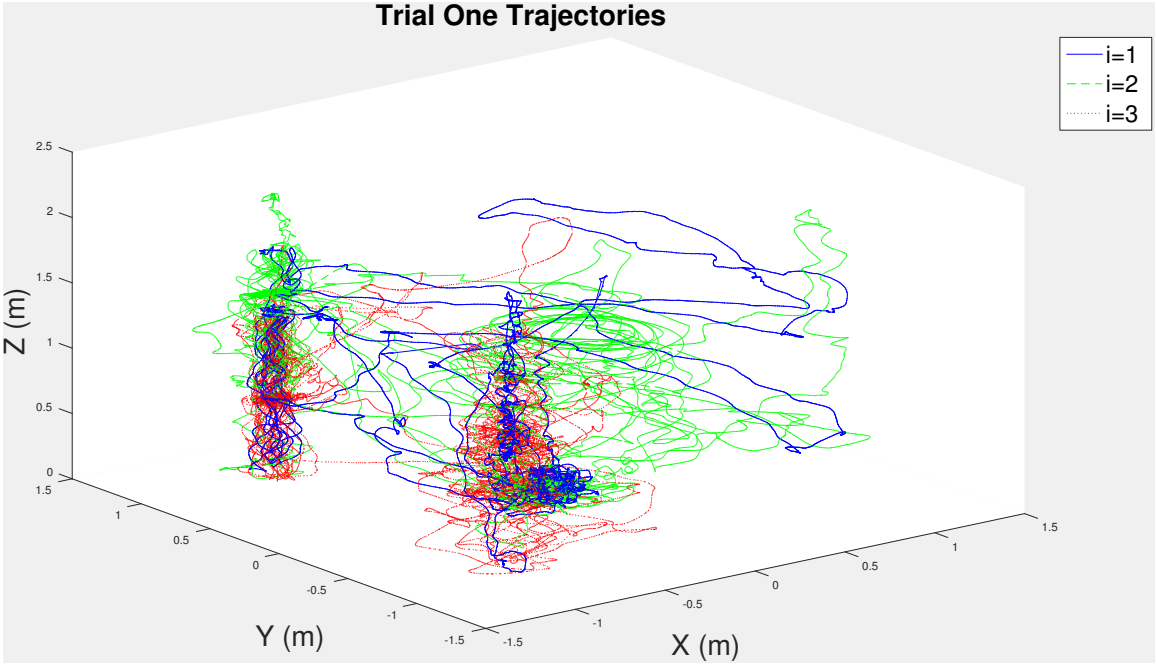


Figure 2.14: The trajectories of the three agents, indexed $i = 1$, $i = 2$ and $i = 3$ respectively, are presented over the course of trial one.

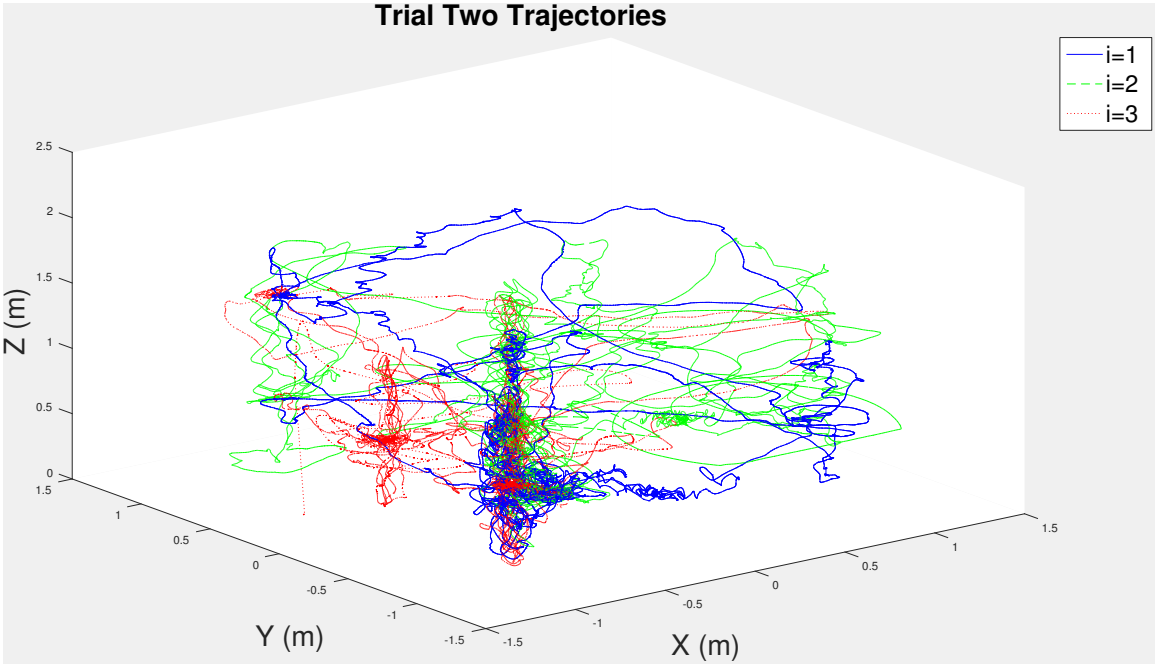


Figure 2.15: The trajectories of the three agents, indexed $i = 1$, $i = 2$ and $i = 3$ respectively, are presented over the course of trial two.

2.2.7.3 Simulation

A simulation was performed with $N = 10$ agents to illustrate the scalability of the algorithm. The following parameters differ from the experimental trials: $r_i = 0.5$, $R_i = 10$, $k_u^{loc} = k_v^{loc} = k_w^{loc} = 0.07$, $k_s^{loc} = 0.015$, and $\bar{T} = T_i = 238$ seconds for all agents. 90% and 95% coverage of the domain were achieved in 4567 and 8438 seconds respectively.

For the interest of data visualization clarity, only the first 300 or 3000 seconds of coverage time are presented in simulation figures. Agents deploy every 23.8 seconds from the charging station. The coverage error and inter-agent distances are presented in Fig. 2.16. Only distances with respect to $i = 1$ are presented to reduce clutter; however, all agents successfully avoid collision.

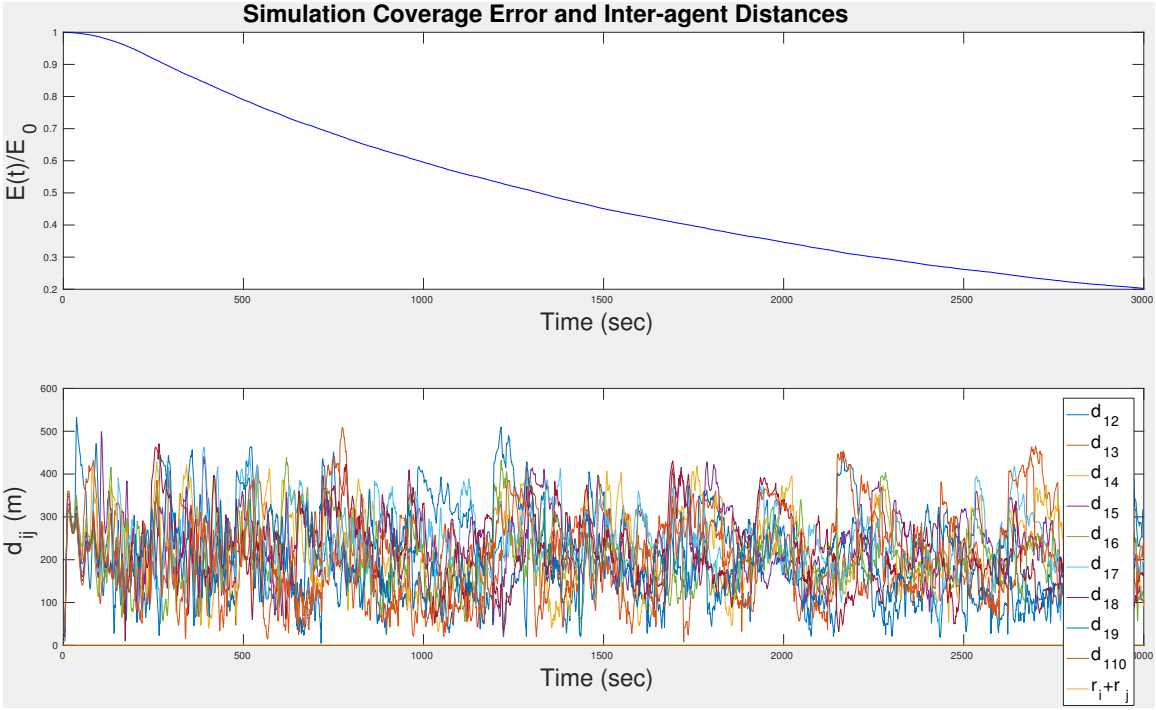


Figure 2.16: All ten agents avoided collision throughout the simulation. Only $i = 1$'s proximity from each agent is presented to reduce clutter. The MATLAB typeset $r_i + r_j$ is intended as the minimum safe distance ($r_i + r_j$).

Battery voltage trends for the ten agents are presented in Fig. 2.17. All agents use the same voltage trend: a recording of one of the DASC lab's quadrotors hovering for 5 minutes. The automaton modes for the first four agents are presented in Fig. 2.18 which provides the most striking contrast to that of the experimental results. Due to the small size of the lab space, it was difficult for agents to spend much time in local coverage mode as their sensing footprints were rather large compared to the size of the space. The increased

exploration volume in simulation allowed for agents to spend more time in local coverage mode.

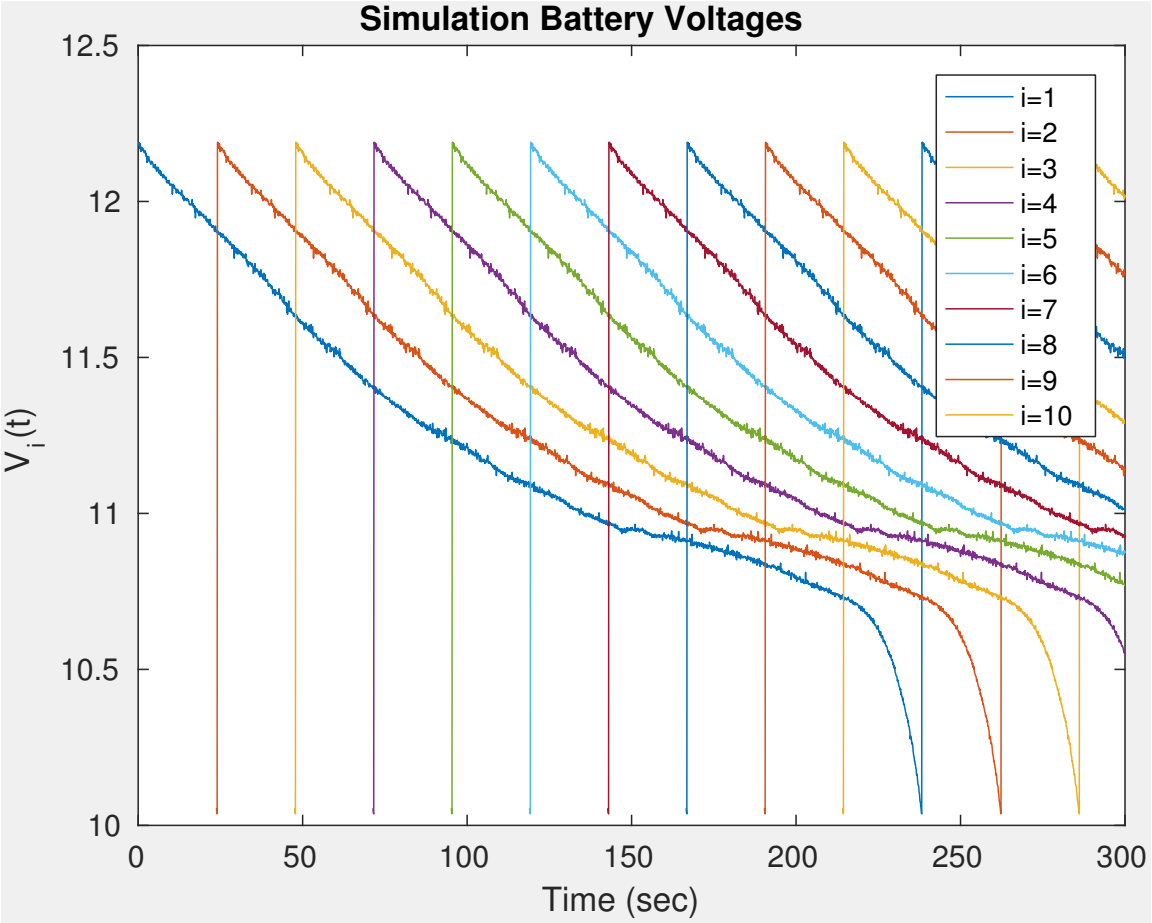


Figure 2.17: Voltage for each of the ten agents is displayed during the first 300 seconds of simulation time. Agents deploy every 23.8 seconds.

The trajectory of agent 1 for the first 300 seconds is presented in Fig. 2.19. The agent deploys from the origin and then transfers to the outer shell. The exploratory trajectory gradually converges upon the origin at 238 seconds. The agent is then redeployed. Fig. 2.20 displays $S_i(\tilde{q}_i, \tilde{p})$ for $i \in \{1, ..., 10\}$ over various shots in time to further illustrate the deployment strategy. The anisotropic quality of $S_i(\tilde{q}_i, \tilde{p})$ is visualized in Fig. 2.20. The coverage level over time of a vertical cross section of the domain is presented in Fig. 2.21.

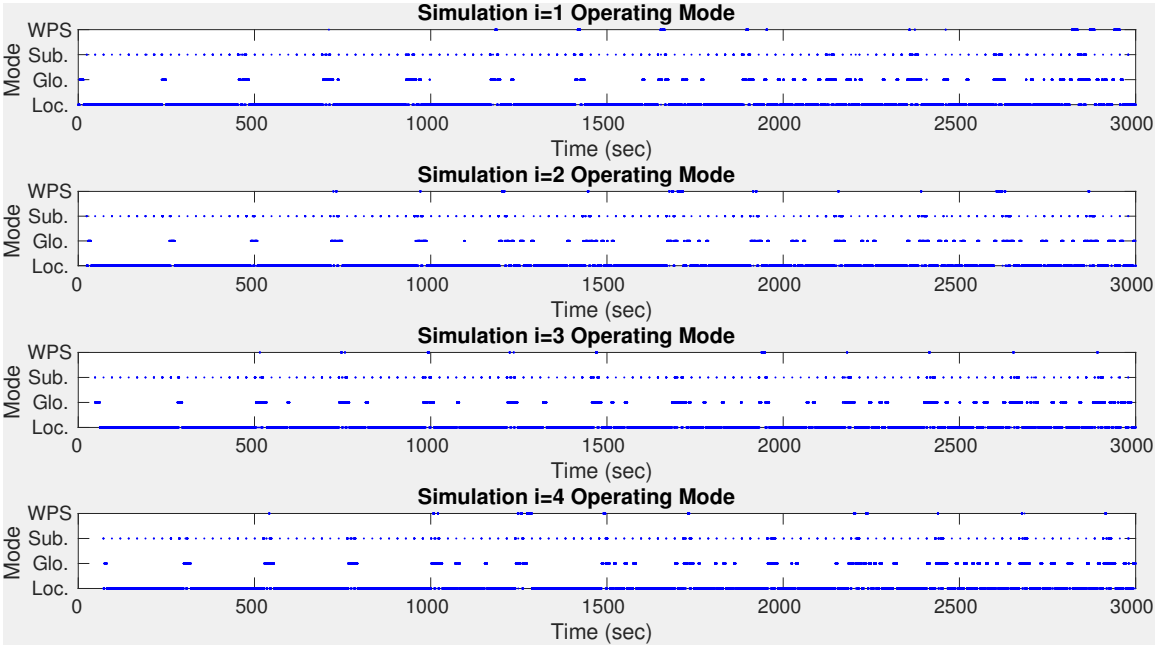


Figure 2.18: Automaton modes for the first four agents are presented. Agents deploy from the charging station and transfer to their designated shells in global coverage mode. Local coverage mode guides active exploration, while short periods of subdomain transfer mode appear every 23.8 seconds to transfer agents to inner shells.

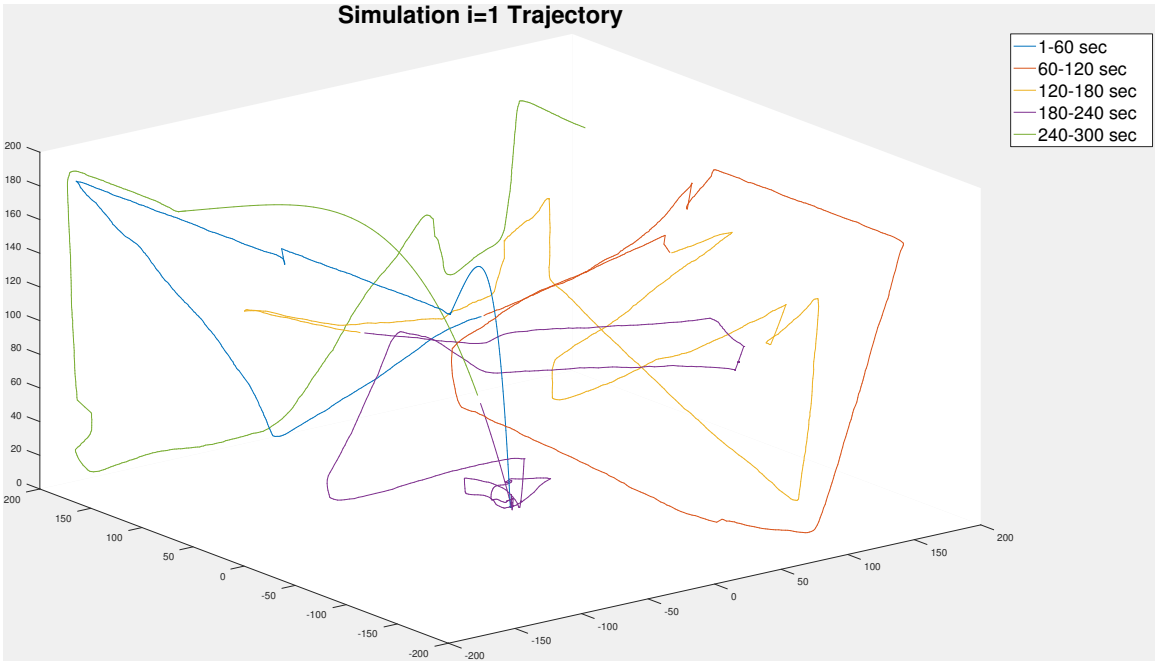


Figure 2.19: The trajectory of $i = 1$ is presented for the first 300 seconds. Color is used to indicate time spans along the trajectory to aid the reader in tracing.

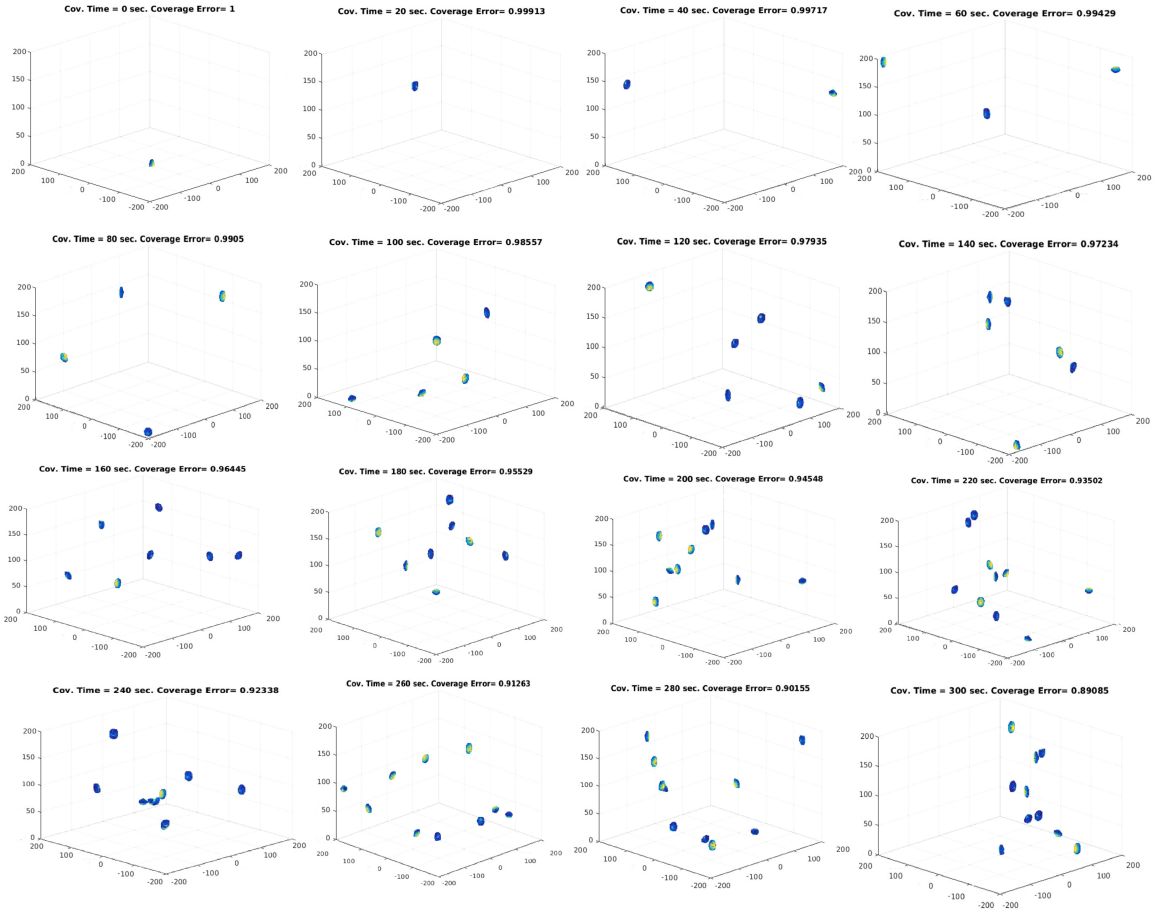


Figure 2.20: The first 300 seconds of simulation time is presented with color plots of $S_i(\tilde{q}_i, \tilde{p})$ for $i \in \{1, ..., 10\}$.

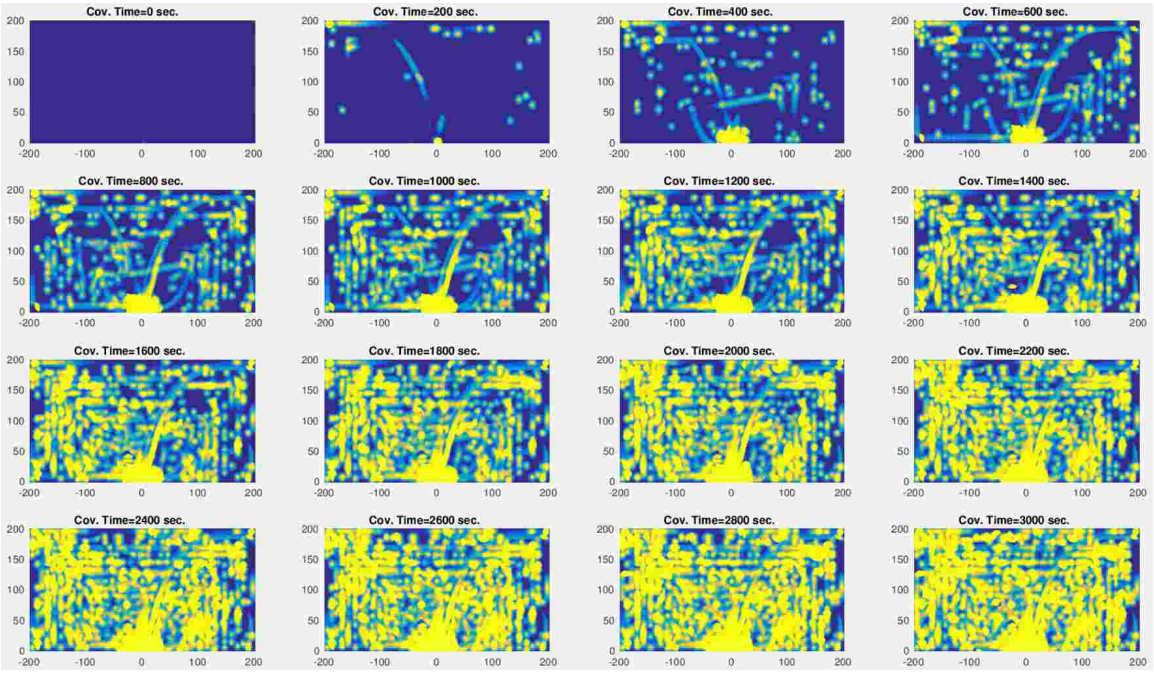


Figure 2.21: The coverage level of a vertical cross section of the domain is presented for the first 3000 seconds using the standard MATLAB color scale with dark blue indicating zero coverage and bright yellow indicating a coverage level of $C^\star$.

2.2.8 Conclusion

This section has presented an energy-aware complete 3D dynamic coverage control algorithm. The use of dynamic domain partitioning guarantees the generation of globally attractive flight trajectories in the limit of battery expiration. This control strategy is applicable to active 3D search, patrol, and environmental monitoring protocols. Previous work on this topic has been restricted quasi-3D coverage domains [50]. The efficacy of the algorithm has been demonstrated through experiments and simulations which highlight the effectiveness of the coverage protocol while maintaining the guaranteed safety provisions.

2.3 Persistent Coverage of 2D Manifolds in Stochastic Environments

2.3.1 Problem Formulation

2.3.1.1 Agent Modeling

Consider a network of spherical autonomous agents indexed $i \in \{1, ..., N\}$, of radius r_i, whose motion is subject to the same 3-D rigid body kinematics defined in (2.1)—(2.2).

Once again, $p_i = [x_i \;\; y_i \;\; z_i]^T$ is the position vector and $\Omega_i = [\Phi_i \;\; \Theta_i \;\; \Psi_i]^T$ is the vector of 3-2-1 Euler angles taken with respect to a global Cartesian coordinate frame $\mathcal{G}$ with origin $\mathcal{O}$. The linear velocities $[u_i \;\; v_i \;\; w_i]^T$ and angular velocities $[q_i \;\; r_i \;\; s_i]^T$ are both presented in the body fixed frame $\mathcal{B}_i$ with origin p_i. The state vector of agent i is once again $\tilde{q}_i = [p_i^T \;\; \Omega_i^T]^T$. The agents travel within a stationary domain, $\mathcal{D} \subset \mathbb{R}^3$. Their task is to survey a two-dimensional manifold, $\mathcal{C} \subset \mathcal{D}$, known as the surface of interest. For the purpose of this work, assume that the surface is an ellipsoid of revolution; however, it should be noted that the coverage laws, as well as the collision avoidance strategy, can be easily adapted for any convex surface. The ellipsoid has semi-major axis $x_{\mathcal{C},r}$ and semi-minor axis $z_{\mathcal{C},r}$ aligned with the global coordinate axes $\hat{x}_{\mathcal{G}}$ and $\hat{z}_{\mathcal{G}}$ respectively with center at $\mathcal{O}$. The circumflex (i.e., hat) symbols denote unit vectors.

Each agent, i, is once again equipped with the forward facing spherical sector sensor model $\mathcal{S}_i$ described in section 2.2. The only difference now is the introduction of two additional restrictions: all $\mathcal{S}_i$ have the same sensing range R, and $R > 2\,r_i + 2\,r_j$ where no other agents have radii larger than agent i or agent j. This is required in order to guarantee collision avoidance in the surface transfer mode described in the sequel. The model for agent i is depicted in Fig. 2.22.

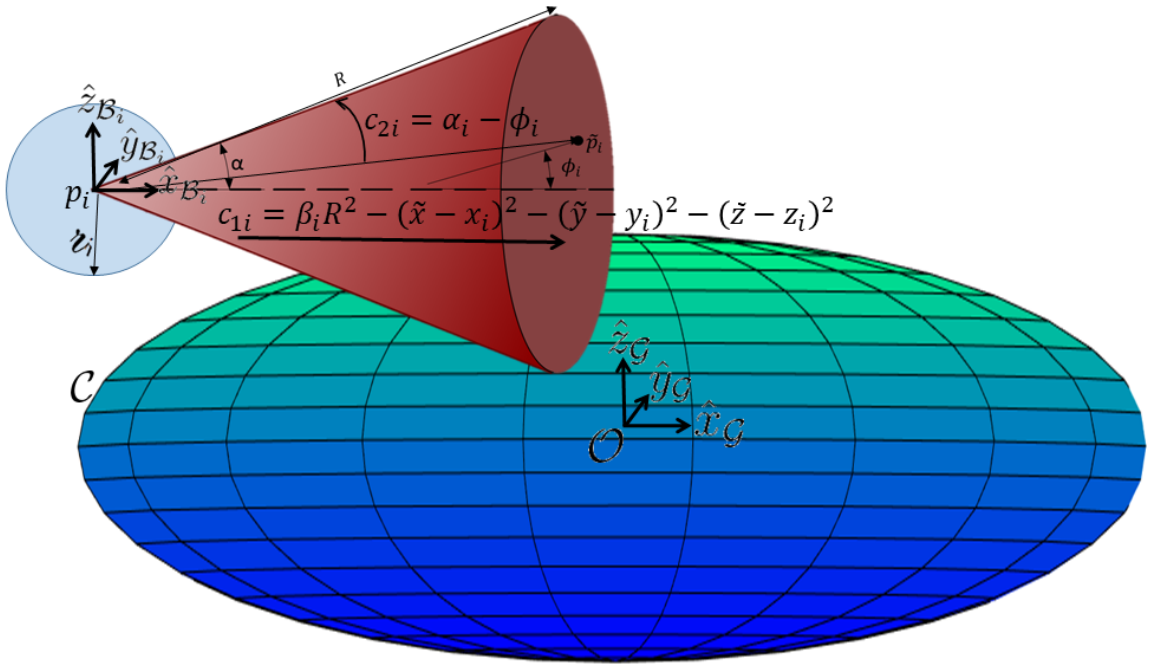


Figure 2.22: Agent i is modeled as a sphere of radius r_i and has a forward facing sensor footprint, $\mathcal{S}_i$. Sensing constraint functions c_{ki}, $\forall k \in \{1, 2\}$, encode a decay in sensing quality along the depth and towards the periphery of $\mathcal{S}_i$.

Define the coverage level provided by agent i at time t along $\mathcal{C}$ as:

$$Q_i^{\mathcal{C}}(t, \tilde{p}) = \int_0^t S_i\left(\tilde{q}_i(\tau), \tilde{p}\right) C\left(\tilde{p}\right) \, d\tau, \tag{2.59}$$

where C is defined as: $C\left(\tilde{p}\right) = \begin{cases} 1, & \forall \tilde{p} \in \mathcal{C}; \\ 0, & \forall \tilde{p} \notin \mathcal{C}, \end{cases}$ and encodes that the accumulation of sensing information only occurs along the surface of interest, $\mathcal{C}$.

Consider also a set of N_p high-speed particle intruders denoted $k \in \{1, ..., N_p\}$, each of which travels in an arbitrary direction at constant velocity, pass through the domain. The particles are assumed to be uncontrolled and cannot deviate from their initial trajectories. No assumptions are made with respect to the source of the particles or whether they are intelligently generated. Each particle contributes a decay term $\Lambda_k\left(\tau, \tilde{p}\right)$, which is defined later in section 2.3.1.3, to the global coverage level along $\mathcal{C}$:

$$Q^{\mathcal{C}}(t, \tilde{p}) = \sum_{i=1}^{N} Q_i^{\mathcal{C}}(t, \tilde{p}) - \sum_{k=1}^{N_p} \int_0^t \Lambda_k\left(\tau, \tilde{p}\right) C\left(\tilde{p}\right) \, d\tau. \tag{2.60}$$

The goal of this section is to derive a hybrid control strategy that persistently sweeps $\mathcal{S}_i$ across $\mathcal{C}$ while emphasizing surveillance within some bound of the predicted impact points of particles $k \in \{1, ..., N_p\}$ on $\mathcal{C}$. More specifically, this section establishes theoretical guarantees on the worst case path length from any agent to any arbitrary impact point, thus guaranteeing interception for prescribed bounds on intruder speed, detection range, and agent velocity while avoiding collisions. Let us define collision and interception.

Definition 2. Agent i is said to have intercepted particle k if i is within a ε_4 bound of the estimated impact point of k for a finite interval of time leading up to the impact.

During the interval of time leading up to the impact, agent i shall sweep the area in local coverage thus gathering information. Note that intruders are unaffected by agents and shall always impact the surface and then disappear. This does not damage the agents, which are free to resume other tasks upon conclusion of interception.

Definition 3. Agent i avoids collision so long as $\|p_i(t) - p_j(t)\| > r_i + r_j,\ \forall t \geq 0,\ \forall j \neq i \in \{1, ..., N\}$ and $\|n_i\| > r_i$, where the vector n_i has direction normal to $\mathcal{C}$ and length equal to the Euclidean distance of its intersection point on $\mathcal{C}$ to p_i.

Agents operate with finite power resources and are required to return every $T^\star$ time units to a fueling station denoted $\mathcal{F}$. Thus, a scheduling protocol is derived whereby agents periodically deploy from $\mathcal{F}$ to cover within assigned partitions of $\mathcal{C}$. These partitions are bounded by latitude lines and are sorted by geodesic distance from $\mathcal{F}$. Agents deploy to the partition that is furthest from $\mathcal{F}$ and then transfer between adjacent partitions every $\frac{T^\star}{N}$ time units as their power resources dwindle. Agents must return to $\mathcal{F}$ within $T^\star$ time units after deployment. This partitioning scheme has the benefit of ensuring that the network of agents is well distributed across $\mathcal{C}$. Agents are nominally assigned to intercept intruders whose predicted impact points lie within their own partition. This high level system geometry is illustrated in Fig. 2.23

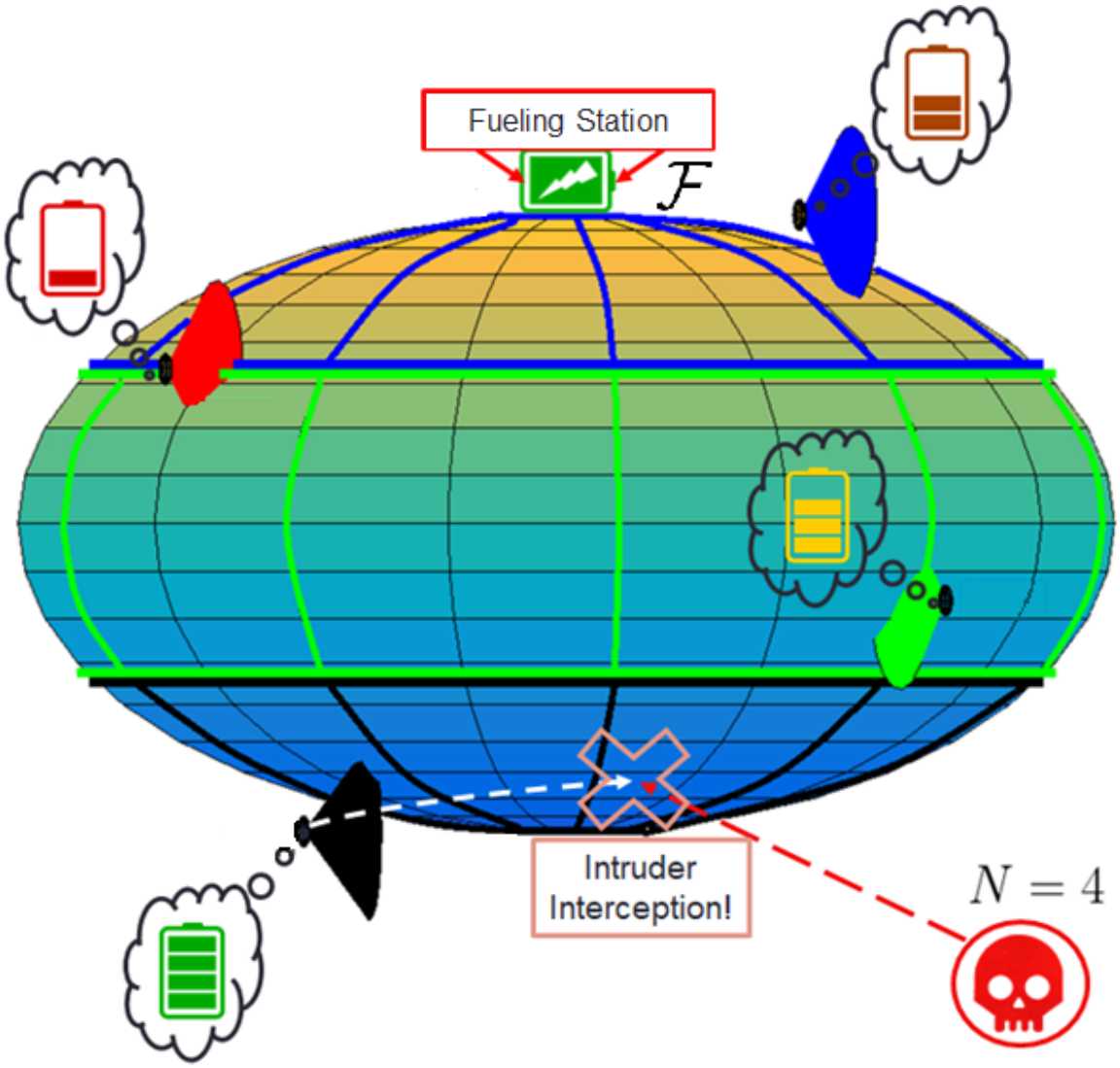


Figure 2.23: Agents have finite power resources and must return to $\mathcal{F}$ periodically for refueling. Upon deployment, agents are assigned to the latitude partition furthest from $\mathcal{F}$. The agents are reassigned to progressively higher partitions every $\frac{T^\star}{N}$ time units as their power resources dwindle.

Agent i is capable of localizing itself in $\mathcal{G}$ and detecting whether there exists j such that $\|p_i(t) - p_j(t)\| \leq R$[1]. Furthermore, agents i and j can communicate their deployment times to one another. A centralized network is required to publish the current coverage level

[1]In a real world implementation, the agent can localize its position using trilateration assuming that an array of radio beacons are present on $\mathcal{C}$. The orientation fix can be captured using star trackers in a space application. Inter-agent detection can be accomplished via radar.

$Q^{\mathcal{C}}(t, \tilde{p})$ to all agents[2] and to estimate the trajectories of intruders using an omnidirectional range sensor whose measurements are fed through an extended Kalman filter. Computation of $Q^{\mathcal{C}}(t, \tilde{p})$ is contingent upon continuous transmission of agent state $\tilde{q}_i$ to the centralized network. The centralized network assigns each intruder to an unassigned agent at closest latitude to the predicted impact point. It also transmits detection time as well as estimated location and time of impact to the agent.

2.3.1.2 Intruder Modeling

Assume that the omnidirectional range sensor (e.g., LiDAR) is co-located with $\mathcal{O}$ and provides measurements of each particle's position in spherical coordinates. Let us also assume that particle detection and state estimate initialization occur while the distance of the particle from $\mathcal{O}$ is greater than or equal to $R_{det} + x_{\mathcal{C},r}$ where R_{det} is a lower bound on distance from detection to impact. Define the model for the motion of particle k:

$$\dot{\tilde{q}}_k(t) = \begin{bmatrix} 0_{3\times3} & \mathbb{I}_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} \tilde{q}_k(t), \tag{2.61}$$

$$\tilde{z}_k(t) = \begin{bmatrix} \sqrt{x_k^2 + y_k^2 + z_k^2} \\ \operatorname{atan2}(y_k,\, x_k) \\ \arccos\left(\frac{z_k}{\sqrt{x_k^2+y_k^2+z_k^2}}\right) \end{bmatrix} + \epsilon, \tag{2.62}$$

where $\tilde{q}_k = [x_k,\, y_k,\, z_k,\, \dot{x}_k,\, \dot{y}_k,\, \dot{z}_k]^T$ and $\tilde{z}_k = [\rho_k,\, \theta_k,\, \psi_k]^T$ are the Cartesian state and spherical coordinate measurement vectors of particle k resolved in $\mathcal{G}$. Let us assume that particle speed is upper bounded such that $\sqrt{\dot{x}_k^2 + \dot{y}_k^2 + \dot{z}_k^2} \leq U_{max}^{int}$. ρ_k, θ_k, and ψ_k are the range, azimuthal angle, and polar angle of k respectively. In the sequel, the matrix in (2.62) shall be denoted $\tilde{h}(x_k,\, y_k,\, z_k)$. Assume that the measurement noise, ϵ, is zero-mean Gaussian and has covariance $\mathbf{R} = diag\left(\sigma_\rho^2, \sigma_\theta^2, \sigma_\psi^2\right)$. This system models high-speed particles incident upon a surface with negligible drag (e.g., micrometeoroids impacting a spacecraft hull); thus, it is reasonable to omit the process noise. The state and covariance estimates, $\hat{\tilde{q}}_k$ and $\mathbf{P}_k$, are computed with a continuous-time extended Kalman filter which is initialized upon particle k's detection at time t_{dk}.

[2]In practice, it is not necessary to publish $Q^{\mathcal{C}}(t, \tilde{p}), \forall \tilde{p} \in \mathcal{D}$ to every agent. For agent i to compute its local coverage control signal, it need only values for $Q^{\mathcal{C}}(t, \tilde{p})$ within a closed ball of radius R due to the fact that $S_i(\tilde{q}_i, \tilde{p}) = 0, \forall \tilde{p} \notin \bar{B}_R p_i(t)$. This substantially reduces the communication overhead.

2.3.1.3 Information Decay

At any time t, consider the decay rate map $\Lambda_k(t, \tilde{p})$ for particle k in terms of its predicted position and covariance evolution over a horizon $T_{H,k}(t)$. As the particles are assumed to travel at fixed velocities[3], the predicted values for Cartesian position $\tilde{p}'_k(t+\tau)$ and associated covariance $\mathbf{P}_k(t+\tau)$ are $\tilde{p}'_k(t+\tau) = G(\tau)\hat{\tilde{q}}_k(t)$, and $\mathbf{P'}_k(t+\tau) = G(\tau)\mathbf{P}_k(t)G(\tau)^T$ respectively, where $G(\tau) = [\mathbb{I}_{3\times3} \ \ \tau\mathbb{I}_{3\times3}]$ and $\hat{\tilde{q}}_k(t)$ is the current estimate for $\tilde{q}_k(t)$. Define the decay rate map associated with particle k as the integral of the predicted normal distribution $\mathcal{N}(\tilde{p}'_k(t+\tau), \mathbf{P'}_k(t+\tau))$ through horizon $T_{H,k}$:

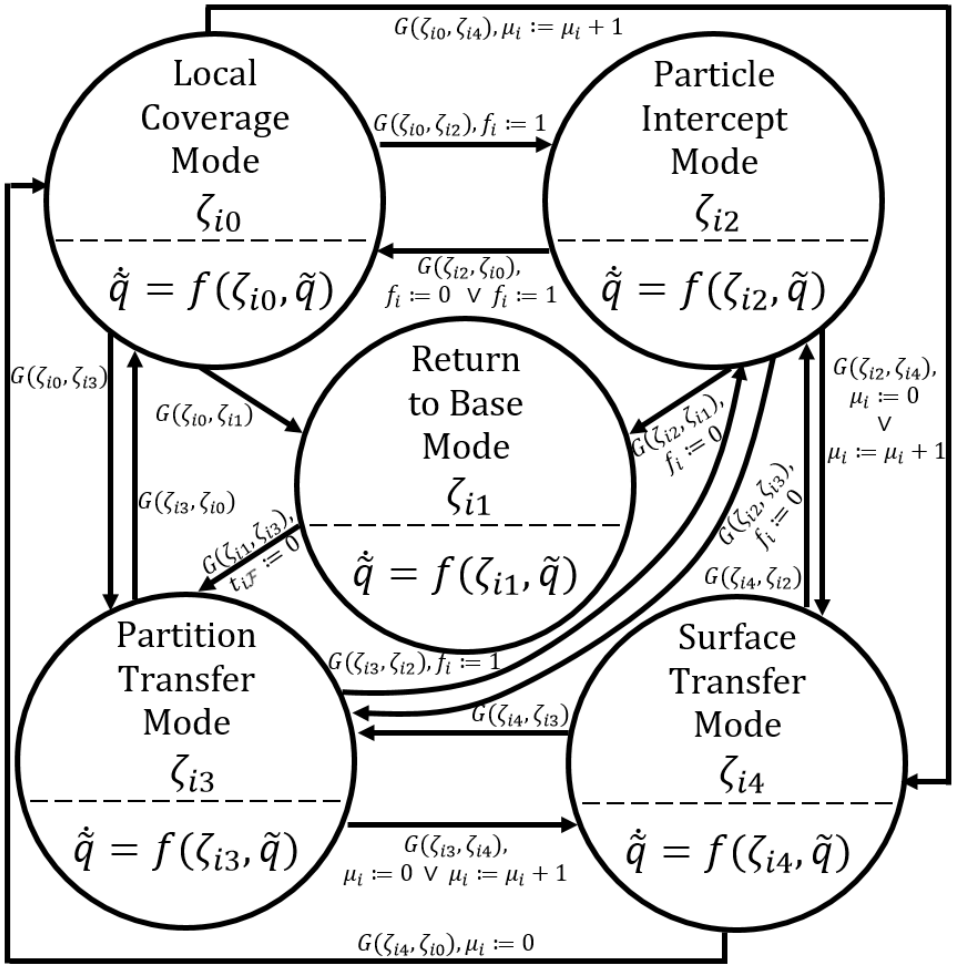
$$\Lambda_k(t, \tilde{p}) = \int_0^{T_{H,k}(t)} \lambda_k \mathcal{N}\big(\tilde{p}'_k(t+\tau), \mathbf{P'}_k(t+\tau)\big)\, d\tau, \tag{2.63}$$

where $\lambda_k > 0$ is a tuning parameter for the decay rate. It is recommended that one choose λ_k sufficiently less than $\max_{\tilde{p}\in\mathcal{S}_i} S_i(\tilde{q}_i(t), \tilde{p})$ such that the rate of coverage shall exceed the decay rate over points intersecting $\mathcal{S}_i$. For $t < t_{dk}$, define $\Lambda_k(t, \tilde{p}) = 0, \forall \tilde{p} \in \mathcal{D}$. The formulation for (2.63) essentially takes a normal distribution for the position of particle k at time t and cumulatively propagates it forward in time up to the horizon $T_{H,k}(t)$. The horizon is lower-bounded by an estimate of the remaining time until impact of particle k on $\mathcal{C}$. This may be computed using $\tilde{q}_k(t)$ along with the surface geometry. With this design, $Q^{\mathcal{C}}(t, \tilde{p})$ decays along the predicted trajectory of k with tapering omnidirectional decay rates spreading out from the predicted path. This design lends itself naturally to the local coverage formulation, which is gradient following in nature, in that the agents may follow these tapered decay paths towards the predicted impact points on the surface of interest.

2.3.1.4 Algorithmic Overview

The coverage strategy for agent i is represented by the hybrid automaton in Fig. 2.24. Definitions of all entities of the automaton, including the guard conditions and reset maps, are included in Appendix B.1. Note that each agent operates in accordance with its own automaton and thus an arbitrary number of agents may be in any operating mode at any given time. Before proceeding, let us provide a brief overview of each mode.

[3]The guarantee of intruder interception presented in this work can be extended to intruders with time-varying velocities that are bounded by U^{int}_{max}. However, it is still required that intruders follow straight line trajectories such that the network may estimate fixed impact points.



$G(\zeta_{i0}, \zeta_{i1})$: Power-critical agent returns to refueling station.
$G(\zeta_{i0}, \zeta_{i2})$: Agent assigned to intercept an intruder and particle assignment flag set.
$G(\zeta_{i0}, \zeta_{i3})$: Agent transfers to partition assigned by time since deployment.
$G(\zeta_{i0}, \zeta_{i4})$: Transfers to higher surface to avoid collision and augments surface assignment index.
$G(\zeta_{i1}, \zeta_{i3})$: Refueled agent transfers to furthest partition and resets time since deployment.
$G(\zeta_{i2}, \zeta_{i0})$: Covers around predicted or past intruder impact point. Particle assignment flag set to zero in the latter case.
$G(\zeta_{i2}, \zeta_{i1})$: Power-critical agent returns to refueling station. Particle assignment flag set to zero.
$G(\zeta_{i2}, \zeta_{i3})$: Transfers to assigned partition after intruder impact. Particle assignment flag set to zero.
$G(\zeta_{i2}, \zeta_{i4})$: Transfers to higher surface to avoid collision or returns to lowest surface upon arrival at destination projection. Surface assignment index reset as appropriate.
$G(\zeta_{i3}, \zeta_{i0})$: Covers in assigned partition.
$G(\zeta_{i3}, \zeta_{i2})$: Agent assigned to intercept an intruder and particle assignment flag set.
$G(\zeta_{i3}, \zeta_{i4})$: Transfers to higher surface to avoid collision or returns to lowest surface upon arrival at destination projection. Surface assignment index reset as appropriate.
$G(\zeta_{i4}, \zeta_{i0})$: Agent in assigned partition with no assigned intruder covers on the lowest surface with surface assignment index set to zero.
$G(\zeta_{i4}, \zeta_{i2})$: Continues along assigned surface towards intruder interception.
$G(\zeta_{i4}, \zeta_{i3})$: Continues along assigned surface towards assigned partition.

Figure 2.24: Agent i operates in accordance with this automaton. For clarity, elements of the reset map and brief descriptions of transitions are included.

- Local Coverage: This mode is similar in principle to that of its counterpart in section 2.2. In this iteration, it governs the active exploration of the surface of interest $\mathcal{C}$. An agent currently assigned to an intruder may operate in Local Coverage while within an ε_4 bound of the the intruder's predicted impact point. Any agent not currently assigned to an intruder shall operate in Local Coverage assuming that it is within its assigned latitude partition. Operation in Local Coverage is always concurrent with agent assignment to the lowest concentric surface (see Surface Transfer below) and transition to Local Coverage can occur from any mode aside from Return to Base Mode.

- Particle Intercept: In this mode, an intruder is assigned to agent i, which in turn is guided along its assigned surface to the predicted impact point of the intruder. After intruder assignment occurs, the agent i will nominally remain in Particle Intercept Mode until after the intruder impacts $\mathcal{C}$; however, the agent may temporarily leave the mode before impact to avoid collision through Surface Transfer Mode or to explore in Local Coverage within a ε_4 bound of the the intruder's predicted impact point.

- Partition Transfer: This mode is defined for agents that are not currently assigned to an intruder, and its purpose is to ensure that the agents are spatially distributed across the entire surface of $\mathcal{C}$. To this end, agents are assigned to separate latitude partitions with all partitions having equal surface area. Activation of this mode will guide agent i along a longitudinally-oriented geodesic trajectory until its position satisfies the partition's associated latitude constraints. Agents travel to the southernmost latitude partition upon deployment and transition through progressively northern partitions as their fuel is depleted. Transition to this mode can occur from any other mode. The partitioning scheme is shown in Fig. 2.28.

- Surface Transfer: This mode's primary purpose is to ensure that agents avoid collision with two distinct cases resulting in its activation. In the first case, two or more agents have violated a safe-proximity condition. The mode removes select agents from the deadlock by guiding them along vectors normal to $\mathcal{C}$ to a higher-altitude ellipsoidal surface concentric with $\mathcal{C}$. An agent trajectory is then temporarily confined to this newly assigned surface until it reaches the surface projection of its destination. The second case occurs under the condition that the agent has arrived at the projection of its destination on a higher-altitude surface. The mode is activated to return the agent to the innermost surface. Transition to this mode can occur from any other mode aside from Return to Base as agents in the latter mode always take priority in a deadlock. The surface transfer geometry is illustrated in Fig. 2.26.

- Return to Base: The final mode is activated when the time since an agent's deployment has surpassed some threshold. It guides the power-critical agent along the optimal trajectory to the refueling station. After charging, the agent is redeployed. Agent deployments occur one at a time with a fixed period. A power critical agent in Particle Intercept Mode or Surface Transfer Mode shall first complete its task of intercepting the assigned intruder or transferring surfaces before transitioning to Return to Base Mode. If an agent is designated as power-critical while in Partition Transfer Mode it shall immediately abandon its task and transition to Local Coverage which shall result in instantaneous transition to Return to Base Mode. Theoretical guarantees on successful return to base with respect to agent power lifespan in accordance with the automaton is presented in Theorem 8 of section 2.3.4.

2.3.2 Local Coverage Mode

Local coverage constitutes one of five hybrid modes in the persistent coverage automaton. As in section 2.2, this mode is gradient-following and commands agent i to orient and translate $\mathcal{S}_i$ such that the volume of uncovered space intersecting $\mathcal{S}_i$ is increased. In this way, it emphasizes active exploration of the domain by agents that are not currently assigned to either monitor intruders or relocate within the domain. In this section, the persistent coverage control laws are derived such that agent motion in local coverage tends to reduce the rate of growth of the global coverage error. Define the global coverage error along $\mathcal{C}$ with respect to $C^\star$ as:

$$E^{\mathcal{C}}(t) = \int_{\mathcal{D}} h\left(C^\star C\left(\tilde{p}\right) - Q^{\mathcal{C}}(t,\tilde{p})\right)\, d\tilde{p}. \tag{2.64}$$

In a similar manner to that in section 2.2, the local coverage control laws for persistent coverage will be derived via differentiation of (2.64). Invoking (2.9) allows for differentiation of (2.64) with respect to time: $\dot{E}^{\mathcal{C}}(t) = \int_{\mathcal{D}} h'(C^\star C\left(\tilde{p}\right) - Q^{\mathcal{C}}(t,\tilde{p}))\left(\frac{-\partial Q^{\mathcal{C}}(t,\tilde{p})}{\partial t}\right) d\tilde{p} + \int_{\partial\mathcal{D}} \left(h(C^\star C\left(\tilde{p}\right) - Q^{\mathcal{C}}(t,\tilde{p}))\right)\mathbf{v}_{(\mathbf{s})} \cdot \mathbf{n}\, dA$, which reduces to: $\dot{E}^{\mathcal{C}}(t) = \int_{D} h'(C^\star C\left(\tilde{p}\right) - Q^{\mathcal{C}}(t,\tilde{p}))\left(\frac{-\partial Q^{\mathcal{C}}(t,\tilde{p})}{\partial t}\right) d\tilde{p}$, and expands to:

$$
\begin{aligned}
\dot{E}^{\mathcal{C}}(t) &= -\int_{D} h'(C^{\star}C\left(\tilde{p}\right) - Q^{\mathcal{C}}(t,\tilde{p}))\Big(\sum_{i=1}^{N} S_i(q_i(t),\tilde{p})C\left(\tilde{p}\right) - \sum_{k=1}^{N_p} \Lambda_k\left(t,\tilde{p}\right)C\left(\tilde{p}\right)\Big)d\tilde{p} \\
&= \sum_{i=1}^{N} \underbrace{\int_{D} -h'(C^{\star}C\left(\tilde{p}\right) - Q^{\mathcal{C}}(t,\tilde{p}))S_i(q_i(t),\tilde{p})C\left(\tilde{p}\right)d\tilde{p}}_{=\hat{e}_i^{\mathcal{C}}(t)} \\
&\quad - \sum_{k=1}^{N_p} \underbrace{\int_{D} -h'(C^{\star}C\left(\tilde{p}\right) - Q^{\mathcal{C}}(t,\tilde{p}))\Lambda_k\left(t,\tilde{p}\right)C\left(\tilde{p}\right)d\tilde{p}}_{=\tilde{e}_k^{\mathcal{C}}(t)} \\
&= \sum_{i=1}^{N} \hat{e}_i^{\mathcal{C}}(t) - \sum_{k=1}^{N_p} \tilde{e}_k^{\mathcal{C}}(t).
\end{aligned}
\tag{2.65}
$$

$\hat{e}_i^{\mathcal{C}}(t)$ is the rate of change of the coverage error due to the motion of the agents, while $\tilde{e}_k(t)$ is the rate of change of the coverage error due to a contrived information decay surrounding the predicted impact point of particle k on $\mathcal{C}$. The proposed strategy is to control the agents' kinematics, recovered in the derivative of $\hat{e}_i^{\mathcal{C}}(t)$, to decrease (2.65). It is not presumed that the local coverage strategy provides any additional bounds on (2.64)—nor are guarantees provided on the rate of growth of this contrived quantity. Curtailing the growth of the coverage error simply imparts the desired effect of active exploration in the vicinity of impact points into the system. Using this strategy, the agents actively seek to increase their rate of coverage by rotating and/or translating $\mathcal{S}_i$ to encompass the most uncovered space in the local vicinity.

Taking the derivative of $\hat{e}_i^{\mathcal{C}}(t)$ with respect to time yields:

$$
\dot{\hat{e}}_i^{\mathcal{C}}(t) = \int_{D_i} \Big(h''(C^{\star}C\left(\tilde{p}\right) - Q^{\mathcal{C}}(t,\tilde{p}))S_i(q_i(t),\tilde{p})C\left(\tilde{p}\right)\frac{\partial Q^{\mathcal{C}}(t,\tilde{p})}{\partial t} - h'(C^{\star}C\left(\tilde{p}\right) - Q^{\mathcal{C}}(t,\tilde{p}))\frac{d}{dt}(S_i(q_i(t),\tilde{p}))C\left(\tilde{p}\right)\Big)d\tilde{p}.
\tag{2.66}
$$

As in section 2.2, $\frac{d}{dt}(S_i(q_i(t),\tilde{p}))$ expands to take the form of (2.15). Now introduce the

following definitions:

$$a_{i0}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) = \int\limits_{D_i} h''(C^{\star}C\,(\tilde{p}) - Q^{\mathcal{C}}(t,\tilde{p}))S_i(q_i(t),\tilde{p})C\,(\tilde{p})\,\frac{\partial Q^{\mathcal{C}}(t,\tilde{p})}{\partial t}d\tilde{p}, \tag{2.67}$$

$$a_{i1}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) = \int\limits_{D_i} h'(C^{\star}C\,(\tilde{p}) - Q^{\mathcal{C}}(t,\tilde{p}))C\,(\tilde{p})\left(\frac{\partial S_i}{\partial x_i}\cos\Theta\cos\Psi\ + \frac{\partial S_i}{\partial y_i}\cos\Theta\sin\Psi - \frac{\partial S_i}{\partial z_i}\sin\Theta\right)d\tilde{p}, \tag{2.68}$$

$$\begin{aligned} a_{i2}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) &= \int\limits_{D_i} h'(C^{\star}C\,(\tilde{p}) - Q^{\mathcal{C}}(t,\tilde{p}))C\,(\tilde{p}) \\ &\left(\frac{\partial S_i}{\partial x_i}(\sin\Phi\sin\Theta\cos\Psi\ - \cos\Phi\sin\Psi) + \frac{\partial S_i}{\partial y_i}(\sin\Phi\sin\Theta\sin\Psi + \cos\Phi\cos\Psi)\ + \frac{\partial S_i}{\partial z_i}\sin\Phi\cos\Theta\right)d\tilde{p}, \end{aligned} \tag{2.69}$$

$$\begin{aligned} a_{i3}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) &= \int\limits_{D_i} h'(C^{\star}C\,(\tilde{p}) - Q^{\mathcal{C}}(t,\tilde{p}))C\,(\tilde{p}) \\ &\left(\frac{\partial S_i}{\partial x_i}(\cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi)\ + \frac{\partial S_i}{\partial y_i}(\cos\Phi\sin\Theta\sin\Psi - \sin\Phi\cos\Psi) + \frac{\partial S_i}{\partial z_i}\cos\Phi\cos\Theta\right)d\tilde{p}, \end{aligned} \tag{2.70}$$

$$a_{i4}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) = \int\limits_{D_i} h'(C^{\star}C\,(\tilde{p}) - Q^{\mathcal{C}}(t,\tilde{p}))C\,(\tilde{p})\left(\frac{\partial S_i}{\partial \Psi_i}\sin\Phi\sec\Theta + \frac{\partial S_i}{\partial \Theta_i}\cos\Phi\right)d\tilde{p}, \tag{2.71}$$

$$a_{i5}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) = \int\limits_{D_i} h'(C^{\star}C\,(\tilde{p}) - Q^{\mathcal{C}}(t,\tilde{p}))C\,(\tilde{p})\left(\frac{\partial S_i}{\partial \Psi_i}\cos\Phi\sec\Theta - \frac{\partial S_i}{\partial \Theta_i}\sin\Phi\right)d\tilde{p}. \tag{2.72}$$

Through the definitions in (2.67)—(2.72), one may restate (2.66) as:

$$\begin{aligned} \dot{\hat{e}}_i^{\mathcal{C}}(t) = a_{i0}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) - u_i(t)a_{i1}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) - v_i(t)a_{i2}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) \\ - w_i(t)a_{i3}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) - r_i(t)a_{i4}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})) - s_i(t)a_{i5}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p})). \end{aligned} \tag{2.73}$$

One can increase the coverage rate by mobilizing the sensor through exploitation of (2.68)—(2.72). One strategy is to define the following control law:

$$u_i^{\mathcal{C},loc} = k_u\frac{a_{i1}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p}))}{\sqrt{{a_{i1}^{\mathcal{C}}}^2 + {a_{i2}^{\mathcal{C}}}^2 + {a_{i3}^{\mathcal{C}}}^2}} + \hat{x}_{\mathcal{B}_i}\cdot\rho_{l,i}, \tag{2.74a}$$

$$v_i^{\mathcal{C},loc} = k_v\frac{a_{i2}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p}))}{\sqrt{{a_{i1}^{\mathcal{C}}}^2 + {a_{i2}^{\mathcal{C}}}^2 + {a_{i3}^{\mathcal{C}}}^2}} + \hat{y}_{\mathcal{B}_i}\cdot\rho_{l,i}, \tag{2.74b}$$

$$w_i^{\mathcal{C},loc} = k_w\frac{a_{i3}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p}))}{\sqrt{{a_{i1}^{\mathcal{C}}}^2 + {a_{i2}^{\mathcal{C}}}^2 + {a_{i3}^{\mathcal{C}}}^2}} + \hat{z}_{\mathcal{B}_i}\cdot\rho_{l,i}, \tag{2.74c}$$

$$r_i^{\mathcal{C},loc} = \bar{r}_i sat\big(\frac{k_r a_{i4}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p}))}{\bar{r}_i}\big) + \hat{y}_{\mathcal{B}_i}\cdot\rho_{a,i}, \tag{2.74d}$$

$$s_i^{\mathcal{C},loc} = \bar{s}_i sat\big(\frac{k_s a_{i5}^{\mathcal{C}}(t,Q^{\mathcal{C}}(t,\tilde{p}))}{\bar{s}_i}\big) + \hat{z}_{\mathcal{B}_i}\cdot\rho_{a,i}, \tag{2.74e}$$

where:

$$\rho_{l,i} = -\ln\left(\frac{1}{\gamma R - \not{z}_i}\left(\|n_i\| - \not{z}_i\right)\right)\mathcal{R}_1^{-1}\hat{n}_i, \tag{2.75}$$

$$\rho_{a,i} = \xi\mathcal{R}_2^{-1}\begin{bmatrix} 0 \\ \arcsin\left(\hat{n}_i \cdot \hat{z}_{\mathcal{G}}\right) - \Theta_i \\ \text{atan2}\left(-\hat{n}_i \cdot \hat{y}_{\mathcal{G}}, -\hat{n}_i \cdot \hat{x}_{\mathcal{G}}\right) - \Psi_i \end{bmatrix}. \tag{2.76}$$

Note that the functional dependencies of $a_{i1}^{\mathcal{C}}$, $a_{i2}^{\mathcal{C}}$ and $a_{i3}^{\mathcal{C}}$ are omitted from the denominators of the local coverage control laws in the interest of space. The term $\rho_{l,i}$ encodes collision avoidance with respect to the surface of interest. It takes a value of zero when agent i's normalized distance from $\mathcal{C}$ is γR for $\gamma \in (\frac{\not{z}_i}{R}, 1]$ and is logarithmically repulsive and attractive from the surface when the distance is decreased or increased respectively. Small values for γ tend to direct the agent to travel closer to the surface. This coincides with a smaller cross section of $\mathcal{S}_i$ intersecting the surface but is also typically associated with a higher quality of sensing. A larger choice for γ will direct the agent to fly at a higher altitude with respect to the surface and thus the area covered by $\mathcal{S}_i$ will tend to be broader with a decreased quality of sensing. The term $\rho_{a,i}$, for $\xi << 1$, encodes that the agents should tend to align $\hat{x}_{\mathcal{B}_i}$ with $-\hat{n}_i$ if the coverage terms associated with r_i and s_i have become sufficiently small. The physical meaning of $\rho_{a,i}$ is to direct $\mathcal{S}_i$ back onto $\mathcal{C}$ if it has reached a configuration in which it no longer intersects $\mathcal{C}$. See Fig. 2.25 for further illustration of the effects of $\rho_{l,i}$ and $\rho_{a,i}$.

$\bar{r}_i$ and $\bar{s}_i$ are saturation limits for the coverage angular velocity inputs to the system. k_u, k_v and k_w are tuning gains which are chosen to satisfy $\sqrt{k_u^2 + k_v^2 + k_w^2} \leq U_{max}^{agt}$. As $\rho_{\ell,i}$ is normal to the surface, it can be shown that U_{max}^{agt} is an upper bound to agent velocity tangential to $\mathcal{C}$.

2.3.3 Particle Intercept Mode

Assuming that particle k is embedded within the surface upon impact, its position trajectory shall intersect $\mathcal{C}$ at most one time. Define particle k's estimated impact time as:

$$t_{ck} = \min\left(t \in \mathbb{R}^+ \mid \frac{\left(\hat{x}_k + \dot{\hat{x}}_k t\right)^2}{x_{\mathcal{C},r}^2} + \frac{\left(\hat{y}_k + \dot{\hat{y}}_k t\right)^2}{x_{\mathcal{C},r}^2} + \frac{\left(\hat{z}_k + \dot{\hat{z}}_k t\right)^2}{z_{\mathcal{C},r}^2} = 1\right), \tag{2.77}$$

with estimated impact point $\tilde{p}_k'(t_{ck}) = G(t_{ck} - t)\,\hat{\tilde{q}}_k(t)$. Upon detection, particle k is assigned to a free agent i with the minimum distance from the estimated point of impact

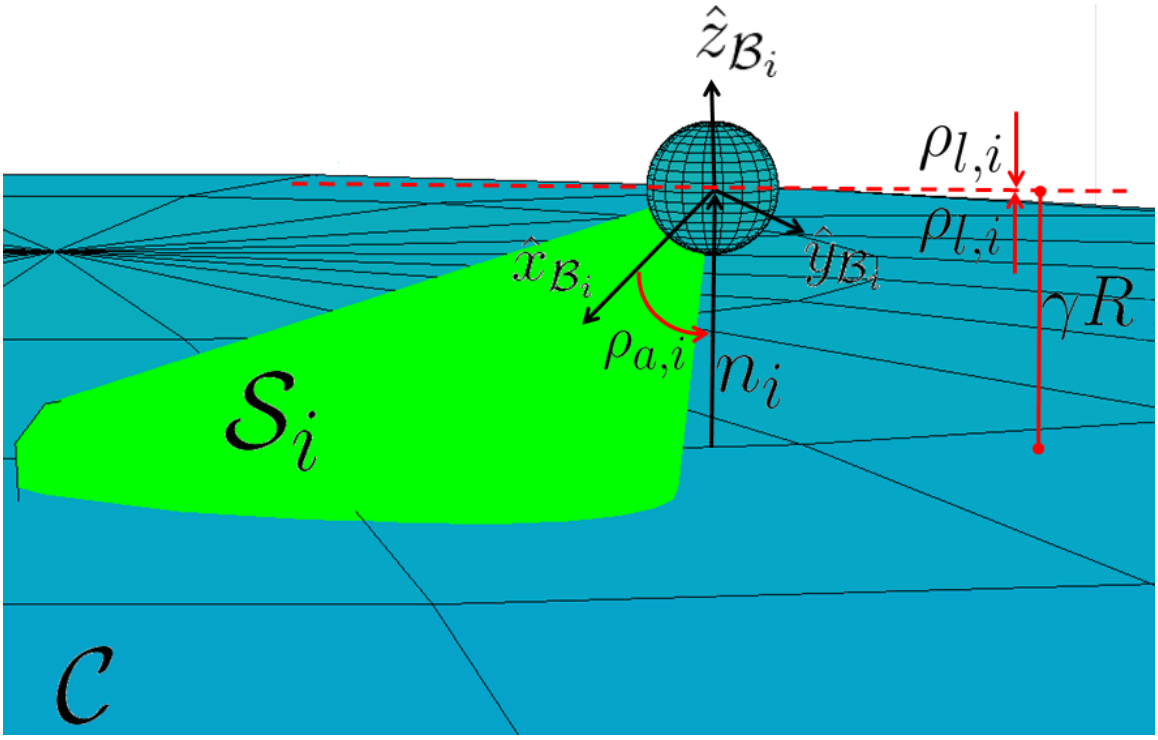


Figure 2.25: As agent i explores $\mathcal{C}$, $\rho_{l,i}$ is parallel to n_i for $\|n_i\| < \gamma R$, antiparallel to n_i for $\|n_i\| > \gamma R$, and the zero vector otherwise. This term prevents collision of i with $\mathcal{C}$ and prevents i from flying away from $\mathcal{C}$. $\rho_{a,i}$ tends to direct $\mathcal{S}_i$ onto $\mathcal{C}$.

along the $\hat{z}_{\mathcal{G}}$ direction. Define a new index, i_k, as the index of the agent assigned to intercept particle k at destination $p_{\star,i} = \tilde{p}'_k(t_{ck})$:

$$i_k = \underset{i \in \{1,\ldots,N\} | i_p \neq 1, f_i \neq 1}{\operatorname{argmin}} \|\tilde{z}'_k(t_{ck}) - z_i(t_{dk})\|. \tag{2.78}$$

Note that $\tilde{z}'_k(t_{ck})$ is the z component of $\tilde{p}'_k(t_{ck})$ and $f_i \in \{0, 1\}$ is a particle assignment flag for agent i, which is 0 when the agent is free (i.e., not currently assigned a particle). $i_p \in \{1, ..., N\}$, the power index of agent i, shall be fully described in section 2.3.4; however, it should be noted that the definition of i_k implies that there are at most $N-1$ agents available for particle interception at any given time.

It is assumed that agents shall maintain a distance γR normal to $\mathcal{C}$ in the nominal case that they are not maneuvering to avoid collision. Define an ellipsoid of revolution, $\mathcal{C}_0$, which is concentric with $\mathcal{C}$ and has the property that each semi-principal axis is γR longer than its associated counterpart in $\mathcal{C}$, i.e., $x_{\mathcal{C}_0,r} = x_{\mathcal{C},r} + \gamma R$, and $z_{\mathcal{C}_0,r} = z_{\mathcal{C},r} + \gamma R$. The nominal trajectories of i are attractive to $\mathcal{C}_0$. Agents maneuvering to avoid collision shall transfer to additional concentric ellipsoidal surfaces each separated by a distance of

R. These surfaces are denoted $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{N-1}$ with associated semi-principal axes $x_{\mathcal{C}_1,r} = x_{\mathcal{C},r} + (\gamma+1)\,R$ and $z_{\mathcal{C}_1,r} = z_{\mathcal{C},r} + (\gamma+1)\,R$, $x_{\mathcal{C}_2,r} = x_{\mathcal{C},r} + (\gamma+2)\,R$ and $z_{\mathcal{C}_2,r} = z_{\mathcal{C},r} + (\gamma+2)\,R$, etc. Surface assignment and transfer scheduling in collision avoidance mode is described in full detail in section 2.3.5 and the geometry is illustrated in Fig. 2.26.

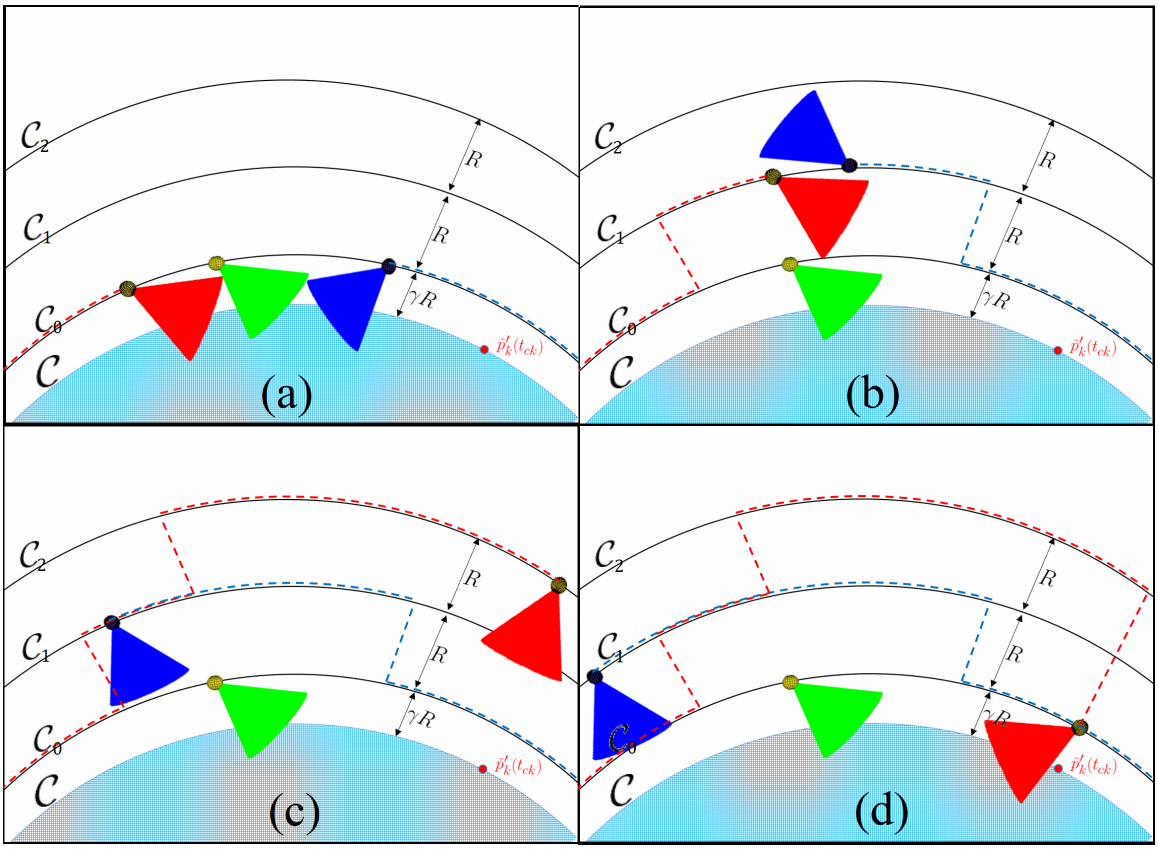


Figure 2.26: Three agents enter a deadlock in (a). The green agent, which has the greatest time since deployment, is prioritized to continue on $\mathcal{C}_0$ and the red and blue agents are each transferred to $\mathcal{C}_1$ before entering a second deadlock in (b). The blue agent, which has the second greatest time since deployment, is prioritized to continue on $\mathcal{C}_1$ and the red agent is transferred to $\mathcal{C}_2$ before continuing along geodesic to $proj_{\mathcal{C}_2}\tilde{p}'_k(t_{ck})$ in (c). Red agent transfers back to $\mathcal{C}_0$ directly above predicted impact point of particle k in (d). Note that surface transfer trajectories are always normal to $\mathcal{C}_{\mu_i}$, $\forall \mu_i \in \{0, ..., N-1\}$.

When agent i has been assigned to intercept particle k, f_i is set to 1 and it is said to have transitioned into particle intercept mode. In this mode, agent i shall nominally follow the distance-optimal trajectory along $\mathcal{C}_0$ to within a ε_4 bound of the projection of point $\tilde{p}'_k(t_{ck})$ onto $\mathcal{C}_0$ (denoted $proj_{\mathcal{C}_0}\tilde{p}'_k(t_{ck})$). The agent shall then transition to local coverage to actively explore within this ε_4 bound until $t > t_{ck}$ at which time f_i is set to 0. If local coverage guides the agent out of the ε_4 bound, particle intercept mode will again guide the agent back inside the bound. The distance-optimal trajectory is referred to as

a geodesic and its computation may be executed in an iterative manner. Specifically, the author uses Vincenty's formulae as presented in [93]. For cases involving nearly antipodal points in which the standard inverse method does not converge, the author uses Vincenty's supplemental algorithm presented in [94].

As an input, Vincenty's algorithm requires an ellipsoid of revolution along with two points, current and desired position, on that surface. The algorithm returns a heading angle measured clockwise from North. This heading angle shall be referred to as χ_i. Define the heading unit vector $\hat{\imath}_i$ which lies in a plane tangent to the surface at p_i. It may be computed by rotating the North-pointing vector at p_i clockwise by an angle of χ_i within the tangent plane. For this implementation of Vincenty's algorithm, the following inputs are required: $\mathcal{C}_{\mu_i}$ for surface assignment index $\mu_i \in \{0, ..., N-1\}$, p_i, and $proj_{\mathcal{C}_{\mu_i}} \tilde{p}'_k(t_{ck})$. The position controller used to guide agent i to $proj_{\mathcal{C}_{\mu_i}} \tilde{p}'_k(t_{ck})$ is composed of two terms: one which commands velocity tangential to $\mathcal{C}_{\mu_i}$ along $\hat{\imath}_i$ and one logarithmic term which commands velocity normal to $\mathcal{C}_{\mu_i}$ in order to constrain the geodesic trajectory of i to $\mathcal{C}_{\mu_i}$. The particle intercept mode position control law is:

$$\begin{bmatrix} u_i^{\mathcal{C},pim} \\ v_i^{\mathcal{C},pim} \\ w_i^{\mathcal{C},pim} \end{bmatrix} = U_{max}^{agt} \mathcal{R}_1^{-1} \frac{\hat{\imath}_i - \ln\left(\frac{1}{(\gamma+\mu_i)R - \mathscr{k}_i}\left(\|n_i\| - \mathscr{k}_i\right)\right)\hat{n}_i}{\|\hat{\imath}_i - \ln\left(\frac{1}{(\gamma+\mu_i)R - \mathscr{k}_i}\left(\|n_i\| - \mathscr{k}_i\right)\right)\hat{n}_i\|}. \tag{2.79}$$

As agent i travels along the geodesic, it is desirable to point $\mathcal{S}_i$ towards $\mathcal{C}$. Therefore, the orientation controller for particle intercept mode is similar to that of section 2.3.2:

$$\begin{bmatrix} q_i^{\mathcal{C},pim} \\ r_i^{\mathcal{C},pim} \\ s_i^{\mathcal{C},pim} \end{bmatrix} = \mathcal{R}_2^{-1} \begin{bmatrix} 0 \\ \arcsin\left(\hat{n}_i \cdot \hat{z}_{\mathcal{G}}\right) - \Theta_i \\ \text{atan2}\left(-\hat{n}_i \cdot \hat{y}_{\mathcal{G}}, -\hat{n}_i \cdot \hat{x}_{\mathcal{G}}\right) - \Psi_i \end{bmatrix}, \tag{2.80}$$

which is essentially a proportional controller that tends to align $\hat{x}_{\mathcal{B}_i}$ with $-\hat{n}_i$. As (2.79) commands the vehicle to follow the optimal length path along $\mathcal{C}_{\mu_i}$ to $proj_{\mathcal{C}_{\mu_i}} \tilde{p}'_k(t_{ck})$, one can establish a few guarantees on system performance. To simplify notation, define:

$$g_{\mathcal{C}_{N-1}} = \left[1 + \sum_{n=1}^{\infty} \left(\frac{(2n-1)!!}{2^n n!}\right)^2 \frac{\left(\frac{x_{\mathcal{C}_{N-1},r} - z_{\mathcal{C}_{N-1},r}}{x_{\mathcal{C}_{N-1},r} + z_{\mathcal{C}_{N-1},r}}\right)^{2n}}{(2n-1)^2}\right], \tag{2.81}$$

and $g_{\mathcal{C}_0}$ is defined similarly in terms of the semi-principal axes of $\mathcal{C}_0$.

Lemma 1. Let us assume that agent i has been assigned to particle k with $f_i := 1$. Given an arbitrary agent position $p_i(t_{dk})$ and an arbitrary predicted impact point for the intruder

$\tilde{p}'_k(t_{ck})$, there exists an upper bound to the maximum path length until interception:

$$\mathcal{P}_{max} \leq \pi x_{\mathcal{C}_{N-1},r} + \frac{\pi}{2}\left(x_{\mathcal{C}_{N-1},r} + z_{\mathcal{C}_{N-1},r}\right) g_{\mathcal{C}_{N-1}} + 2\left(N-1\right) R. \tag{2.82}$$

Proof. At the moment that $f_i := 1$, agent i transitions to Particle Intercept Mode. Under the condition that the agent has not yet come within proximity of the predicted impact point, i.e., $\|p_i - \mathit{proj}_{\mathcal{C}_{\mu_i}} \tilde{p}'_k(t_{ck})\| > \varepsilon_4$, only $G\left(\zeta_{i2}, \zeta_{i4}\right)$ and $G\left(\zeta_{i4}, \zeta_{i2}\right)$ are permissible states (see Appendix). These two transitions occur sequentially for each deadlock event that agent i encounters as it approaches $\mathit{proj}_{\mathcal{C}_{\mu_i}} \tilde{p}'_k(t_{ck})$.

In any given deadlock arrangement, one agent remains on its current surface without ascending to a higher one. This implies that $\mu_i = 1$ for at most $N-1$ agents as the remaining agent would be unable to encounter a deadlock on $\mathcal{C}_0$. Furthermore, this implies that $\mu_i = 2$ for at most $N-2$ agents et cetera. It follows that $\mu_i = N$ for at most zero agents. The worst cast surface assignment that can be incurred during sequential cycles of $\left(\left(\zeta_{i2}, \zeta_{i4}\right), \left(\zeta_{i4}, i2\right)\right)$ would be $\mu_i = N-1$.

As the geodesic path length between any two points projected onto surface $\mathcal{C}_{\mu_i}$ is less than the geodesic path length between the same two points projected onto surface $\mathcal{C}_{\mu_i+1}$, one may bound the geodesic portion of the trajectory by a trajectory that is constrained entirely to $\mathcal{C}_{N-1}$. Denote this term $\mathcal{P}_{geo}$. As any two points on $\mathcal{C}_{\mu_i}$ can be connected by a path of constant latitude $\mathcal{P}_{lat}$ followed by a path of constant longitude $\mathcal{P}_{long}$, it follows that:

$$\mathcal{P}_{geo} \leq \mathcal{P}_{lat} + \mathcal{P}_{long}. \tag{2.83}$$

For two generic points on $\mathcal{C}_{N-1}$, it is true that:

$$\mathcal{P}_{lat} \leq \pi x_{\mathcal{C}_{N-1},r}, \tag{2.84}$$

$$\mathcal{P}_{long} \leq \frac{\pi}{2}\left(x_{\mathcal{C}_{N-1},r} + z_{\mathcal{C}_{N-1},r}\right) g_{\mathcal{C}_{N-1}}. \tag{2.85}$$

where the bound on $\mathcal{P}_{lat}$ is half of the circumference of the ellipsoid of revolution $\mathcal{C}_{N-1}$ about its equator and the bound on $\mathcal{P}_{long}$ is half of the perimeter of the revolved ellipse. The infinite series expression term, denoted $g_{\mathcal{C}_{N-1}}$ in (2.85), is first presented in [95]. The remaining portion of the path length is simply the straight line segments connecting $\mathcal{C}_0$ to $\mathcal{C}_{N-1}$ and back again. This length is precisely $2\left(N-1\right) R$. Thus:

$$\mathcal{P}_{max} = \mathcal{P}_{geo} + 2\left(N-1\right) R, \tag{2.86}$$

as illustrated in Fig. 2.27. Invoking (2.84) and (2.85) gives us:

$$\mathcal{P}_{max} \leq \pi x_{\mathcal{C}_{N-1},r} + \frac{\pi}{2}\left(x_{\mathcal{C}_{N-1},r} + z_{\mathcal{C}_{N-1},r}\right) g_{\mathcal{C}_{N-1}} + 2\left(N-1\right) R. \tag{2.87}$$

This concludes the proof. □

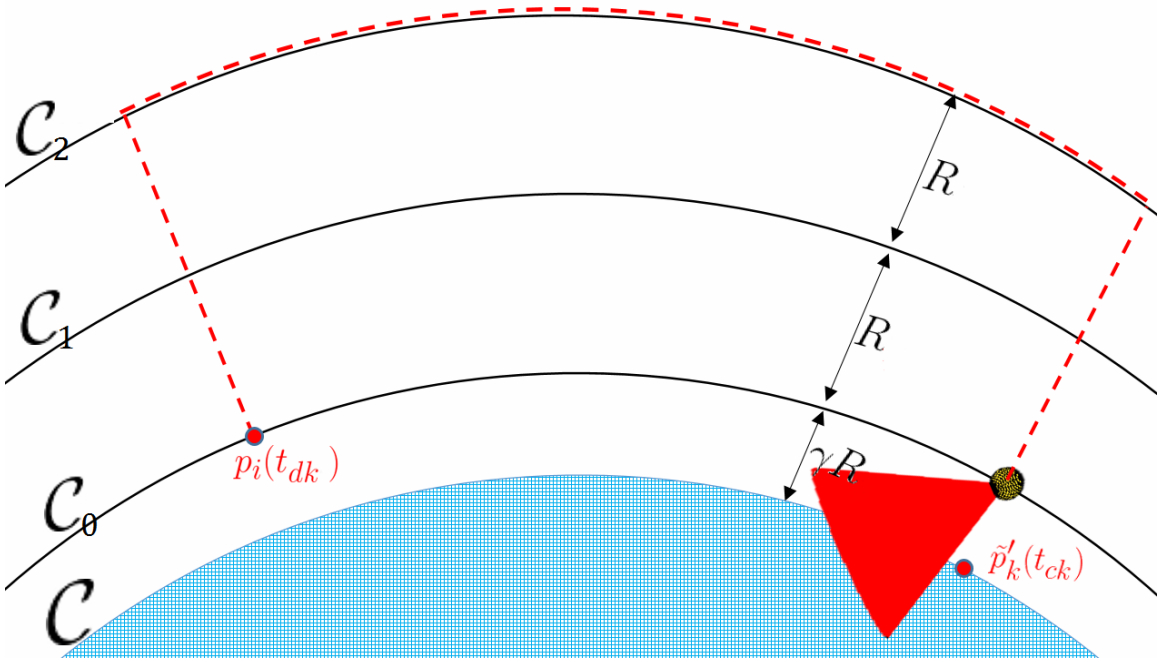


Figure 2.27: The longest possible path from $p_i(t_{dk})$ to $\tilde{p}'_k(t_{ck})$, taken by agent i assigned to intercept particle k, is illustrated above. Denote this path as $\mathcal{P}_{max}$. It may be upper bounded as established in Lemma 1.

Theorem 7. If the bounds on intruder velocity and range from detection to impact satisfy $\frac{R_{det}}{U^{int}_{max}} > \frac{\mathcal{P}_{max}}{U^{agt}_{max}}$, where $\mathcal{P}_{max}$ is bounded as in Lemma 1, agent i reaches $proj_{\mathcal{C}_0}\tilde{p}'_k(t_{ck})$ before t_{ck}.

Proof. Given that agents in the particle intercept and surface transfer modes travel at speed U^{agt}_{max}, it follows that the time t_{ik} required to travel from $p_i\left(t_{dk}\right)$ to $proj_{\mathcal{C}_0}\tilde{p}'_k(t_{ck})$ must satisfy: $t_{ik} \leq \frac{\mathcal{P}_{max}}{U^{agt}_{max}}$. Given that $t_{ck} - t_{dk} \geq \frac{R_{det}}{U^{int}_{max}}$, agent i reaching $proj_{\mathcal{C}_0}\tilde{p}'_k(t_{ck})$ before t_{ck} implies that $\frac{R_{det}}{U^{int}_{max}} > t_{ik}$. This is guaranteed if $\frac{R_{det}}{U^{int}_{max}} > \frac{\mathcal{P}_{max}}{U^{agt}_{max}}$ This concludes the proof. □

Remark 4. At any given time, there are at most $N-1$ agents available to intercept particles. Thus, satisfaction of Theorem 7 implies that the network is capable of intercepting all particles so long as a maximum of $N-1$ particle impacts occur in any moving time window

of:

$$\hat{T} = \frac{\pi x_{\mathcal{C}_{N-1},r} + \frac{\pi}{2}\left(x_{\mathcal{C}_{N-1},r} + z_{\mathcal{C}_{N-1},r}\right) g_{\mathcal{C}_{N-1}} + 2\left(N-1\right)R}{U_{max}^{agt}}.$$

2.3.4 Energy-aware Scheduling Protocol

2.3.4.1 Domain Partitioning

An agent deployment and scheduling protocol that realistically considers the agents' finite power and/or propulsive resources is now considered. The proposed strategy is to periodically deploy agents from a fueling station $\mathcal{F}$ that is assumed to be located at the North pole of $\mathcal{C}_0$, i.e., at the point $[0 \;\; 0 \;\; z_{\mathcal{C}_0,r}]^T$. Define $T^\star$ as the power lifespan of each agent in the network, i.e., $T^\star$ is the amount of time that an agent can operate upon deployment from $\mathcal{F}$. Given $T^\star$ and N, let us define a deployment and scheduling protocol such that one agent is deployed from $\mathcal{F}$ every $\frac{T^\star}{N}$ seconds. The initial deployment is that of agent $i = 1$ at $t = 0$ seconds with agent $i = 2$ following at $t = \frac{T^\star}{N}$. This continues indefinitely with the second deployment of agent $i = 1$ occurring at $t = T^\star$ seconds.

In order to distribute agents across $\mathcal{C}$, it is desirable to partition the domain and assign agents to monitor separate regions. Specifically, partitioning the domain by latitude, rather than longitude, ensures that agents are poised to intercept particles without the need for frequent crossings of the equator which tend to be associated with larger values of $\mathcal{P}_{geo}$ on an oblate spheroid.

Define the power index of agent i as $i_p(t) = 1 + \mod\left(i - 2 - \left\lfloor \frac{tN}{T^\star} \right\rfloor, N\right)$ where the first argument of the modulo operation is the dividend and the second argument is the divisor. The lower-bracketed delimiters represent the floored division operation. Upon deployment from $\mathcal{F}$, agent i has power index $i_p = N$ and this index is reduced by one every $\frac{T^\star}{N}$ seconds until $i_p = 1$, i.e., agent i is the power critical agent. Note that no two agents may share the same power index as a result of the periodic deployment and scheduling protocol.

Latitude partitions are characterized by a static upper bound in $\hat{z}_{\mathcal{G}}$ denoted $\bar{z}_{i_p-2}$ and a static lower bound $\bar{z}_{i_p-1}$. Rather than dynamically sizing partitions relative to agent power resources, let us divide partitions such that $N-1$ agents are assigned equal surface areas of $\mathcal{C}$ to explore. This choice maximizes the coverage of any individual partition as the allocation of a larger partition to a recently deployed agent would result in less effective coverage of that partition. Agents are sorted by their remaining power and transfer between partitions that are progressively closer to $\mathcal{F}$ as their power resource expires. Define the

surface area of the ellipsoid of revolution $\mathcal{C}$ as:

$$A_{\mathcal{C}} = 2\pi x_{\mathcal{C},r}^2 \left(1 + \frac{1 + \left(1 - \frac{z_{\mathcal{C},r}^2}{x_{\mathcal{C},r}^2}\right)}{\left(\sqrt{1 - \frac{z_{\mathcal{C},r}^2}{x_{\mathcal{C},r}^2}}\right)} \operatorname{artanh}\left(\sqrt{1 - \frac{z_{\mathcal{C},r}^2}{x_{\mathcal{C},r}^2}}\right) \right). \tag{2.88}$$

The agent with $i_p = 2$ is assigned to monitor the partition characterized by upper bound at north pole of $\mathcal{C}$, i.e., $\bar{z}_0 = z_{\mathcal{C},r}$. The lower bound $\bar{z}_1$ may be computed by dividing (2.88) by $(N-1)$, equating with the integral of ellipse cross sectional circumferences parametrized by $\tilde{z}$, and then numerically solving for $\bar{z}_1$:

$$\frac{A_{\mathcal{C}}}{N-1} = \int_{z_{\mathcal{C},r}}^{\bar{z}_1} 2\pi \sqrt{\left(x_{\mathcal{C},r}^2 - \frac{x_{\mathcal{C},r}^2 \tilde{z}^2}{z_{\mathcal{C},r}^2}\right)\left(1 + \frac{\tilde{z}^2 x_{\mathcal{C},r}^4}{x_{\mathcal{C},r}^2\left(z_{\mathcal{C},r}^4 - z_{\mathcal{C},r}^2 \tilde{z}^2\right)}\right)} d\tilde{z}. \tag{2.89}$$

One may then iteratively solve for the remaining bounds for increasing values of i_p up to $i_p = N-1$:

$$\frac{A_{\mathcal{C}}}{N-1} = \int_{\bar{z}_{i_p-2}}^{\bar{z}_{i_p-1}} 2\pi \sqrt{\left(x_{\mathcal{C},r}^2 - \frac{x_{\mathcal{C},r}^2 \tilde{z}^2}{z_{\mathcal{C},r}^2}\right)\left(1 + \frac{\tilde{z}^2 x_{\mathcal{C},r}^4}{x_{\mathcal{C},r}^2\left(z_{\mathcal{C},r}^4 - z_{\mathcal{C},r}^2 \tilde{z}^2\right)}\right)} d\tilde{z}. \tag{2.90}$$

The final computation of (2.90) for $i_p = N$ is not necessary as $\bar{z}_{N-1}$ is the south pole of $\mathcal{C}$, i.e., $\bar{z}_{N-1} = -z_{\mathcal{C},r}$, although this may be shown through numerical computation as well. The partitioning strategy for the case where $N = 4$ is presented in Fig. 2.28.

Note that no partition has been assigned to the agent for which $i_p = 1$. This is the power critical agent and it shall have flag $f_i := 1$ at the instant $i_p := 1$. The power critical agent cannot be assigned a new particle to intercept after $i_p := 1$ as this opens the possibility that particle assignment could occur near the end of the $\frac{T^\star}{N}$ time window during which time the agent with $i_p = 1$ should be transitioning back to $\mathcal{F}$ to exchange its power source. The power critical agent will instead spend the majority of this time window in local coverage mode assisting the other agents in gathering information. It can only be tasked with intercepting a particle if this assignment had occurred previously when $i_p = 2$. In this scenario, the agent should be capable of intercepting particle k and then transitioning back to $\mathcal{F}$ so long as a bound is established on the length of the deployment scheduling window $\frac{T^\star}{N}$.

Theorem 8. If agent power lifespan $T^\star$ satisfies $\frac{T^\star}{N} \geq t_{ck} - t_{dk} + \frac{\pi}{2U_{max}^{agt}}\left(x_{\mathcal{C}_0,r} + z_{\mathcal{C}_0,r}\right) g_{\mathcal{C}_0},\ \forall k$ then the agent with $i_p = 1$ shall always be capable of reaching $\mathcal{F}$ within $\frac{T^\star}{N}$ of the time at which $i_p := 1$.

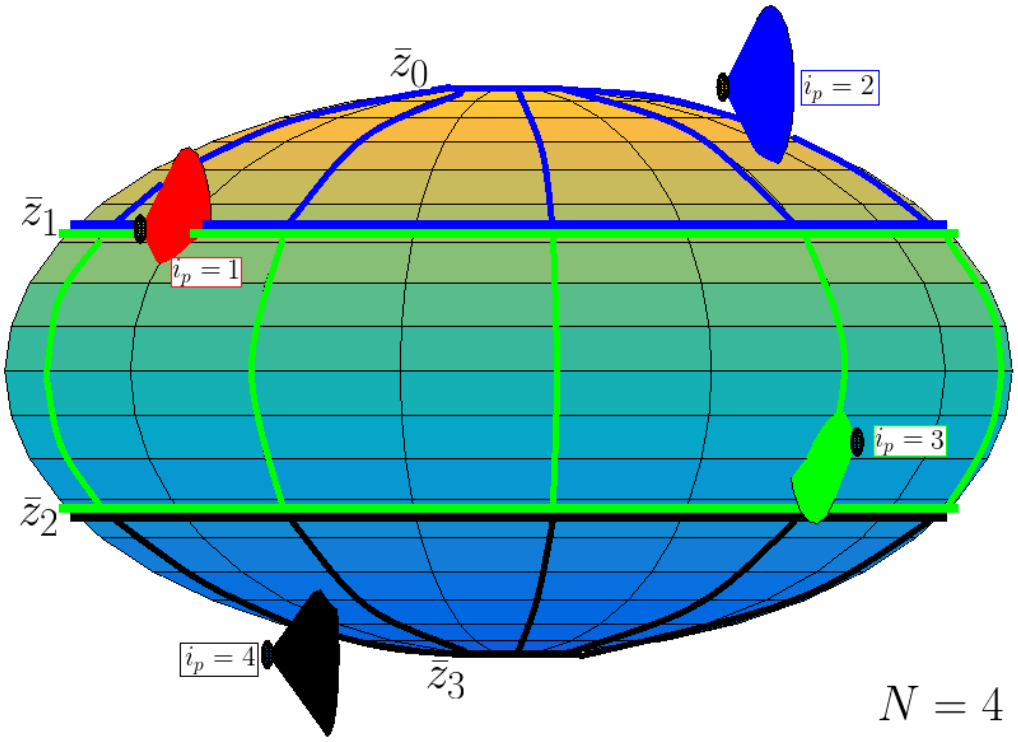


Figure 2.28: The domain partitioning scheme for $\mathcal{C}$ is illustrated above. Agents with $i_p \in \{2, 3, 4\}$ are indicated with blue, green and black $\mathcal{S}_i$ respectively. Their partitions are separated by latitude lines upper bounded at $\bar{z}_{i_p-2}$ and lower bounded at $\bar{z}_{i_p-1}$. The power critical has $\mathcal{S}_i$ indicated in red.

Proof. Consider the worst-case scenario in which the agent with $i_p = 2$ is assigned to intercept particle k at the instant before $i_p := 1$. Its remaining flight time is currently $\frac{T^\star}{N}$. The time required to intercept the particle is $t_{ck} - t_{dk}$, after which the control strategy dictates that the agent will follow a geodesic trajectory to $\mathcal{F}$. As $\mathcal{F}$ lies at the north pole of $\mathcal{C}_0$, this will be a trajectory of constant longitude which may be upper bounded by a length half the perimeter of the revolved ellipsoid: $\frac{\pi}{2}\left(x_{\mathcal{C}_0,r} + z_{\mathcal{C}_0,r}\right) g_{\mathcal{C}_0}$ by definition. As the agent is controlled by (2.79) with a North-pointing $\hat{\imath}_i$, it will proceed along this geodesic at speed U^{agt}_{max}. Thus the time required to complete this trajectory is $\frac{\pi}{2U^{agt}_{max}}\left(x_{\mathcal{C}_0,r} + z_{\mathcal{C}_0,r}\right) g_{\mathcal{C}_0}$ and one may bound the deployment window:

$$\frac{T^\star}{N} \geq t_{ck} - t_{dk} + \frac{\pi}{2U^{agt}_{max}}\left(x_{\mathcal{C}_0,r} + z_{\mathcal{C}_0,r}\right) g_{\mathcal{C}_0}, \ \forall k. \tag{2.91}$$

This concludes the proof. □

Remark 5. The appropriate design method for this surveillance system is to first ensure that the time from detection to impact of any arbitrary particle, $t_{ck} - t_{dk}$, as governed by

the omnidirectional range sensor satisfies Theorem 7. One must subsequently ensure that power lifespan $T^\star$, for all agents, satisfies Theorem 8.

2.3.4.2 Partition Transfer and Return to Base

If an agent with $i_p \in \{2, ..., N\}$ lies outside of its prescribed partition, and if $i_f = 0$, then the agent shall enter partition transfer mode. This mode uses the same geodesic position and orientation controllers (2.79) and (2.80) with the destination position set to the point:

$$p_{\star,i} = \begin{bmatrix} x_{\star,i} \\ y_{\star,i} \\ z_{\star,i} \end{bmatrix} = \begin{bmatrix} x_{\mathcal{C},r} \cos\left(\arcsin\left(\frac{z_{\star,i}}{z_{\mathcal{C},r}}\right)\right) \cos\Big(\operatorname{atan2}\big(y_i(t), x_i(t)\big)\Big) \\ y_{\mathcal{C},r} \cos\left(\arcsin\left(\frac{z_{\star,i}}{z_{\mathcal{C},r}}\right)\right) \sin\Big(\operatorname{atan2}\big(y_i(t), x_i(t)\big)\Big) \\ \bar{z}_{i_p-1}, \text{if } z_i < \bar{z}_{i_p-1}; \text{or } \bar{z}_{i_p-2}, \text{if } z_i > \bar{z}_{i_p-2} \end{bmatrix}, \tag{2.92}$$

i.e., the closest point along the agent's current longitude which lies on the boundary of its assigned partition.

The return to base mode is similar to partition transfer mode but is defined for the agent with $i_p = 1$. This mode is activated when the time since agent i's last deployment from $\mathcal{F}$, denoted $t_{i\mathcal{F}} \geq T^\star - \frac{\pi}{2U_{max}^{agt}} \left(x_{\mathcal{C}_0,r} + z_{\mathcal{C}_0,r}\right) g_{\mathcal{C}_0}$ as established in Theorem 8. The control strategy is the same as partition transfer mode with the desired position set to $\mathcal{F}$. Control laws for partition transfer mode and return to base shall be denoted with superscripts ptm and rtb respectively.

2.3.5 Surface Transfer Mode

The primary purpose of surface transfer mode is to encode collision avoidance, and it can be transitioned into from any other mode aside from the return to base mode. This mode is triggered for agent i, assigned to surface $\mathcal{C}_{\mu_i}$, under the condition that $\|p_i - p_j\| \leq R$ for $i \neq j$ is satisfied. Denote $\bar{j} = i \cup j$ as the set of agents satisfying this condition. Agents in $\bar{j}$ are ranked by $t_{\bar{j}\mathcal{F}}$. One agent, denoted i_{pr}, whose value for $t_{\bar{j}\mathcal{F}}$ is highest, i.e., $i_{pr} = \operatorname{argmax}_{\bar{j}} \left(t_{\bar{j}\mathcal{F}}\right)$ is permitted to proceed. The remaining agents increment their surface assignment indices, μ_i, by one and transition to surface transfer mode. This mode controls the agents to follow $\hat{n}_i$ until they have transferred to their newly assigned concentric surface at a height R above the previous. Note that in general, convexity of surface $\mathcal{C}$ is required to ensure that intersections of n_i and n_j, $\forall i \neq j$, lie within the interior space that is bounded by the surface.

The surface transfer position control strategy is:

$$\begin{bmatrix} u_i^{\mathcal{C},stm} \\ v_i^{\mathcal{C},stm} \\ w_i^{\mathcal{C},stm} \end{bmatrix} = U_{max}^{agt} \mathcal{R}_1^{-1} \frac{\ln\left(\frac{1}{(\gamma+\mu_i)R-\dot{z}_i}\left(\|n_i\| - \dot{z}_i\right)\right)\hat{n}_i}{\|\ln\left(\frac{1}{(\gamma+\mu_i)R-\dot{z}_i}\left(\|n_i\| - \dot{z}_i\right)\right)\hat{n}_i\|}. \tag{2.93}$$

As the agents ascend to a point at which R does not intersect $\mathcal{C}$, sensing information is not gathered in avoidance mode and thus the avoidance orientation control is simply $\begin{bmatrix} q_i^{\mathcal{C},stm} & r_i^{\mathcal{C},stm} & s_i^{\mathcal{C},stm} \end{bmatrix}^T = [0 \;\; 0 \;\; 0]^T$. Agents are said to have converged upon their newly assigned surface when:

$$|\ln \frac{\|n_i\| - \dot{z}_i}{(\gamma+\mu_i)\,R - \dot{z}_i}| < \varepsilon_5. \tag{2.94}$$

At this point, each agent shall transition back to its prior mode as described in the following two scenarios.

1. If agent i had been in either particle intercept or partition transfer mode before the deadlock, it shall resume that mode and continue along a geodesic trajectory on the newly assigned surface until it reaches the projection of its destination. At this point, the condition that $\|p_i - proj_{\mathcal{C}_{\mu_i}} p_{\star,i}\| \leq \varepsilon_4$ triggers a reset $\mu_i := 0$ concurrent with a transition back to surface transfer mode thus allowing the agent to transfer back to $\mathcal{C}_0$. The agent then resumes coverage of $\mathcal{C}_0$ in its prior mode. For additional details on flag conditions in these transitions, see guards $G\left(\zeta_{i2}, \zeta_{i4}\right), G\left(\zeta_{i4}, \zeta_{i2}\right), G\left(\zeta_{i3}, \zeta_{i4}\right)$, and $G\left(\zeta_{i4}, \zeta_{i3}\right)$ of the hybrid automaton as presented in the appendix.

2. If agent i had been in local coverage mode before the deadlock, it shall then transition back to local coverage mode concurrent with reset $\mu_i := 0$. This transition is dependent upon the conditions that $f_i = 0$ and that the agent is operating within its assigned partition. The agent shall oscillate between local coverage and surface transfer at an altitude of R above $\mathcal{C}_0$ until i_{pr} has moved along $\mathcal{C}_0$ to resolve the deadlock. At this point, the local coverage controller shall attract agent i back to the surface.

While similar work on multi-agent systems often invoke avoidance barrier functions to encode collision avoidance, such as in [90], it may be impossible to bound the time that agents spend avoiding one another in these maneuvers—especially when the algorithm is scaled to many agents. In contrast, the proposed technique results in an explicit bound on path length to an intruder as was proven in Lemma 1. With an additional assumption on the size of agents, one can establish a guarantee on collision avoidance for agents in surface transfer mode.

Theorem 9. For agents $\{i, j\} \in \bar{j}$, the sensing requirement that $R > 2\,\text{ɾ}_i + 2\,\text{ɾ}_j$ implies that i does not collide with j.

Proof. Consider the case in which $i \neq i_{pr}$ and $j \neq i_{pr}$. Both agents operate in accordance with (2.93) and follow trajectories along $\hat{n}_i$ and $\hat{n}_j$ respectively. Both unit vectors are normal to surface $\mathcal{C}_{\mu_i}$, an ellipsoid of revolution, and thus diverge from one another away from $\mathcal{C}_{\mu_i}$. Agents i and j shall enter surface transfer mode at an instant when $\|p_i - p_j\| \geq R$ and their distance shall tend to increase under (2.93). Thus $R > \text{ɾ}_i + \text{ɾ}_j$ and subsequently $R > 2\,\text{ɾ}_i + 2\,\text{ɾ}_j$ imply that they avoid collision.

Consider the case in which $i = i_{pr}$ and thus $j \neq i_{pr}$. In the instant that j transitions to surface transfer mode, it is true that $\|p_i - p_j\| \geq R$. Thus the distance for i to travel until collision is greater than or equal to $R - \text{ɾ}_i - \text{ɾ}_j$. This straight line path for i is a conservative estimate as the true path is curved. Collision will be avoided if agent j, whose path is normal to the surface, may cover a distance $\text{ɾ}_i + \text{ɾ}_j$ before i covers $R - \text{ɾ}_i - \text{ɾ}_j$. As j moves at speed U_{max}^{agt} and i's tangential speed is upper bounded by U_{max}^{agt}, this condition is satisfied if $R - \text{ɾ}_i - \text{ɾ}_j > \text{ɾ}_i + \text{ɾ}_j$. This may equivalently be written as $R > 2\,\text{ɾ}_i + 2\,\text{ɾ}_j$. These arguments apply to the case in which $j = i_{pr}$ and $i \neq i_{pr}$ as well. This concludes the proof. □

2.3.6 Simulations

A simulation was performed in MATLAB to verify the efficacy of the algorithm. Four agents are deployed to monitor the surface of an ellipsoid of revolution, $\mathcal{C}$, whose radius in the xy-plane is 80 and whose radius in the z-plane is 20. For each agent, $R = 10$, $\text{ɾ}_i = 1$, $\alpha_i = 30°$, $k_u = 1$, $k_v = 5$, $k_w = 1$, $k_r = 0.1$, $k_s = 0.1$, $\bar{r}_i = 0.4$, $\bar{s}_i = 0.4$, $U_{max}^{agt} = 6$, and $T^\star = 792$. Upon initialization of the simulation, $\mathcal{C}$ was set to a fully covered level of $C^\star = 20$ which would begin decaying upon detection of the first intruder $k \in \{1, ..., 4\}$ at $t = 600$ seconds. The four agents were deployed from $\mathcal{F}$ sequentially at times $t = 0$, $t = \frac{T^\star}{4}$, $t = \frac{2T^\star}{4}$, and $t = \frac{T^\star}{4}$ seconds respectively. Upon deployment, each agent was initialized in local coverage mode with $\Phi_i = 0$, $\Theta_i = \frac{\pi}{2}$, and $\Psi_i = 0$. Intruders traveled in random directions with $U_{max}^{int} = 0.7$, though were still constrained to always impact the surface, and were generated every 35 seconds beginning at $t = \frac{3T^\star}{4}$ seconds. The detection system had a lower bound on range $R_{det} = 80$, decay rate parameter $\lambda_k = 0.05$, and measurement variances $\sigma_\rho^2 = 0.0625$, $\sigma_\theta^2 = 0.25\,\text{deg}^2$, and $\sigma_\psi^2 = 0.25\,\text{deg}^2$ respectively Agents were able to successfully intercept all particles along their geodesic trajectories while actively avoiding collision over the entire duration of the attack (see Fig. 2.29 and Fig. 2.30); however, it should be noted that one avoidance anomaly occurred before the

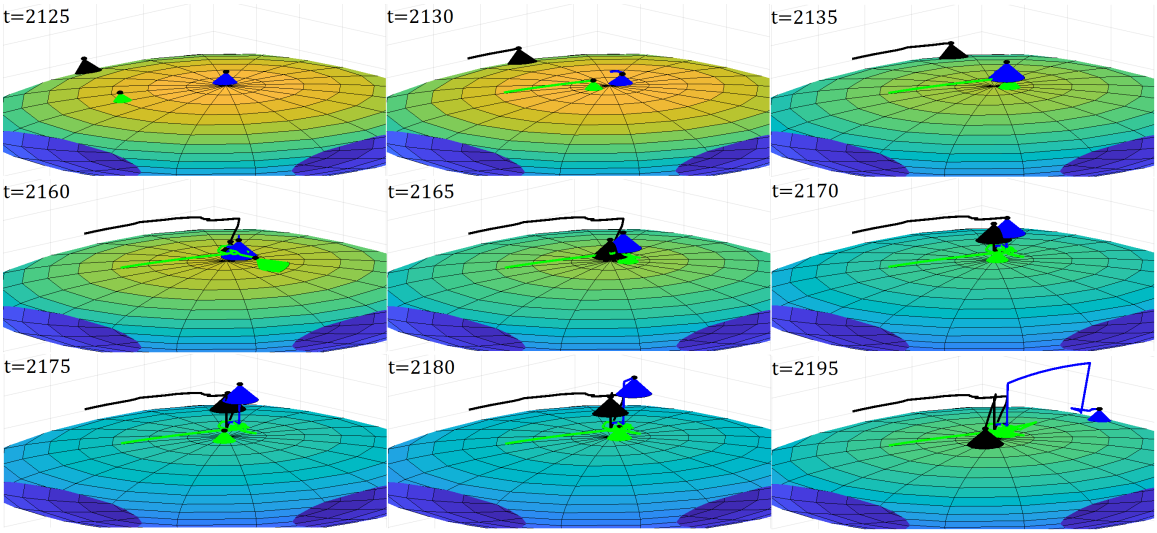


Figure 2.29: Agent $i = 3$, indicated with green $\mathcal{S}_i$, is on a collision course with agent $i = 1$, indicated with blue $\mathcal{S}_i$, during the interval from $t = 2125$ to $t = 2130$. At $t = 2135$, agent $i = 1$ has transitioned to surface transfer mode and is following a trajectory normal to the surface while agent $i = 4$, indicated with black $\mathcal{S}_i$, follows a collision course through $t = 2160$. Agent $i = 4$ transitions to surface transfer mode as well leading to the conditions that $\mu_1 = 2$ and $\mu_4 = 1$, i.e., agent $i = 1$ is assigned to the second tier of avoidance surfaces at a higher altitude than $i = 4$ as illustrated at $t = 2180$. Both agents proceed along their respective $\mathcal{C}_{\mu_i}$ towards their destination with $i = 1$ having arrived and transferred back to $\mathcal{C}_0$ before $t = 2195$. Note that agent trajectories for $t \geq 2125$ are plotted.

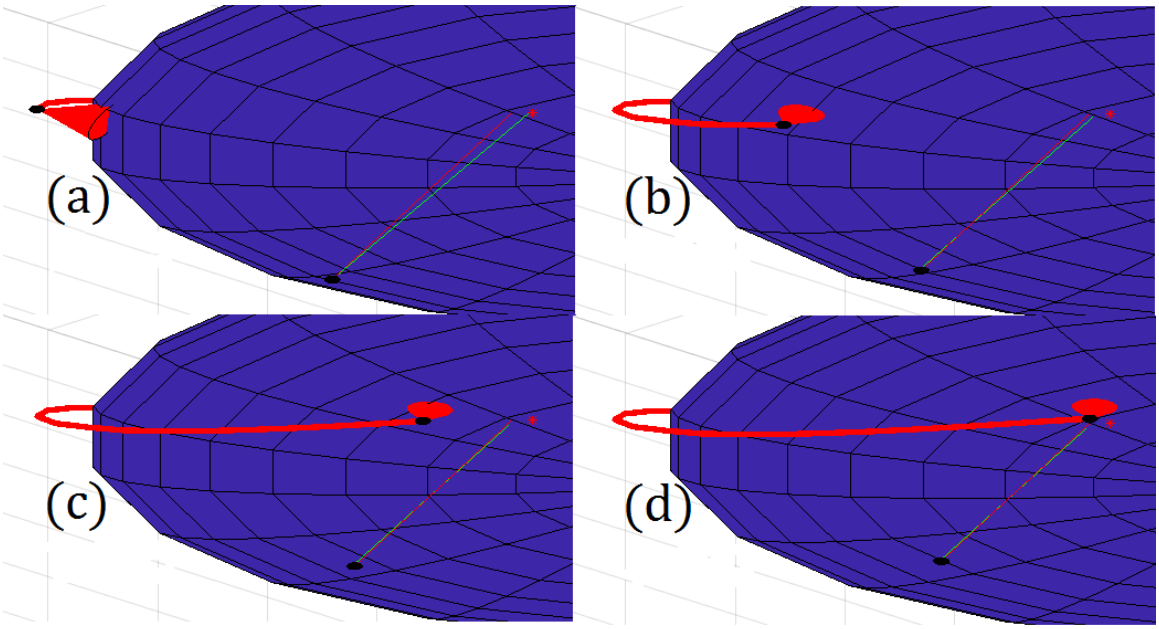


Figure 2.30: Agent $i = 2$ follows its geodesic trajectory to the predicted impact point of particle k over time lapse (a)-(d). The true trajectory of the particle is indicated in red and the estimated trajectory in green.

initial intruder was generated during the interval of $t = 418 - 420$. As the sensing range for agents was $R = 10$ with simulation time step size equal to one, it is clear that this anomaly occurred due to a selection of U_{max}^{agt} that was too large relative to the time step. In a continuous time implementation, a transition to surface transfer mode would have occurred between $t = 418$ and $t = 419$ thus preventing collision. Aside from this anomaly, the simulation parameters adequately approximated the continuous time agent kinematics.

The coverage error on $\mathcal{C}$, normalized with respect to the maximum error in which all of $\mathcal{C}$ takes a value of zero for $Q^{\mathcal{C}}(t, \tilde{p})$, as well as the minimum inter-agent distance over time are presented in Fig. 2.31. The error tends to spike upon particle detections with agents effectively curtailing these spikes as they cover around the vicinity of predicted impact points in local coverage mode. Two particularly large spikes occur at $t = 3225$ and $t = 5180$ respectively. These anomalies are each associated with particle impacts occurring close to the equator of the ellipsoid where even small values of σ_θ^2 and σ_ψ^2 result in an estimated particle trajectory that does not initially intersect $\mathcal{C}$ thus delaying an agent assignment. In both cases, the estimated trajectory did eventually intersect $\mathcal{C}$ with enough time to allow for agent interception. However, this delay in assignment significantly reduced the time the agent spent exploring in the vicinity of the predicted impact point thus contributing to a noticeable rise in the coverage error. One potential solution to this problem would be to prescribe some boundary tolerance to the surface $\mathcal{C}$ thus loosening the definition of an

impacting particle for the sake of measurement uncertainty.

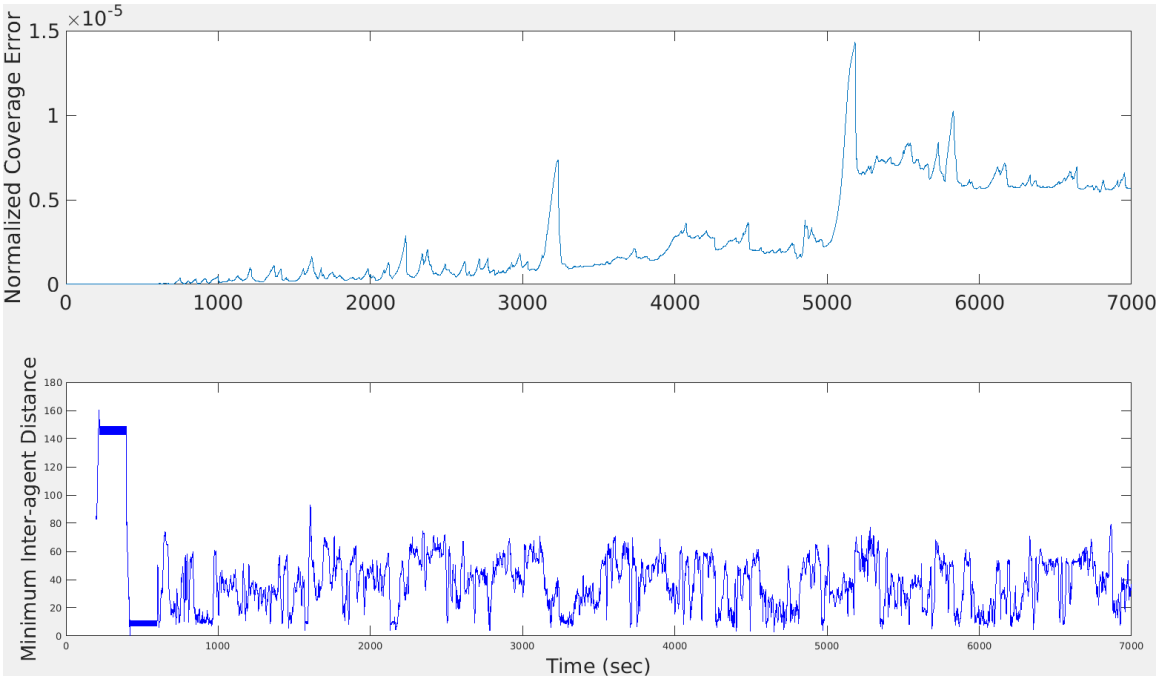


Figure 2.31: The coverage error, normalized to the maximum possible value, is presented. Anomalies are observed at $t = 3225$ and $t = 5180$ respectively. The minimum distance between any two agents at any given time is also presented with an anomaly observed at $t = 419$.

Agent $i = 1$'s operating modes with respect to time are presented in Fig. 2.32. It should be noted that the most frequent transition out of particle intercept mode is to local coverage mode. This corresponds to an agent arriving at the estimated impact point of a particle and then surveying the local area up until the moment of impact. As the agent surveys it tends to hit the ε_4 proximity boundary to the impact point thus requiring a short operation in particle intercept mode to direct the agent back within the ε_4 boundary.

To demonstrate scalability, consider an additional simulation with 100 agents available online: `https://1drv.ms/f/s!AsiVOlIEkwNEgX2o1eV2hJ_bbaQU`.

2.3.7 Conclusion

This section presented a hybrid formulation for the persistent coverage problem in an environment subject to stochastic intruders. This formulation was motivated in part by extravehicular applications of the NASA Mini AERCam. Agents operated with finite power resources and were required to periodically return to a refueling station while patrolling assigned latitude partitions along the surface of an ellipsoid. Formal guarantees were es-

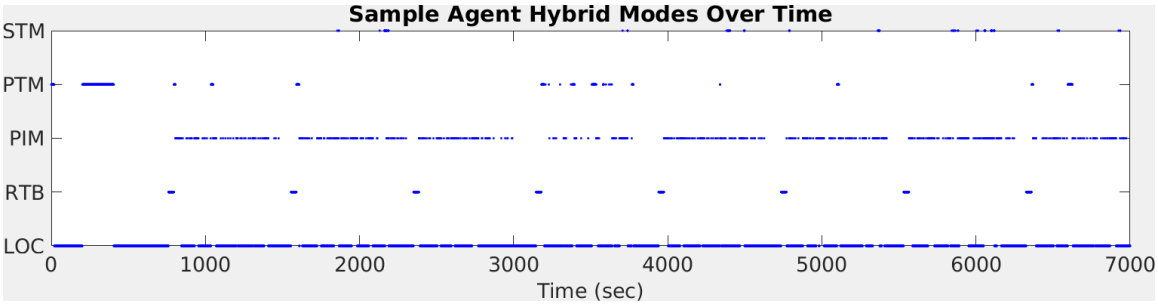


Figure 2.32: A typical agent's hybrid modes are presented over time. Abbreviations from top to bottom refer to surface transfer mode, partition transfer mode, particle intercept mode, return to base, and local coverage mode respectively.

tablished on the ability of agents to intercept all intruders in the face of an arbitrary number of collision avoidance maneuvers. The efficacy of the algorithm was demonstrated in simulation.

2.4 Preliminary Work in Gesture-based Control: Coverage Meets HRI

In this section, consider a brief extension of complete 3D coverage to a network consisting of a single agent supervised by a human via gesture commands. As in section 2.2, the goal is to achieve complete coverage of the domain $\mathcal{D}$ such that all points $\tilde{p} \in \mathcal{D}$ are covered to a satisfactory level $C^\star$. However, the human is now in charge of the high level planning, i.e., the human can direct the robot towards any region of the environment before releasing the robot to engage in local coverage. Agent i operates under the same kinematic and sensing model as section 2.2 and has state vector $\tilde{q}_i = \left[p_i^T, \Omega_i^T\right]^T$. The human's state is characterized by the position and orientation of their head, left hand, and right hand. Define these states as $\tilde{q}_H = \left[p_H^T, \Omega_H^T\right]^T$, $\tilde{q}_L = \left[p_L^T, \Omega_L^T\right]^T$, and $\tilde{q}_R = \left[p_R^T, \Omega_R^T\right]^T$ respectively as illustrated in Fig. 2.33.

The hybrid automaton for this system is composed of three states: local coverage mode (defined in 2.2), gesture following, and return to supervisor. The guard conditions which trigger transitions between these modes are entirely determined by the height of the human's hands relative to the height of their torso denoted z_T—nominally assumed to be approximately 0.9 meters. The agent nominally engages in local coverage mode so long as the human keeps both of their hands below their torso. Gesture following mode is triggered any time that the human raises their left hand above their torso. Return to supervisor is triggered under the condition that the human's right hand is above their torso while their

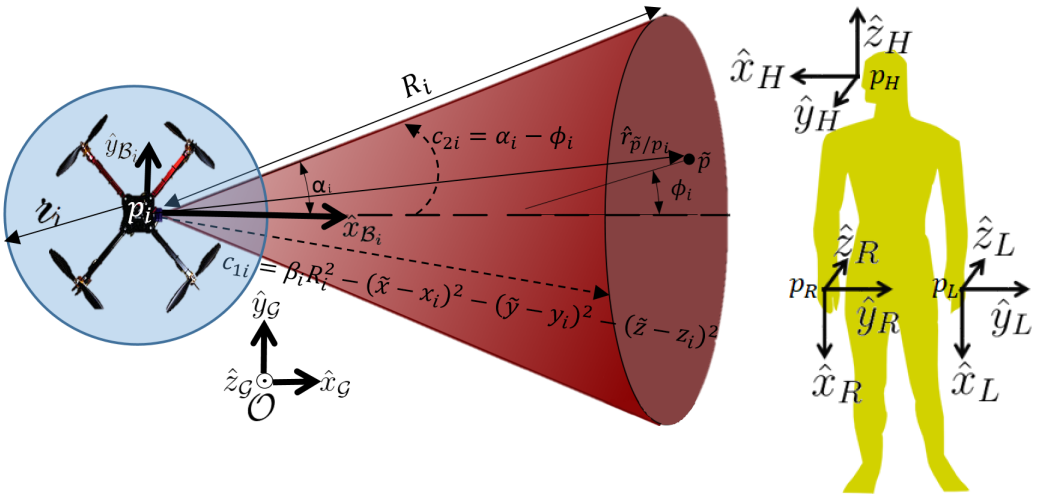


Figure 2.33: The sensing model of the agent, identical to that of section 2.2, is illustrated above along with reference frames denoting the configuration of the human supervisor.

left hand is below their torso. An illustration of the automaton is presented in Fig. 2.34.

The guard conditions controlling these transitions are as follows:

$$G\left(\zeta_{i1}, \zeta_{i0}\right) = G\left(\zeta_{i2}, \zeta_{i0}\right) = \{\tilde{q}_L, \tilde{q}_R \mid z_L < z_T \wedge z_R < z_T\}, \tag{2.95}$$

$$G\left(\zeta_{i0}, \zeta_{i1}\right) = G\left(\zeta_{i2}, \zeta_{i1}\right) = \{\tilde{q}_L \mid z_L \geq z_T\}, \tag{2.96}$$

$$G\left(\zeta_{i0}, \zeta_{i2}\right) = G\left(\zeta_{i1}, \zeta_{i2}\right) = \{\tilde{q}_L, \tilde{q}_R \mid z_L < z_T \wedge z_R \geq z_T\}. \tag{2.97}$$

The operating principle of the gesture following mode is for the agent to continuously increase the X component of its position relative to and resolved in the human's right hand frame while decreasing the Y and Z positions relative to and resolved in this frame. In this way, the human observes the robot converging upon and traveling down a vector extending forth from their pointed index finger. This is illustrated in Fig. 2.35.

Given v_x, the desired velocity of the agent, the control laws in gesture following mode take the following form:

$$\begin{bmatrix} u_i^{ges} \\ v_i^{ges} \\ w_i^{ges} \end{bmatrix} = \mathcal{R}_1^{-1}\left(\Omega_i\right) \mathcal{R}_1^{-1}\left(\Omega_R\right) \left(\begin{bmatrix} v_x \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathcal{R}_1\left(\Omega_R\right)\left(p_i - p_R\right) \right). \tag{2.98}$$

The operating principle of the return to supervisor mode is that the agent shall decrease its X and Y position relative to the human as resolved in the global coordinate frame. It shall do this to within a safety distance of r_s in the XY plane while tracking a desired height h_s. By switching back and forth between the gesture following and return to supervisor

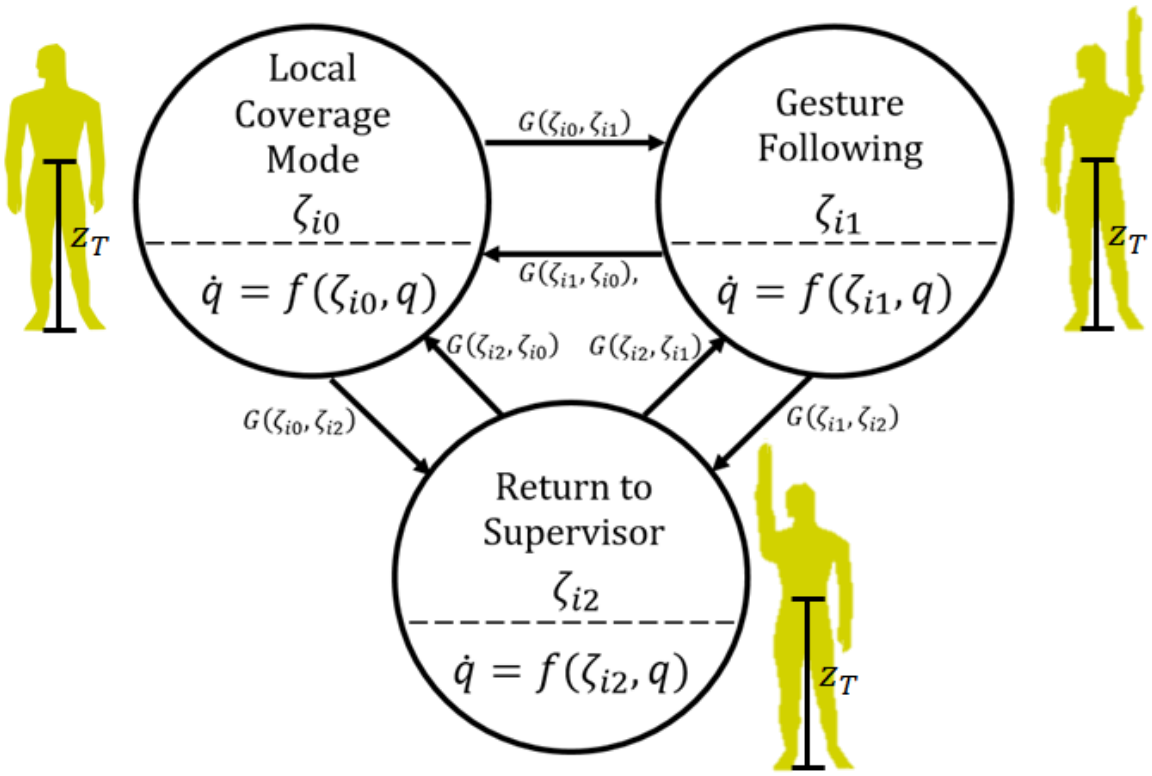


Figure 2.34: The agent's transitions are determined by the height of the human's hands.

modes, the human can quickly learn how to drive the agent to any desired position within the domain. The return to supervisor mode is illustrated in Fig. 2.36.

The control laws in this mode take the form:

$$\begin{bmatrix} u_i^{rts} \\ v_i^{rts} \\ w_i^{rts} \end{bmatrix} = \mathcal{R}_1^{-1} \begin{bmatrix} (x_H - x_i)\,\mathrm{sgn}\left(\sqrt{(x_H - x_i)^2 + (y_H - y_i)^2} - r_s\right) \\ (y_H - y_i)\,\mathrm{sgn}\left(\sqrt{(x_H - x_i)^2 + (y_H - y_i)^2} - r_s\right) \\ (h_s - z_i) \end{bmatrix}. \tag{2.99}$$

If provided with a live visual plot of $Q\,(t, \tilde{p})$, the human can quickly assess which portions of the domain to which they should direct the agent to explore. With practice, complete coverage can be achieved in this manner. This technique has been demonstrated within the DASC laboratory (see https://www.youtube.com/watch?v=feovNpLsQXs).

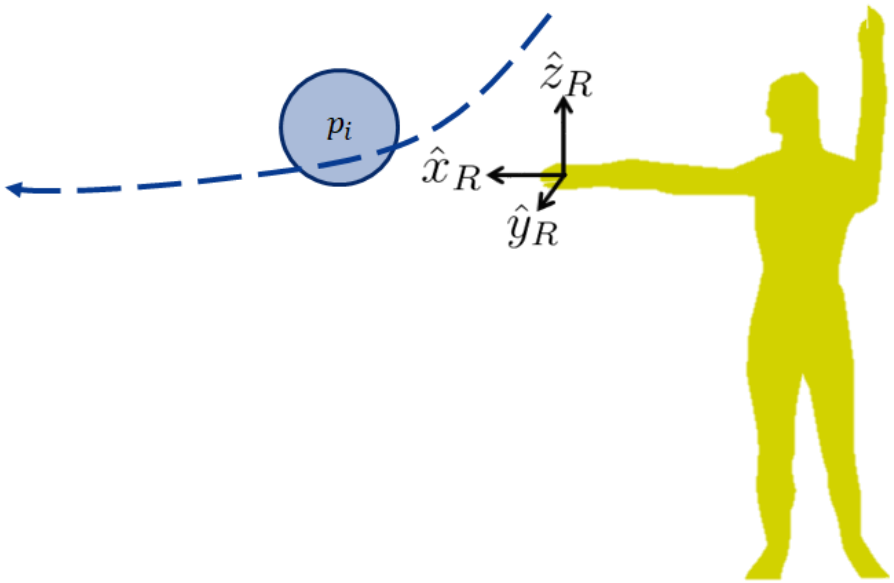


Figure 2.35: In the gesture following mode, the agent converges upon and flies along a vector which extends forward from the human's pointed index finger.

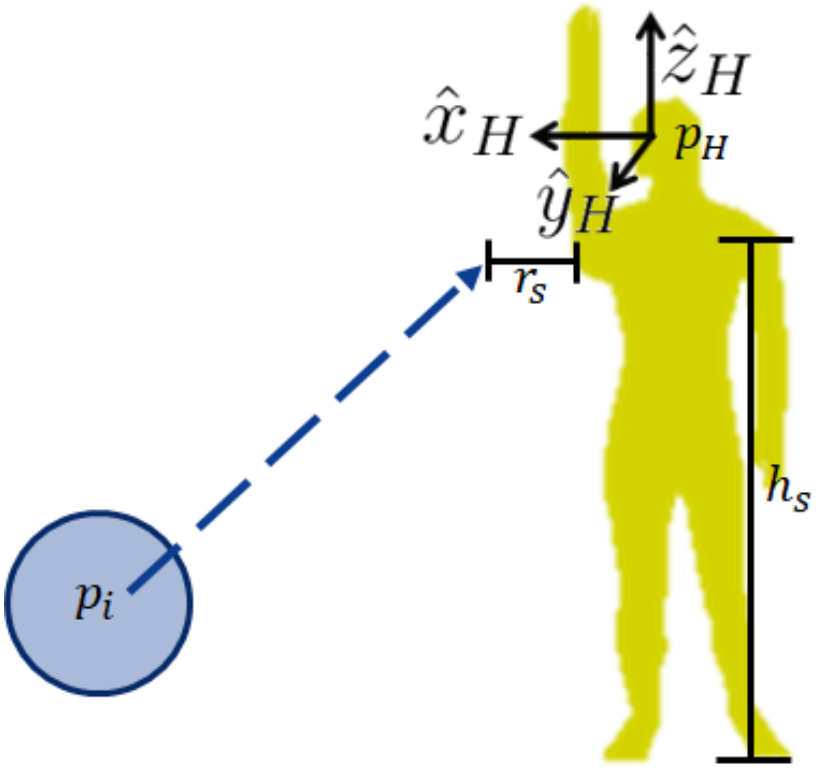


Figure 2.36: In the return to supervisor mode, the agent tracks a horizontal distance relative to the human of r_s while tracking a desired height of h_s.

CHAPTER 3

Human-Aerial/Space Robot Collaboration

This chapter presents the culmination of the author's work on human-aerial/space robot collaboration. The main idea is to extend certain concepts from dynamic coverage, introduced in Chapter 2, to a multi-agent network in which a human being is one of the agents. In this chapter, it is the behavior of the human that will directly impact the manner in which the robotic agent covers the environment with it's sensing footprint $\mathcal{S}_{\mathcal{R}}$.

In this chapter, the human's supervisory role is reduced as they now have their own objectives which they seek to accomplish. The robot must determine for itself which regions of space ought to be covered in an effort to increase the human's situational awareness. Specifically, a novel method is proposed whereby an aerial robot can learn sequences of task-relevant camera views within a multitasking environment. The robot learns these views by tracking the visual gaze of a human collaborator wearing an AR headset. The human's integrated visual gaze is fit to a Gaussian mixture model (GMM) via expectation maximization (EM). The modes of this model represent the visual interest regions of the environment. Using Q-learning, the robot is trained on which region it ought to photograph at any time given the human's most recent K tasks as represented by an MDP. Camera views are continuously streamed to the human's AR headset with the intent of increasing the human's multitasking performance while reducing physical and mental effort. An experimental study is presented in which 24 humans were asked to complete toy construction tasks while monitoring spatially separated flashing buttons. Subjects participated in four 2-hour sessions over multiple days. The MDP view selection system was also compared against control trials containing no assistance as well as supervised trials in which the subjects could directly command the robot to switch between views. The merits of this system were evaluated through both subjective measures, e.g., System Usability Scale (SUS) and NASA Task Load Index (TLX), as well as objective measures, e.g., task completion time, reflex time, and head angular velocity. This algorithm is applicable to multitasking environments that require persistent monitoring of regions outside of a human's (possibly restricted) field of view, e.g., spacecraft extravehicular activity.

The author wishes to acknowledge his co-authors Sahib Dhanjal and Long Qian who contributed portions of the software used in the experimental trials of section 3.2.

3.1 Notation

The following is a list of variables which frequently appear in this section:

- $\mathcal{D}$ - Domain in $\mathbb{R}^3$,
- $\mathcal{G}$ - Global Cartesian coordinate frame with axes $\hat{i}_{\mathcal{G}}$, $\hat{j}_{\mathcal{G}}$, and $\hat{k}_{\mathcal{G}}$,
- $\mathcal{I}$ - Set of visual regions of interest, containing elements $\mathcal{I}_i, \forall i \in \{1, ..., N\}$,
- N - Number of visual regions of interest,
- H - Human agent,
- R - Free-flying robotic agent,
- $\mathcal{S}_{\mathcal{H}}$ - The human's visual field,
- $\mathcal{S}_{\mathcal{R}}$ - The robot's sensing footprint,
- $\mathcal{H}$ - The human's visual field frame with axes $\hat{i}_{\mathcal{H}}$, $\hat{j}_{\mathcal{H}}$, and $\hat{k}_{\mathcal{H}}$ and with origin located directly in front of the human's eyes,
- $p_{\mathcal{H}} = [x_{\mathcal{H}}, y_{\mathcal{H}}, z_{\mathcal{H}}]^T$ - Origin of $\mathcal{H}$ and vertex of $\mathcal{S}_{\mathcal{H}}$,
- $\mathcal{R}$ - The robot's body-fixed frame with axes $\hat{i}_{\mathcal{R}}$, $\hat{j}_{\mathcal{R}}$, and $\hat{k}_{\mathcal{R}}$ and with origin located directly in front of the robot's camera,
- $p_{\mathcal{R}} = [x_{\mathcal{R}}, y_{\mathcal{R}}, z_{\mathcal{R}}]^T$ - Origin of $\mathcal{R}$ and vertex of $\mathcal{S}_{\mathcal{R}}$,
- $\alpha_{\mathcal{H}}$ - Half of the cone angle of $\mathcal{S}_{\mathcal{H}}$, i.e., half of the human's FOV angle,
- $\alpha_{\mathcal{R}}$ - Half of the cone angle of $\mathcal{S}_{\mathcal{R}}$,
- $\tilde{p}$ - Point in $\mathcal{D}$ relative to $\mathcal{G}$,
- $\phi_{\mathcal{H}}(\tilde{p})$ - Angle between $\tilde{p} - p_{\mathcal{H}}$ and $\hat{i}_{\mathcal{H}}$,
- $\phi_{\mathcal{R}}(\tilde{p})$ - Angle between $\tilde{p} - p_{\mathcal{R}}$ and $\hat{i}_{\mathcal{R}}$,
- $\tilde{q}_{\mathcal{H}} = [p_{\mathcal{H}}, \Phi_{\mathcal{H}}, \Theta_{\mathcal{H}}, \Psi_{\mathcal{H}}]^T$ - The position and orientation of $\mathcal{S}_{\mathcal{H}}$ where the latter three terms are 3-2-1 intrinsic Euler angles relating $\mathcal{H}$ relative to $\mathcal{G}$,

- $\tilde{q}_{\mathcal{R}} = [p_{\mathcal{R}}, \Phi_{\mathcal{R}}, \Theta_{\mathcal{R}}, \Psi_{\mathcal{R}}]^T$ - The position and orientation of $\mathcal{S}_{\mathcal{R}}$ where the latter three terms are 3-2-1 intrinsic Euler angles relating $\mathcal{R}$ relative to $\mathcal{G}$,
- $\mathcal{C}_{\mathcal{H}}$ - Configuration space of $\tilde{q}_{\mathcal{H}}$,
- $\mathcal{C}_{\mathcal{R}}$ - Configuration space of $\tilde{q}_{\mathcal{R}}$,
- $\sigma_{\mathcal{H}}$ - Standard deviation of truncated Gaussian modeling human visual acuity,
- $S_{\mathcal{H}}\left(\tilde{q}_{\mathcal{H}}, \tilde{p}\right)$ - Model for human visual acuity,
- $S_{\mathcal{R}}\left(\tilde{q}_{\mathcal{R}}, \tilde{p}\right)$ - Model for robot quality of sensing,
- $\mathcal{P}\left(\tilde{q}_{\mathcal{H}}, \tilde{p}\right)$ - Point cloud data returned from AR headset,
- $W\left(t, \tilde{p}\right)$ - Visual acuity-weighted accumulated point cloud,
- $\tilde{P}$ - Set of points forming a discretization of $\mathcal{D}$,
- M - Defined as the cardinality of $\tilde{P}$,
- T - Set of M samples from the uniform distribution in $(0, 1)$,
- t_r - Time of resampling,
- W_c - The cumulative sum of the normalized elements of $W\left(t_r, \tilde{P}\right)$,
- $[i]$ - Used to denote the i^{th} element of an array,
- $\tilde{P}_f$ - The array of filtered points output by Algorithm 3.1,
- $\mathcal{N}\left(\mu_i, \Sigma_i\right)$ - Denotes the normal distribution with mean μ_i and covariance Σ_i,
- $\bar{\psi}_{\mathcal{H}}$ - The human's visual interest function—a mixture of Gaussians,
- μ_i - Set of N means output from EM-GMM solver,
- Σ_i - Set of N covariance matrices output from EM-GMM solver,
- π_i - Set of N mixing coefficients output from EM-GMM solver,
- d_s - Minimum safe distance between R and H,
- $\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i}$ - The optimal configuration of $\mathcal{S}_{\mathcal{R}}$ for observation of $\mathcal{I}_i$,
- $\mathcal{I}_{i^\star}$ - The element of $\mathcal{I}$ that is most intersecting H's visual gaze at the present time,

- $i^\star$ - The index of the element of $\mathcal{I}$ most intersecting H's visual gaze at the present time,
- K - The robot's running memory length of recent human tasks in sequence,
- k - indexes the discrete events at which the human switches their gaze between elements of $\mathcal{I}$,
- s_k - A sequence of the most recent K elements of $\mathcal{I}$ observed by the human,
- A - The action set associated with the proposed MDP,
- a_k - The k^{th} action taken from A,
- r_k - The reward issued to the R in accordance with Q-learning,
- h_k - The human input signal portion of r_k which takes value in $\{-1, 0, 1\}$,
- f_k - The transition time reward portion of r_k,
- η - Tunable parameter to balance the influence of f_k relative to h_k,
- t_k^m - The time spent maneuvering $\mathcal{S}_\mathcal{R}$ to its k^{th} desired configuration,
- t_k^o - The time $\mathcal{S}_\mathcal{R}$ spends in its k^{th} desired configuration before receiving its next command,
- ε - The probability that action a_k will be random (defines the ε-Greedy policy),
- α - Learning rate in Q-learning,
- γ - Discount factor in Q-learning,
- $Q(s_k, a_k)$ - Q-value representing quality of a state-action combination.

3.2 Online Learning of Assistive Dynamic Camera Views by an Aerial Co-robot in Augmented Reality Multitasking Environments

3.2.1 System Model

Consider a domain $\mathcal{D} \subset \mathbb{R}^3$ that contains a set $\mathcal{I}$ of N visual regions of interest denoted $\mathcal{I}_i, \forall i \in \{1, ..., N\}$. These regions may contain individual tasks in the multitasking en-

vironment as well as any relevant information sources. A human H, moving in $\mathcal{D}$ and wearing an AR headset containing a depth camera, must split their attention among all the elements of $\mathcal{I}$ in order to achieve abstract multitasking objectives. A free-flying robot R, also moving in $\mathcal{D}$, is tasked to stream live camera views of $\mathcal{I}_i$ to the headset, in order to increase the efficiency by which H achieves their abstract objectives. H has access to two buttons which provide reward and punishment signals respectively to R in order to train its selections from $\mathcal{I}_i$. Note that while H is required to provide R with a value for N, the precise locations of $\mathcal{I}_i$ are unknown *a priori*. The system layout for the $N = 3$ task environment considered in the experimental study is illustrated in Fig. 3.1.

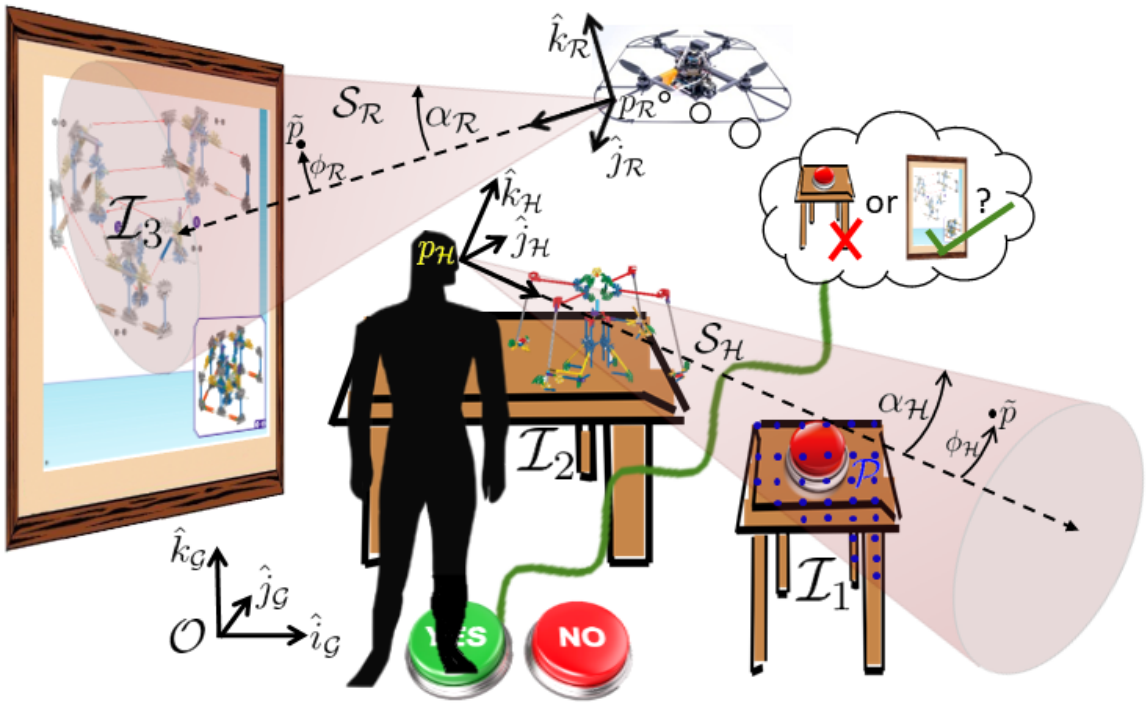


Figure 3.1: In this sample scenario, H is engaged in an assembly task within an environment containing a set of 3 visual interest regions $\mathcal{I}$. H provides reinforcement to R which must observe the appropriate $\mathcal{I}_i$ at any time.

3.2.1.1 Human Visual Acuity Model

Let us model the visual field of the human as a spherical sector $\mathcal{S}_{\mathcal{H}}$ with cone angle $2\alpha_{\mathcal{H}}$ and infinite radius. The vertex of this sector is located at $p_{\mathcal{H}} = [x_{\mathcal{H}} \;\; y_{\mathcal{H}} \;\; z_{\mathcal{H}}]^T$ relative to the origin $\mathcal{O}$ of a global Cartesian coordinate frame $\mathcal{G}$. Note that all position vectors in the sequel are similarly taken relative to $\mathcal{O}$ and resolved in $\mathcal{G}$. Define the visual field frame $\mathcal{H}$ as having origin $p_{\mathcal{H}}$ and $\hat{i}_{\mathcal{H}}$ axis extending through the centerline of the sector. As $\mathcal{S}_{\mathcal{H}}$ is

radially symmetric about $\hat{i}_{\mathcal{H}}$, $\hat{j}_{\mathcal{H}}$ may be chosen as any unit vector orthogonal to $\hat{i}_{\mathcal{H}}$ and $\hat{k}_{\mathcal{H}}$ completes the right-handed frame. The pose of $\mathcal{S}_{\mathcal{H}}$ is thus $\tilde{q}_{\mathcal{H}} = [p_{\mathcal{H}} \;\; \Phi_{\mathcal{H}} \;\; \Theta_{\mathcal{H}} \;\; \Psi_{\mathcal{H}}]^T$ where the latter 3 states refer to the 3-2-1 intrinsic Euler angles defining the orientation of frame $\mathcal{H}$ relative to $\mathcal{G}$. $\tilde{q}_{\mathcal{H}}$ is defined over the configuration space $\mathcal{C}_{\mathcal{H}} \subset \mathbb{R}^6$ such that $p_{\mathcal{H}} \in \mathcal{D}$, $\Phi_{\mathcal{H}} \in (-\pi, \pi)$, $\Theta_{\mathcal{H}} \in \left(\frac{-\pi}{2}, \frac{\pi}{2}\right)$, and $\Psi_{\mathcal{H}} \in (-\pi, \pi)$.

Human visual acuity is anisotropic in nature, e.g., it tends to degrade in quality towards one's periphery. Therefore, assuming that visual regions lying along $\hat{i}_{\mathcal{H}}$ are of the greatest interest, define the model for visual acuity as:

$$S_{\mathcal{H}}\left(\tilde{q}_{\mathcal{H}}, \tilde{p}\right) = \frac{\exp\left(-\frac{\phi_{\mathcal{H}}(\tilde{p})^2}{2\sigma_{\mathcal{H}}^2}\right)}{\sqrt{2\pi}\sigma_{\mathcal{H}}\left(\frac{\alpha_{\mathcal{H}}}{\sigma_{\mathcal{H}}\sqrt{2}}\right)}, \tag{3.1}$$

where $\tilde{p} = [\tilde{x} \;\; \tilde{y} \;\; \tilde{z}]^T$ is the position of a point in $\mathcal{D}$ relative to $\mathcal{G}$ and $\phi_{\mathcal{H}}\left(\tilde{p}\right)$ is angle between $\tilde{p} - p_{\mathcal{H}}$ and $\hat{i}_{\mathcal{H}}$. Let us define (3.1) using the probability density function associated with a zero-mean truncated normal distribution taking nonzero values between $\pm\alpha_{\mathcal{H}}$ and whose underlying Gaussian has standard deviation $\sigma_{\mathcal{H}}$. The parameters of the truncated Gaussian distribution are chosen as $\alpha_{\mathcal{H}} = 60^\circ$ and $\sigma_{\mathcal{H}} = 8^\circ$ since they provide a relatively good fit for the clinically-inspired plots of visual acuity with respect to angle from fovea that are presented in sensory physiology textbooks [96, 97]. Furthermore, this model enforces that visual acuity is zero beyond the periphery bound.

Let us assume the AR headset contains a depth camera which returns point cloud data $\mathcal{P}\left(\tilde{q}_{\mathcal{H}}, \tilde{p}\right)$. $\mathcal{P}\left(\tilde{q}_{\mathcal{H}}, \tilde{p}\right)$ has a FOV lower bounded by $\alpha_{\mathcal{H}}$ and takes a value of 1 for $\tilde{p}$ intersecting physical objects within the FOV and 0 otherwise.

As H shifts their attention between the elements of $\mathcal{I}$, the regions of space intersecting $\mathcal{S}_{\mathcal{H}}$ and $\mathcal{P}$ vary. Bearing this in mind, define the acuity-weighted accumulated point cloud:

$$W\left(t, \tilde{p}\right) = \int_0^t S_{\mathcal{H}}\left(\tilde{q}_{\mathcal{H}}\left(\tau\right), \tilde{p}\right) \mathcal{P}\left(\tilde{q}_{\mathcal{H}}\left(\tau\right), \tilde{p}\right) d\tau, \tag{3.2}$$

where the time dependence of $\tilde{q}_{\mathcal{H}}$, nominally omitted to reduce clutter, is notated here for clarity. (3.2) physically represents the human's accumulation of visual information along surfaces of physical objects within the domain.

3.2.1.2 Robot Kinematic and Sensor Model

Let us assume that R is equipped with a camera whose sensing footprint is a spherical sector denoted $\mathcal{S}_{\mathcal{R}}$ which is similar in shape, though not in quality of sensing, to $\mathcal{S}_{\mathcal{H}}$. In defining $\mathcal{S}_{\mathcal{R}}$'s motion, let us adopt the same robot kinematic and sensor model as Chapter 2. Thus,

the state vector for the motion model of $\mathcal{S}_\mathcal{R}$ is defined as $\tilde{q}_\mathcal{R} = [p_\mathcal{R} \;\; \Phi_\mathcal{R} \;\; \Theta_\mathcal{R} \;\; \Psi_\mathcal{R}]^T$. The configuration space $\mathcal{C}_\mathcal{R}$ of $\tilde{q}_\mathcal{R}$ is defined with robot states constrained in the same manner as those of the human in $\mathcal{C}_\mathcal{H}$. In further analogy to Chapter 2, the quality of sensing $S_\mathcal{R}(\tilde{q}_\mathcal{R}, \tilde{p})$ is defined in the same manner as (2.5).

3.2.2 Approach

3.2.2.1 Online Learning of Relevant Views

One cannot draw immediate conclusions with respect to the locations of $\mathcal{I}_i$ directly from integration of raw visual acuity data in (3.2). Instead, some form of online clustering is necessary in order to guide the motion of R. A natural choice for clustering is EM of a GMM. This approach allows for the raw data, upon resampling, to be converted into a mixture of Gaussians whose gradient is smooth and defined over all $\mathcal{D}$.

Let us assume that values for $W(t, \tilde{p})$ are computed at a finite number of discrete points $\tilde{P} \subset \mathcal{D}$ on a three-dimensional grid. Consider the values of W for each element of $\tilde{P}$, i.e., each sample of $\tilde{P}$, to be an *importance weight* associated with that *sample*. This terminology is consistent with descriptions of the importance resampling step of the particle filter as presented in [98]. Using importance resampling, elements of $\tilde{P}$ are selected at random where the likelihood of selection is proportional to the weight of the sample. The selected set of points, which can include multiple copies of individual points, forms an appropriate input for EM-GMM.

Let us employ the *select with replacement* algorithm as it is among the most common resampling methods used in particle filters [99]. The algorithm generates $M = card\left(\tilde{P}\right)$ samples from a uniform distribution in $(0, 1)$. This set, T, is sorted into ascending order and then augmented with an additional element equal to 1. A second array, W_c, is then defined as the cumulative sum of the normalized elements of $W\left(t_r, \tilde{P}\right)$ where t_r is the time of resampling. Starting with $W_c[1]$, each ascending element of T is compared against $W_c[1]$ with $\tilde{P}[1]$ selected for output repeatedly until an element of T is found that exceeds $W_c[1]$. The process resumes with $W_c[2]$ until M points have been chosen. The effect is that elements of $\tilde{P}$ with small values for W, i.e., points which contain negligible visual interest to the human, are filtered out. Importance resampling is substantially more tractable than the alternative approach, i.e., to take one sample of the point associated with the smallest value of $W\left(t_r, \tilde{P}\right)$ and proportionally repeated samples of all other points associated with $W\left(t_r, \tilde{P}\right)$. The resampling implementation is described in full detail in Algorithm 3.1 where the array of filtered points is denoted $\tilde{P}_f$.

The EM-GMM implementation requires N and $\tilde{P}_f$ as inputs in order to estimate the

Algorithm 3.1 Select with Replacement

Inputs: $\tilde{P}, W\left(t, \tilde{P}\right)$

Initialize: $i \leftarrow 1, j \leftarrow 1,\ M \leftarrow card\left(\tilde{P}\right)$

while $i < M + 1$ **do**

  if $i < 2$ **then**

    $W_c[i] \leftarrow \frac{W\left(t_r, \tilde{P}[i]\right)}{\sum_{i=1}^{M} W\left(t_r, \tilde{P}[i]\right)},\ T[i] \leftarrow rand\,(0, 1)$

  else

    $W_c[i] \leftarrow A_c[i-1] + \frac{W\left(t_r, \tilde{P}[i]\right)}{\sum_{i=1}^{M} W\left(t_r, \tilde{P}[i]\right)}$

    $T[i] \leftarrow rand\,(0, 1)$

  end if

end while

$T \leftarrow sort\,(T)\,,\ T[M+1] \leftarrow 1.0,\ i \leftarrow 1$

while $i < M + 1$ **do**

  if $T[i] < A_c[j]$ **then**

    $\tilde{P}_f[i] \leftarrow \tilde{P}[j],\ i \leftarrow i + 1$

  else

    $j \leftarrow j + 1$

  end if

end while

return $\tilde{P}_f$

human visual interest function:

$$\bar{\psi}_{\mathcal{H}} = \sum_{i=1}^{N} \pi_i \mathcal{N}\left(\mu_i, \Sigma_i\right), \tag{3.3}$$

where π_i, μ_i, and Σ_i are the mixing coefficients, means, and covariances outputted from the EM-GMM solver. A full description of EM-GMM is available for reference in [100]. An example $\bar{\psi}$ associated with the task scenario in Fig. 3.1 is presented in Fig. 3.2. The

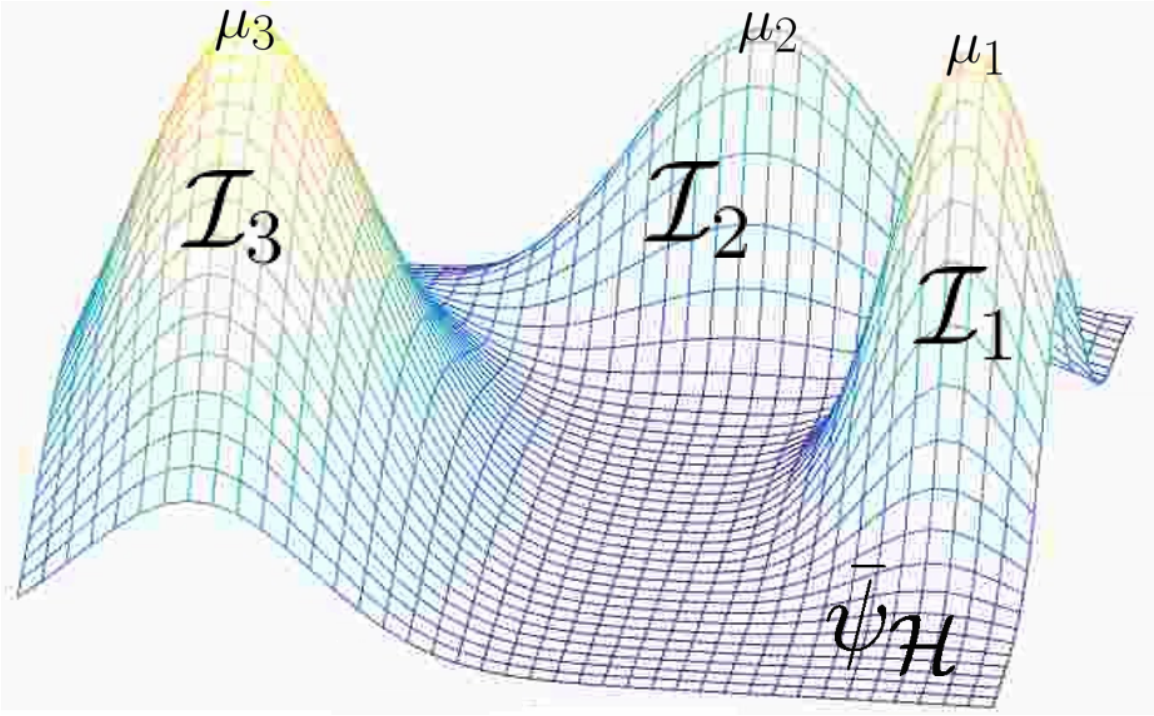


Figure 3.2: This mixture of Gaussians corresponds to a cross section of the task scenario presented in Fig. 3.1 with mixing coefficients $\pi_1 \approx 0.35$, $\pi_2 \approx 0.30$, and $\pi_3 \approx 0.35$.

optimal configuration of $\mathcal{S}_{\mathcal{R}}$ to observe $\mathcal{I}_i$ is the configuration at which the highest-quality portion of the robot's sensing footprint intersects the location of $\mathcal{I}_i$ subject to the safety constraint that H and R maintain a distance of greater than d_s. This is found via the following constrained minimization problem:

$$\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i} = \min_{\tilde{q}_{\mathcal{R}} \in \mathcal{C}_{\mathcal{R}} : \|p_{\mathcal{R}} - p_{\mathcal{H}}\| > d_s} \Big(\int_{\mathcal{D}} \big(S_{\mathcal{R}}\left(\tilde{q}_{\mathcal{R}}, \tilde{p}\right) - \mathcal{N}\left(\mu_i, \Sigma_i\right) \big) d\tilde{p} \Big). \tag{3.4}$$

If R were to select $\mathcal{I}_i$ for observation at any given time, it can compute the optimal configuration for viewing $\mathcal{I}_i$ via (3.4). The robot will then assume this configuration. For this implementation, a waypoint-based proportional-integral-derivative (PID) controller was

chosen to accomplish this objective. The topic of the following section is the logic by which R shall select $\mathcal{I}_i$ provided the recent history, i.e., temporal context, of human actions.

3.2.2.2 Sequential View Selection

At any given time, assume that the attention of H is focused on one element of $\mathcal{I}$, denoted $\mathcal{I}_{i^\star}$, where:

$$i^\star = \underset{i \in \{1,\ldots,N\}}{\operatorname{argmin}} \int_{\mathcal{D}} \left(S_{\mathcal{H}} \left(\tilde{q}_{\mathcal{H}}, \tilde{p} \right) - \mathcal{N} \left(\mu_i, \Sigma_i \right) \right) d\tilde{p}, \tag{3.5}$$

i.e., $\mathcal{I}_{i^\star}$ is the visual interest region most intersecting H's gaze. Consider the changes in H's attention to be discrete events in time at which the solution of (3.5) changes and is unique. Each event is indexed k with the most recent K events denoted by sequence $s_k = \left(\mathcal{I}_{i_k^\star}, \mathcal{I}_{i_{k-1}^\star}, ..., \mathcal{I}_{i_{k-K+1}^\star} \right)$. s_k contains the temporal context of H's attention state.

Let us adopt the sequence s_k of the K most recent human tasks as a discrete state of the proposed MDP. It follows that the set of all discrete states S has a length equal to the number of K-permutations of $\mathcal{I}$ with non-consecutive repetition allowed: $card\left(S\right) = N\left(N-1\right)^{K-1}$, i.e., there are N possibilities for the first element of the sequence and $N-1$ possibilities for the remaining $K-1$. For the user study, the author considered $N = 3$ as well as $N = 4$ tasks with memory length $K = 3$, i.e., $card\left(S\right) = 12$ and $card\left(S\right) = 36$ respectively. This value of K is large enough to allow for sequences containing all N tasks within individual MDP states. Furthermore, this choice is small enough such as to allow for exploration of the entire state-action space within a reasonable time. Larger values for K will result in greater temporal context with respect to H's attention but will require a greater number of training trials to converge upon the optimal state-action policy.

The author found through the user study (presented in section 3.2.4) that the choices of $N = 4$ and $K = 3$ resulted in approximately 90% of the state-action space having been explored after 6-7 training sessions—which was reasonable given the constraints of the experimental design. The choice of $K = 3$ follows from an assumption that the human is only required to mentally consider a maximum of three tasks at any given time, e.g., their mental image of the task instructions, their physical manipulation of the task puzzles, and perhaps an ancillary mental query as how long ago they last checked the buzzers. In general, K ought to be chosen as equal to the maximum number of interdependent tasks of mental relevance to the human at any given time. Considering that the short term memory of humans is limited to approximately 7 pieces of information [101], the author suggests that K ought to be upper bounded by 7 within any environment. A choice of K greater

than 7 will tend to overspecialize the system as a multitude of states will be redundant in any conceivable task environment.

Define the action set associated with the proposed MDP as $A = \{\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i}\}, \forall i \in \{1, ..., N\}$, and denote the k^{th} action as a_k. The action set encodes that upon any transition of human attention the robot will translate and orient itself such that $\mathcal{S}_{\mathcal{R}}$ becomes coincident with one of the optimal camera views defined in (3.4). One MDP associated with the considered user study is presented in Fig. 3.3.

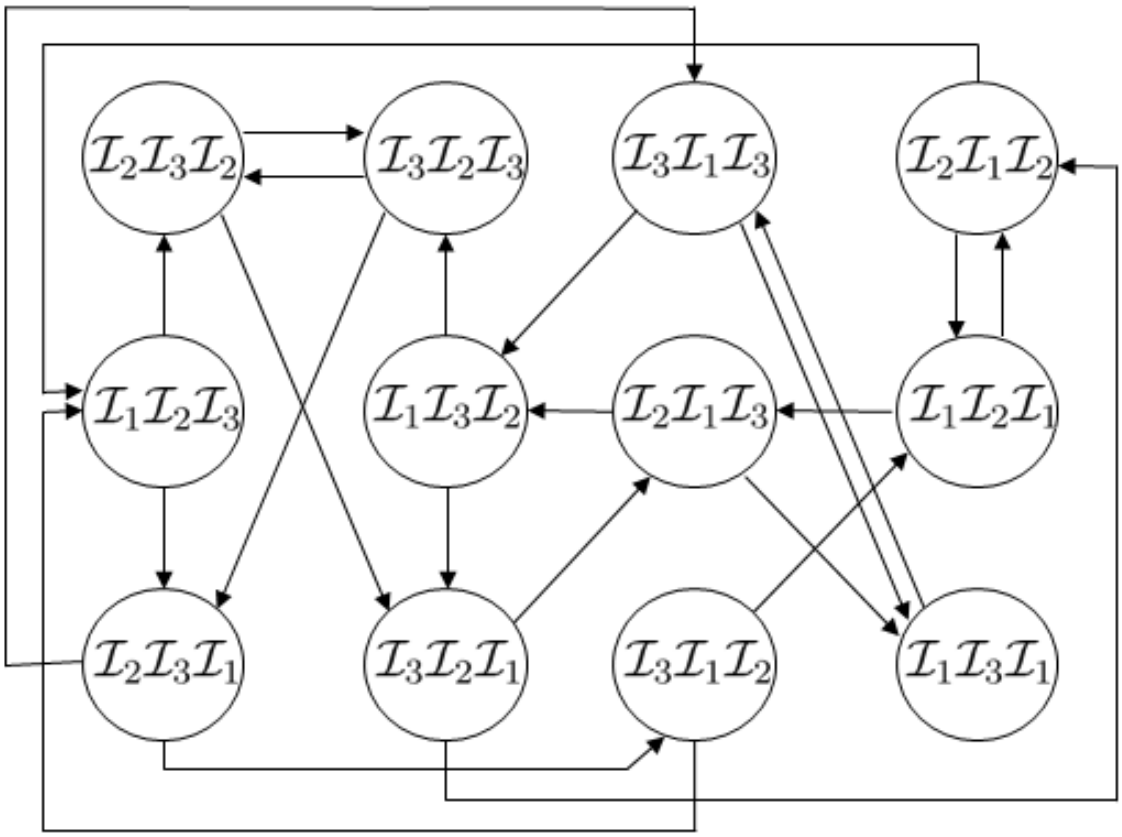


Figure 3.3: The MDP for $N = 3$ and $K = 3$ is illustrated above. Each state refers to the most recent sequence of K visual interest regions observed by the human. Each transition arrow is accompanied by a robot action of either $\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_1}$, $\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_2}$, or $\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_3}$. The appropriate robot action associated with each of the 24 transition arrows must be trained via reinforcement learning.

The robot's action policy is trained online using Q-learning [102]. This technique is one of the most popular approaches to reinforcement learning due to its provable convergence to the optimal action-value function $Q : S \times A \rightarrow \mathbb{R}$.

Each time H switches tasks, R is provided with a reward of the form $r_k = h_k + f_k$. The human input signal $h_k \in \{-1, 0, 1\}$ nominally takes a value of 0; however, H can set

this to either 1 or -1 once per transition to reinforce whether or not R's action was helpful. Regarding the second term, consider the transition time reward $f_k = \eta \,\mathrm{sat}\left(\frac{t_k^o - t_k^m}{t_k^m \eta}\right)$ where t_k^m and t_k^o refer respectively to the time spent maneuvering to $\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i}$ and then observing $\mathcal{I}_i$ before the next transition. The reward f_k "punishes" the robot for taking actions that result in greater time spent maneuvering than observing, e.g., if in some contexts the human tends to glance only briefly at $\mathcal{I}_i$ then the robot ought not to attempt to change tasks. $\eta > 0$ is a tunable parameter for balancing the influence of transition time rewards relative to human inputs.

In training episodes, the robot actions are chosen using an ε-Greedy policy. That is, R chooses action: $a_k = \mathrm{argmax}_{\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i} \in A} Q\left(s_k, \mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i}\right)$, with probability $1 - \varepsilon$, and otherwise chooses a random action. This allows for the human trainer to observe the efficacy of their own training online without too much compromise on the exploration of the state-action space. A purely exploratory policy may have punished the trainers for bothering to put forth any effort. Additional tunable parameters include the learning rate $\alpha \in (0, 1)$ and discount factor $\gamma \in (0, 1)$. Increasing the latter term emphasizes the importance of future rewards, while setting $\gamma = 0$ trains the system to only care about the next reward. The proposed method for Q-learning of sequential relevant camera views is presented in Algorithm 3.2.

Algorithm 3.2 Q-learning of Sequential Relevant Camera Views

Inputs: $h_k, t_k^o, t_k^m, s_{k+1}$
Initialize: $Q\left(s, \mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i}\right) = 0, \forall s \in S$ and $\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i} \in A$
Observe H until GMM model parameters are stable.
Deploy R into task environment.
while 1 **do**
  $\zeta \leftarrow rand(0, 1)$
  if $\zeta < \varepsilon$ **then**
    $a_k \leftarrow rand(A)$
  else
    $a_k \leftarrow \mathrm{argmax}_{\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i} \in A} Q\left(s_k, \mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i}\right)$
  end if
  R executes a_k and observes h_k, t_k^o, t_k^m, and s_{k+1}.
  $r_k \leftarrow h_k + \eta \,\mathrm{sat}\left(\frac{t_k^o - t_k^m}{t_k^m \eta}\right)$
  $Q\left(s_k, a_k\right) \leftarrow Q\left(s_k, a_k\right) +$
  $\alpha\left(r_k + \gamma \max_{\mathcal{S}_{\mathcal{R}}^{\mathcal{I}_i} \in A} Q\left(s_{k+1}, a_{k+1}\right) - Q\left(s_k, a_k\right)\right)$
  $s_k \leftarrow s_{k+1}$
end while

3.2.3 Hardware Description

In all trials, subjects wore a Microsoft Hololens AR unit whose position and orientation was tracked via a VICON motion capture system. The AR unit was outfitted with Pupil Labs binocular eye trackers. The eye trackers' gaze data, when combined with the VICON data, provided an accurate estimate of the position and orientation of $\mathcal{S}_{\mathcal{H}}$ at any given time. The subjects also wore an Intel RealSense depth camera that provided measurements of the point cloud $\mathcal{P}$. The augmented AR setup is illustrated in Fig. 3.4. Raw data from the RealSense and the eye trackers were processed on an Intel NUC Mini PC worn, along with its power supply, on the users' backs. The full wearable system is illustrated in Fig. 3.5.

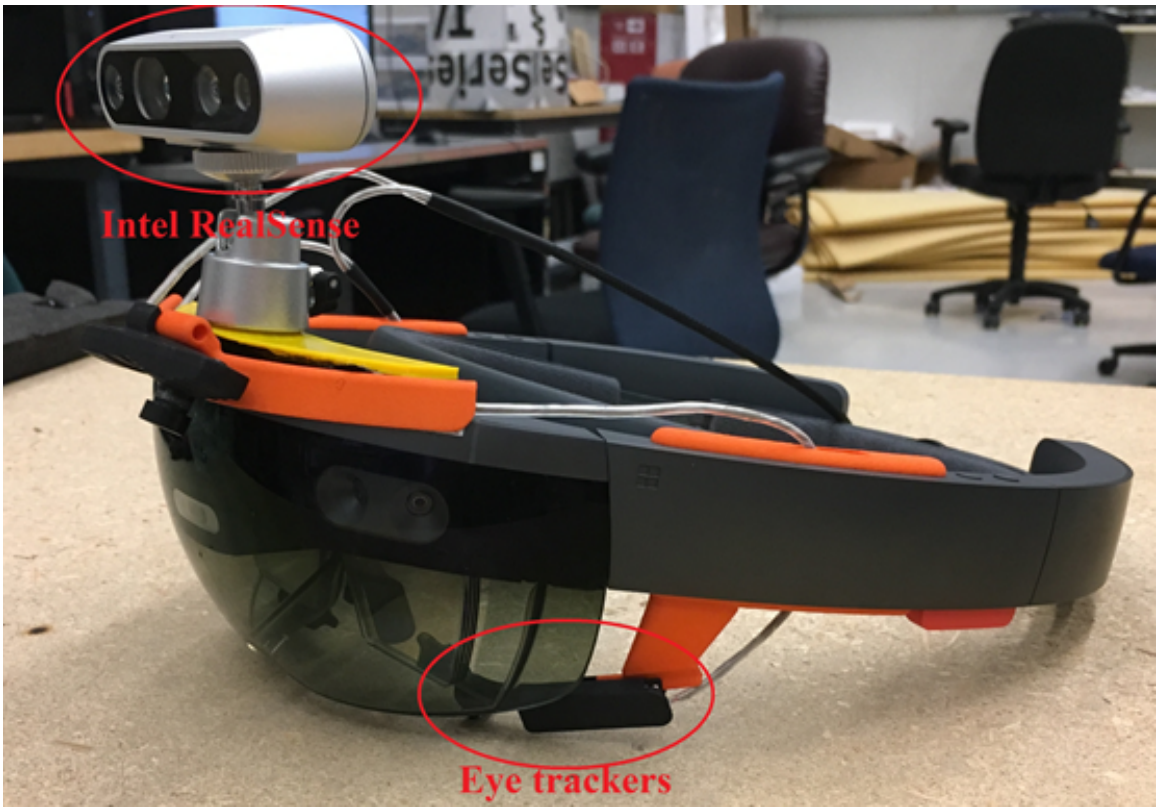


Figure 3.4: Subjects wear a Microsoft Hololens over their eyes to view the UAV's camera feed. Their gaze is tracked via Pupil Labs eye trackers and the point cloud $\mathcal{P}$ is gathered via an Intel RealSense depth camera.

Figure 3.5: Subjects wear a harness with an Intel NUC and power bank attached to their backs. The NUC processes data from the Intel RealSense depth camera as well as the Pupil Labs eye trackers. Both sensors are fastened to a Microsoft Hololens worn on the users head.

A Parrot AR.Drone 2.0 UAV served as the aerial robotic videographer. The UAV was guided between sequential solutions to (3.4) via a low-level waypoint-based PID controller. Note that in environment B, defined in the following subsection, an additional constraint was included in the configuration set $\mathcal{C}_{\mathcal{R}}$ of (3.4). Specifically, $\mathcal{C}_{\mathcal{R}}$ was modified such that $y_{\mathcal{R}} \in [0, 1)$. This encodes that acceptable waypoints for a UAV in environment B are only defined within a cuboid of space in the foreground of Fig 3.8, i.e., a volume of space containing no obstacles which is forward of all of the task locations. This additional constraint was included in the more cluttered task environment, i.e., task environment B, for the purpose of guaranteeing collision avoidance and safety in the presence of human subjects.

The author had originally intended to adopt more robust guidance protocols that generate curved trajectories around the task locations (unknown *a priori*). However, sufficient time did not exist during the design of this study to experimentally validate the safety of more aggressive UAV trajectories.

The UAV carried a 720p HD camera which operated at 30 frames per second. The camera's video feed was broadcast over a TCP stream to the Hololens where it was displayed in the lower right-hand corner of the display's FOV. The hardware communication architecture is presented in Fig. 3.6.

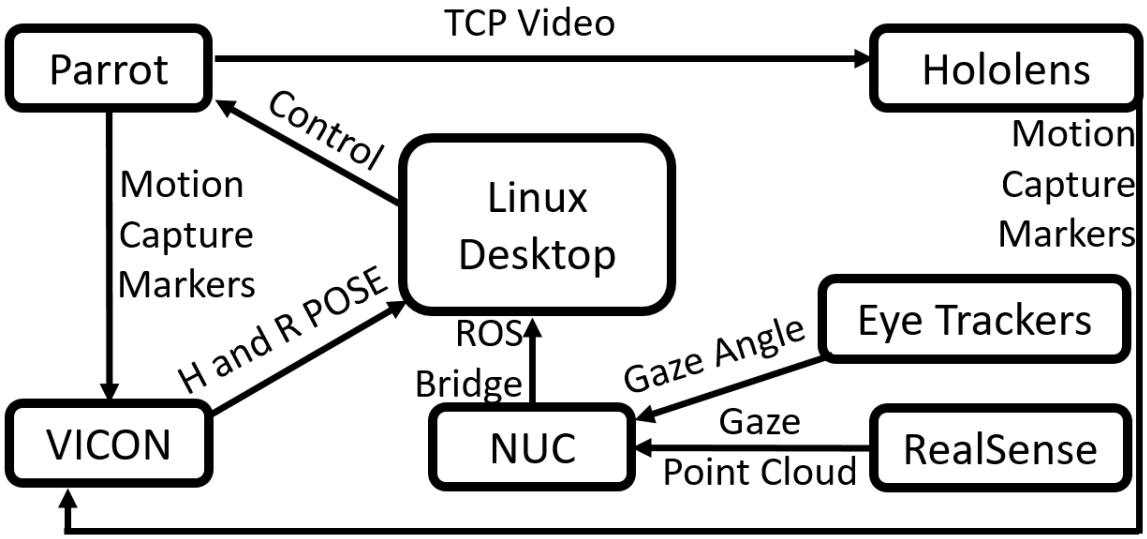


Figure 3.6: The network communication architecture is illustrated above.

3.2.4 Procedure

Under IRB approval HUM00145810, experimental trials were conducted on 24 subjects (ages 19-35). Subjects participated in four sessions, each of which were two hours in length. These four sessions are referred to as sessions *A*, *B*, *C*, and *D* in the sequel. During each session, subjects completed 8 trials—the first four of which were in task environment *A* and the latter four in task environment *B*. These environments will be described in full detail in the following two paragraphs. In each of these trials, the primary objective was to assemble a K'NEX toy construction kit as quickly and accurately as possible. In total, four K'NEX puzzles were used with the sequence of those four randomized in trials 1-4 and 5-8 respectively.

Environment *A* is depicted in Fig. 3.1. It consists of $N = 3$ visual interest regions: a set of instructions for the assembly of their K'NEX toy construction kit, a work bench upon

which to assemble the toy, and a second table containing a button programmed to begin flashing at random times generated by a Poisson counting process. Let us refer to these three regions as $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$ respectively in the context of environment *A*. The button was placed off to the side and behind the subjects' backs so as to be outside of their normal FOV when facing the assembly bench. The subjects were instructed to complete their assembly task as quickly and accurately as possible while also continually monitoring the button. Any time that the subjects noticed the button flashing they were to stop what they were doing and immediately shut off the flashing button. The button task was intended to simulate the random arrivals of time-sensitive tasks which might distract a human attempting to complete some higher level objective.

Environment *B*, depicted in Fig. 3.8, consists of $N = 4$ visual interest regions. It is similar to environment *A* but includes an additional button placed behind the assembly instructions. A subject standing at the workbench can glance to their left in order to view the button denoted $\mathcal{I}_1$; however, this button is entirely obstructed from view when they stand in the vicinity of the button denoted $\mathcal{I}_4$. In addition, both buttons are equipped with an exterior set of red and green lights. The buttons are programmed such that at random times one of the two (i.e., either $\mathcal{I}_1$ or $\mathcal{I}_4$), with equal probability, will begin flashing while the other initiates a cycle between red and green signals. Subjects were instructed to complete their assembly task as quickly and accurately as possible while continually monitoring both buttons. Any time that the subjects noticed either button flashing, they were to stop what they were doing and immediately shut off the flashing button. The buttons were programmed such that the flashing could only be disabled by an input received during the green signal. Subjects were informed that there was no penalty for pressing a button at the wrong time—it would simply remain flashing until they pressed it at the correct time. An illustration of this event is provided in Fig. 3.7.

Environment *B* was contrived to evaluate whether the fully autonomous mode, i.e., the robot automatically selecting actions in accordance with the MDP, is more useful in environments that require information flow between visually obstructed tasks than the supervised autonomy mode, i.e, the mode for which the robot switches tasks only in response to direct commands from the human. For this supervised autonomy mode, subjects were provided with a wireless clicker device. Each time subjects pressed the clicker, R was reassigned to observe the next $\mathcal{I}_i$ in a sequence with incrementally increasing values of i. When assigned to observe $\mathcal{I}_N$, a clicker input would result in R returning to observe $\mathcal{I}_1$ thus reinitializing the sequence. In the fully autonomous mode, there was no clicker and R would instead select elements of $\mathcal{I}$ on its own in accordance with the MDP policy under study in that trial.

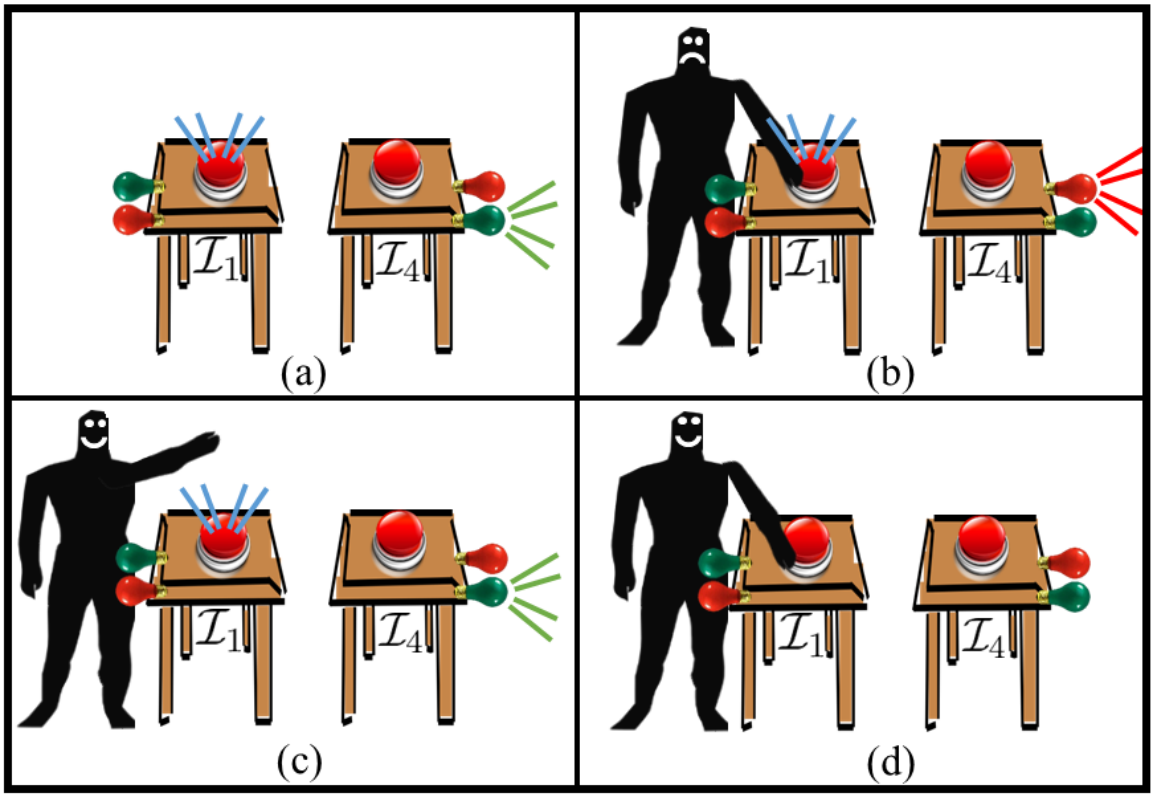


Figure 3.7: H is alerted to a flashing button at $\mathcal{I}_1$ in (a). The subject tries unsuccessfully to shut off $\mathcal{I}_1$ in (b). The light remains flashing though there is no penalty. The subject notices the green signal from $\mathcal{I}_4$ in (c) which indicates that it is now possible to shut off the button in (d). Visual obstructions, such as $\mathcal{I}_2$, are not depicted so as to reduce clutter.

In all sessions, the sequence of trials administered in either task environment were the same: warm-up, control, supervised autonomy, and full autonomy. The warm-up trial was viewed as a practice session in which subjects could refamiliarize themselves with a given task environment. For both the warm-up trial, as well as the control trial, subjects wore their AR device while multitasking—though it, as well as the UAV, were turned off. In addition to serving as a baseline for comparison, the control trial was also used as a training session in which R learned the model for $\bar{\psi}_{\mathcal{H}}$ which would be used in the following two trials.

The subjects were divided into three main test groups. Groups 1-3 consisted of subjects 1-9, 10-18, and 19-24 respectively. For group 1, the value function $Q(s, a)$ was initialized to zero during subject 1's session *A* and then allowed to evolve over the course of all fully autonomous trials within this test group. That is, each subject continued to train R through an explore/exploit policy that built upon the knowledge of all of prior subjects' training. For group 2, a purely exploitive policy was used in all fully autonomous trials. The value function used by group 2 was trained offline using only the training data

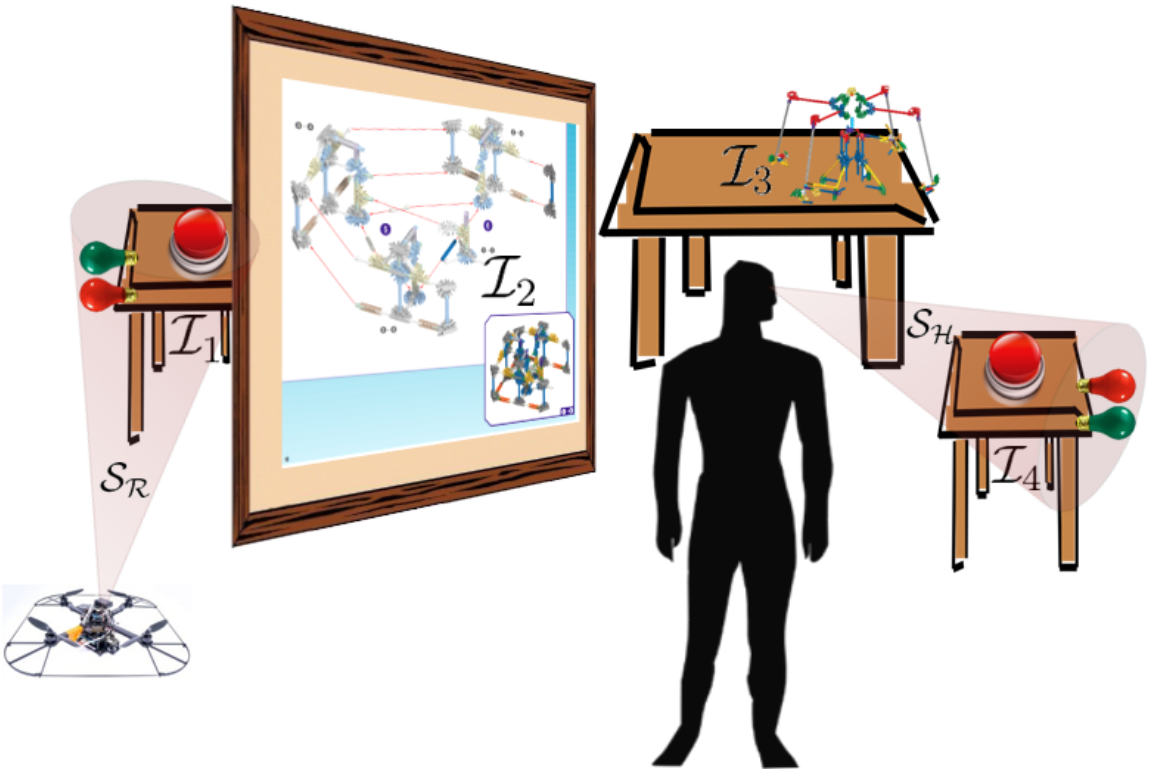


Figure 3.8: The $N = 4$ task environment is similar to the $N = 3$ task environment. It contains an additional button which is obstructed by the assembly instructions. Either button can flash at random times; however, the flashing of one button coincides with the other entering a red/green cycle. A button can only be shut off by H when a green signal is displayed across the room.

provided by a subset of subjects in group 1 whose average button reflex time across all sessions, i.e., situational awareness, was improved over the control trials through use of the fully autonomous system. The reasoning for this decision was that those subjects whose reflex time was improved were providing "good" training data. Those subjects in group 1 who were not helped by the system would have their training data omitted from the value function evaluated by group 2.

Group 3 was commissioned midway through the study as a test case for future work. This group evaluated a modified version of the MDP which had an additional action available in all states. The new action was referred to as *no action* and would simply instruct R to continue observing the same element of $\mathcal{I}$ that had been commanded in the previous MDP state. In addition, the transition time reward f_k was removed from Algorithm 3.2. Instead, for any state in which R attempts to transition to another element of $\mathcal{I}$ but is unable to before receiving a new command, i.e., $t_k^o = 0$, the attempted action is overwritten with *no action* and the value function is updated with a very small reward that is many orders of

Table 3.1: Numerical Parameters Used for Groups 1-3

Parameter	Group 1	Group 2	Group 3 Sessions $\{A, B, C, D\}$
α	0.5	0	$\{0.1, 0.1, 0.1, 0\}$
γ	0.5	0.5	0.1
ε	0.15	0	$\{0.75, 0.5, 0.25, 0\}$
η	0.01	0.01	0.01
$R_{\mathcal{R}}$	$2\,m$	$2\,m$	$2\,m$
$\alpha_{\mathcal{R}}$	$60°$	$60°$	$60°$
$\alpha_{\mathcal{H}}$	$60°$	$60°$	$60°$
$\sigma_{\mathcal{H}}$	$6°$	$6°$	$6°$
d_s	$1\,m$	$1\,m$	$1\,m$

magnitude smaller than $\max h_k$. Thus for a given s_k, *no action* will likely become the optimal action if no other training data is ever received in that state. This is often the case for states that H transitions through quickly, e.g., glancing briefly at the instruction board. This is similar to the motivation behind the transition time reward described in section 3.2.2.2; however, it is based upon the assumption that none of the assistive views are inherently optimal during a brief glance by the human and that additional action by the robot in such states is more distracting than helpful.

Numerical parameters used in this experiment are presented in Table 3.1. For group 1, a learning rate of $\alpha = 0.5$ was used so as to emphasize the importance of new training data. This allowed for $Q(s, a)$ to adapt more quickly to the preferences of each subject while still capitalizing on R's prior experience. For group 2, $\alpha = 0$ and $\varepsilon = 0$ were used as these sessions were purely exploitive with subjects not asked to train R. Group 2 was commissioned to evaluate the performance of $Q(s, a)$ having been trained by subjects in group 1 that were helped by the system. For group 3, $Q(s, a)$ was reinitialized to zero for each subject. In this group, subjects would train their own personalized value function with a smaller learning rate of $\alpha = 0.1$. ε was reduced by increments of 0.25 from the starting value of 0.75 through sessions A, B, C, and D. In this way, the subject initially interacted with a robot that took many random actions so as to quickly explore the state space. As the human became more comfortable with the setup over the four sessions, the robot gradually took more predictable actions. In session D, the robot purely exploited its trained policy with the human no longer needing to provide any reinforcement. In addition, the discount factor was also reduced in group 3 as it was determined to have been too far-sighted in earlier trials.

3.2.5 Hypotheses

Let us consider 17 hypotheses on the interactions between humans, AR, and UAVs in the proposed task environments. The first sixteen of these hypotheses are organized into 4 higher level groups which independently assess the effect of both interface types and both task environments on task assembly time, button reaction time (as a metric of situational awareness), average head angular velocity (as a metric of physical effort), and NASA TLX score (as a metric of mental effort). The final hypothesis assesses subject preference for interface type in either environment.

- *Hypothesis 1.1* The supervised autonomous system will reduce average assembly times of the construction tasks when compared to the control trials within environment *A*.
- *Hypothesis 1.2* The fully autonomous system will reduce average assembly times of the construction tasks when compared to the control trials within environment *A*.
- *Hypothesis 1.3* The supervised autonomous system will reduce average assembly times of the construction tasks when compared to the control trials within environment *B*.
- *Hypothesis 1.4* The fully autonomous system will reduce average assembly times of the construction tasks when compared to the control trials within environment *B*.
- *Hypothesis 2.1* The supervised autonomous system will improve average reaction times to the buzzers when compared to the control trials within environment *A*.
- *Hypothesis 2.2* The fully autonomous system will improve average reaction times to the buzzers when compared to the control trials within environment *A*.
- *Hypothesis 2.3* The supervised autonomous system will improve average reaction times to the buzzers when compared to the control trials within environment *B*.
- *Hypothesis 2.4* The fully autonomous system will improve average reaction times to the buzzers when compared to the control trials within environment *B*.
- *Hypothesis 3.1* The supervised autonomous system will reduce the average absolute value of subject head angular velocity when compared to the control trials within environment *A*.

- *Hypothesis 3.2* The fully autonomous system will reduce the average absolute value of subject head angular velocity when compared to the control trials within environment *A*.
- *Hypothesis 3.3* The supervised autonomous system will reduce the average absolute value of subject head angular velocity when compared to the control trials within environment *B*.
- *Hypothesis 3.4* The fully autonomous system will reduce the average absolute value of subject head angular velocity when compared to the control trials within environment *B*.
- *Hypothesis 4.1* The supervised autonomous system will reduce the average NASA TLX score of subjects when compared to the control trials within environment *A*.
- *Hypothesis 4.2* The fully autonomous system will reduce the average NASA TLX score of subjects when compared to the control trials within environment *A*.
- *Hypothesis 4.3* The supervised autonomous system will reduce the average NASA TLX score of subjects when compared to the control trials within environment *B*.
- *Hypothesis 4.4* The fully autonomous system will reduce the average NASA TLX score of subjects when compared to the control trials within environment *B*.
- *Hypothesis 5* More subjects will prefer the supervised autonomous system than the fully autonomous system in environment *A* which contains fewer tasks and no visual obstructions. In addition, more subjects will prefer the fully autonomous system than the supervised autonomous system in environment *B* which contains more tasks and includes visual obstructions between tasks.

3.2.6 Results

The results are presented below as a set of line graphs which include trends for each interface under evaluation: control, supervised autonomy, and full autonomy. The data points of each trend refer to averages within that test group or, in the case of the fourth column, the average taken over all 24 subjects. To account for the learning curve, the discussion of results and their statistical significance will generally refer to the overall group in session D for any given performance metric under consideration. Such results will inform the acceptance or rejection of *Hypotheses 1.1-4.4*. Note that these results are specific to the contrived task environments *A* and *B* under consideration and do not necessarily generalize to other

Table 3.2: Average System Usability Scale Ratings for Supervised and Autonomous Interfaces

Task Env.	Interface	Group 1	Group 2	Group 3	Overall
A	Supervised	81.4	81.4	73.8	79.5
A	Autonomous	71.4	62.5	75	69.0
B	Supervised	75.3	73.1	70.8	73.3
B	Autonomous	72.5	70.3	68.8	70.7

environments. They do, however, provide some insight as to how abstract metrics, such as situational awareness and physical/mental effort, might improve within other unstructured multitasking environments.

One-way analysis of variance (ANOVA) tests were performed on each of the 32 subplots in Figs. 3.9, 3.10, 3.11, and 3.12 in order to determine whether the differences between interface types in each test group and task environment were statistically significant. The results of these tests are presented in Table 3.4. Results that were significant, i.e., $p < 0.05$ as is standard, indicate that one may reject the null hypothesis that all three interfaces result in the same mean value for the considered performance metric within that environment and test group. Results for which the null hypothesis was rejected were then further analyzed using Tukey's honest significance multiple comparison test in order to determine which interface pairs showed significant performance differences. The results of these pairwise tests are presented in Table 3.5. Significant results are highlighted in bold in both tables. Please see [103]) for a complete discussion of ANOVA and Tukey's honest significance test.

Within training sessions, subjects generally trained R to observe the button tasks. For environment *A* (see Fig. 3.1), this reduced to providing positive reinforcement signals to R whenever it observed $\mathcal{I}_3$ and punishment signals elsewise. For environment *B* (see Fig. 3.8), subjects tended to train R such that it would observe $\mathcal{I}_1$ while they were working at the table $\mathcal{I}_3$. In this way, subjects could look for either a flashing light or a red/green signal appearing at $\mathcal{I}_1$ as the indication that they ought to service the button tasks. Subjects tended to train R to observe $\mathcal{I}_4$ when they stood in the vicinity of $\mathcal{I}_1$ and vice versa. Thus, they could observe the red/green signaling from across the room via their AR device while preparing to service the flashing button.

At the end of the supervised and autonomous trials of session D, subjects were asked to complete SUS surveys of the two assistive interfaces. The results of this survey are presented in Table 3.2. The overall average ratings were all in excess of 68 which generally indicates above average system usability [104].

3.2.6.1 A Discussion of *Hypotheses 1.1-1.4*

The average task assembly times per session are presented in Fig. 3.9. As expected, assembly times generally reduce with each subsequent session across all interfaces. This is associated with the learning curve of the task environment. For both environment *A* and *B*, the overall results indicate that the control trials had the shortest task assembly times. For task environment A, the control trials were completed on average 10-13 seconds faster than the supervised and autonomous interfaces, i.e., a 5.8-7.5% increase in assembly times. However, the one-way analysis of variance presented in Table 3.4 indicates that assembly time differences between interfaces for environment *A* are not statistically significant.

For environment *B*, one does observe statistical significance, $p = 0.0277$, in the proposition that assembly time is worse with supervised autonomy over the control trials. As a qualitative observation, some subjects in environment *B* spent quite a bit of time experimenting with the drone's response to the clicker. These subjects seemed distracted as they stood by the flashing buttons at $\mathcal{I}_1$ and $\mathcal{I}_4$. Rather than shutting them off as quickly as they could, they would instead patiently cycle the robot through each of its learned observation regions until they had found the view they desired. In such a way, the system seemed to slow them down from accomplishing their main objective as they had seemingly forgotten it in lieu of curiosity for the system. Although, it would have been interesting to provide additional reminders to the subjects to remember their primary objective of task completion time across all trials regardless of interface. Instead, subjects were only coached at the top of each session to complete their task as quickly and accurately as possible.

In summary, the author rejects the claim of *Hypothesis 1.3* and finds the claims of *Hypotheses 1.1, 1.2,* and *1.4* to be inconclusive.

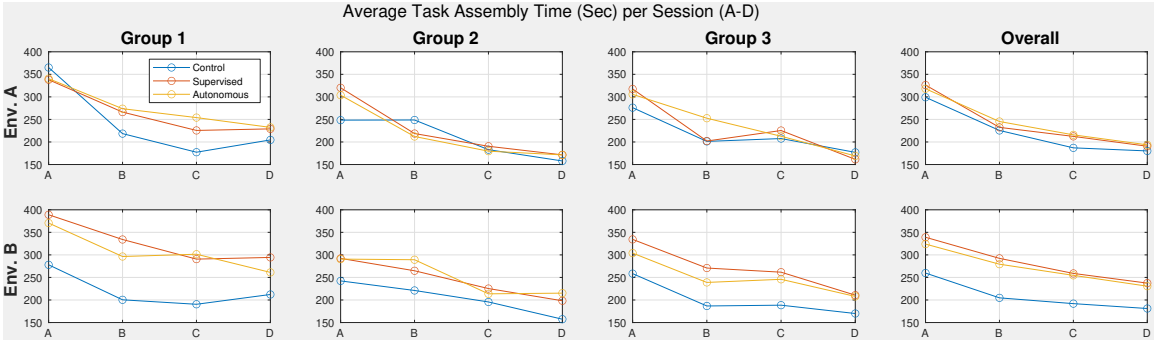


Figure 3.9: Average assembly task completion times over the course of sessions A-D are presented above. The assembly task generally took longer to complete while subjects interacted with the assistive video stream; however, this was not the case for group 3 in environment *A*.

3.2.6.2 A Discussion of *Hypotheses 2.1-2.4*

Average reaction times to the buzzer tasks are presented in Fig. 3.10. In task environment *A*, the supervised autonomy reduced the overall average reaction time down from 8.1 seconds in the control trials to 6.6 seconds, i.e., an 18.5% improvement. In addition, the autonomous system also reduced average reflex time to 8.0 seconds, i.e., a 1.2% improvement. However, it should be noted that only the former improvement is statistically significant as indicated in Table. 3.5.

For task environment *B*, overall reflex times were significantly better (by approximately 2.5 seconds, i.e., 30%) in the control trials than with either of the other interfaces. However, within environment *B* it is problematic to use reflex time as a pure measure of situational awareness due to the long cycling period of the red/green signal. The total period of the signal was always set to 5 seconds with 3.5 seconds allocated to red and 1.5 seconds allocated to green. In such a way, a subject being notified of a flashing light a few seconds early (via AR) did not necessarily translate into them shutting off the buzzer sooner. Instead, subjects often had to wait at either $\mathcal{I}_1$ or $\mathcal{I}_2$ for up to 3.5 seconds for the arrival of a green signal. In addition, subjects in the supervised and autonomous trials tended to wait for the R to finish flying towards its assigned visual interest region before interact with their buzzer. These subjects would often miss a cycling period while R was in transition. This was in stark contrast to subjects in the control trials who would simply run to the flashing button and press it repeatedly until the flashing ceased. In retrospect, the study could have been improved by recording the number of incorrect button inputs provided by subjects. Rather, they had been told that there was no penalty for pressing the button at the wrong time. Qualitatively, it was observed that subjects in the supervised and autonomous trials only needed to press their buttons one time as they patiently relied upon their AR device rather than aggressively *spamming* the button.

For task environment *B*, the author offers that the design of the experiment was problematic in this task environment and proposes that subjects pressing the buttons at incorrect times ought to be discouraged and measured in future experiments. Qualitatively, subjects displayed greater situational awareness using their AR setup in environment *B* as they typically only required one strike of the button in order to shut it off. This contrasts with the *spamming* observed in control trials.

In summary, the author accepts the claim of *Hypothesis 2.1*, finds the claim of *Hypothesis 2.2* to be inconclusive, and rejects the claims of *Hypotheses 2.3-2.4*.

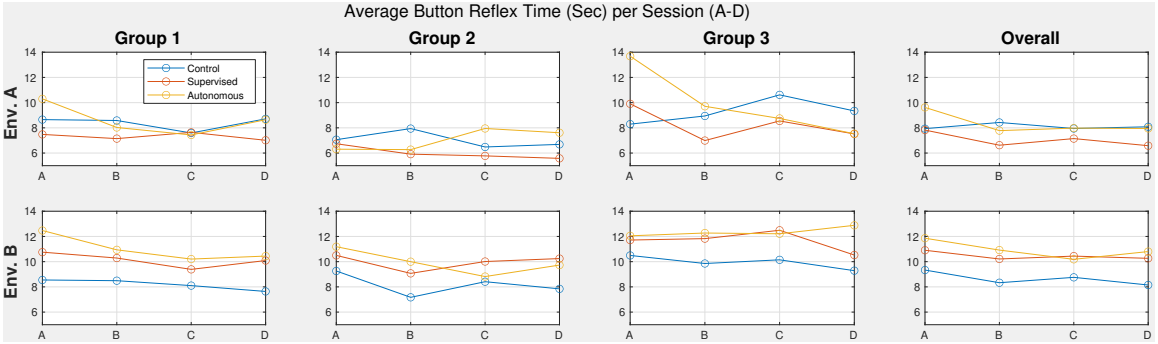


Figure 3.10: Subjects' average reflex times to the buzzers, which flash at random times, are presented above for sessions A-D. The most promising results occur in environment *A*. Furthermore, the autonomous system is most helpful in this regard with the modifications that were introduced for group 3.

3.2.6.3 A Discussion of Hypotheses 3.1-3.4

The average absolute values of subjects' head angular velocities are presented in Fig. 3.11. In environment *A*, the overall average for session D control trials was 1.05 rad/sec. The autonomous and supervised systems reduced this value to 0.90 and 0.83 rad/sec respectively, i.e., a 14-21% reduction in physical effort. In environment *B*, the control trial average of 1.01 rad/sec was reduced in the supervised and autonomous systems to 0.87 and 0.84 rad/sec respectively, i.e., a 14-17% reduction in physical effort. Noting Table 3.5, the improvements associated with the supervised system are statistically significant in environment *A* while the improvements associated with the autonomous system are statistically significant in environment *B*.

In summary, the author accepts the claims of *Hypotheses 3.1* and *3.4*. The claims of *Hypotheses 3.2-3.3* are inconclusive.

3.2.6.4 A Discussion of Hypotheses 4.1-4.4

Subject average TLX ratings are presented in Fig. 3.12. These results serve as a subjective measure of mental effort. Subjects generally rated that their perceived task load was reduced by using either assistive system. For environment *A*, session D overall TLX scores were reduced from 42 in control trials to 38.5 and 36.6 using the autonomous and supervised systems respectively, i.e., an 8.3% and 12.9% improvement. For environment *B*, these values were 41.8, 37.5, and 37.2 respectively, i.e., a 10.3% and 11.0% improvement. The ANOVA tests cannot confirm that these results are statistically significant. Thus, the claims of *Hypotheses 4.1-4.4* are inconclusive. For this study, only weighted TLX scores were considered and this may have contributed to increased variance within this data set.

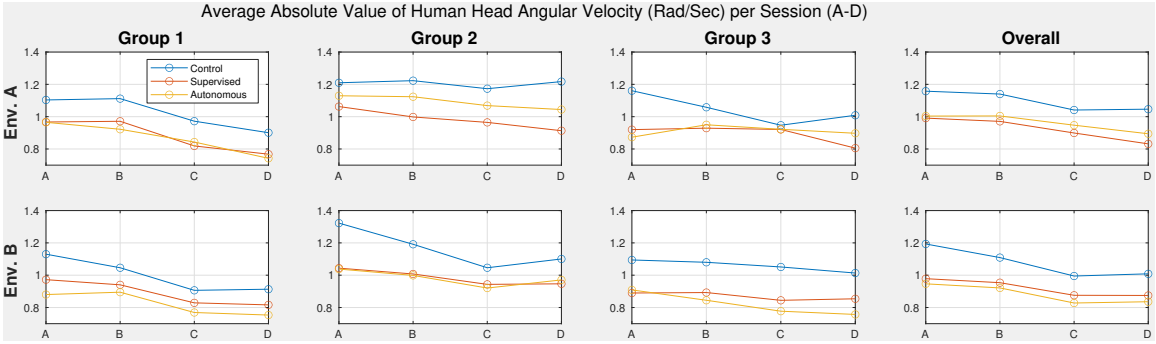


Figure 3.11: The average absolute value of human head angular velocity over the course of sessions A-D are presented above. This serves as one measure of physical effort. Both interfaces reduced head motions in the overall session D averages. The supervised autonomy was most helpful in this regard for environment *A* while the full autonomy was most helpful in environment *B*.

For future work, it is recommended that raw TLX scores be gathered as well. A larger sample size of subjects as well as the use of raw TLX scores may increase the statistical significance of the TLX scores.

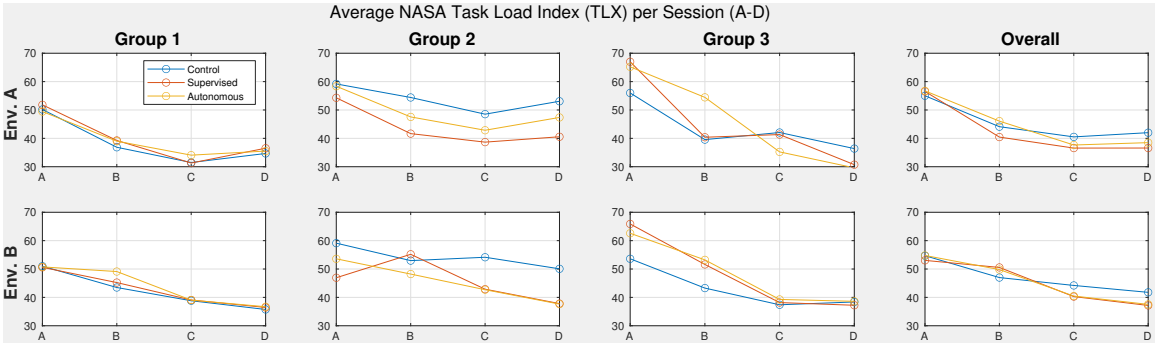


Figure 3.12: Subject average task load indices are presented above for sessions A-D. This survey data serves as a measure of perceived effort which is analogous to mental effort. Both interfaces reduced perceived task load in the overall average for session D.

3.2.6.5 A Discussion of Hypothesis 5

At the completion of session D, subjects were asked directly whether they preferred the autonomous or supervised systems in both task environment *A* as well as *B*. The results of this post study survey are presented in Table 3.3. Overall, 87.5% of subjects preferred the supervised clicker system in task environment *A* and 70.8% of subjects preferred the fully autonomous system in task environment *B*. To analyze the significance of these results, the binomial test (see [105]) was performed in each environment with the null hypothesis

Table 3.3: Percentage of Subjects Preferring Autonomous Over Supervised Control

Task Env.	Group 1	Group 2	Group 3	Overall
A	0%	0%	50%	12.5%
B	66.7%	77.8%	66.7%	70.8%

Table 3.4: One-way ANOVA Statistics for Control, Supervised, and Autonomous System Comparison

Measurement Type	Task Env.	Group 1	Group 2	Group 3	Overall
Avg. Task Assembly Time	A	$F(2,24) = 0.37$, $p = 0.6952$	$F(2,24) = 0.26$, $p = 0.7748$	$F(2,15) = 0.2$, $p = 0.8221$	$F(2,69) = 0.31$, $p = 0.7341$
	B	$F(2,24) = 2.15$, $p = 0.1382$	$F(2,24) = 2.08$, $p = 0.1463$	$F(2,15) = 1.17$, $p = 0.3367$	$\boldsymbol{F(2,69) = 4.14}$, $\boldsymbol{p = 0.0201}$
Avg. Abs. Val. of Head Ang. Vel.	A	$F(2,24) = 2.59$, $p = 0.0959$	$F(2,24) = 1.38$, $p = 0.2708$	$F(2,15) = 0.79$, $p = 0.4735$	$\boldsymbol{F(2,69) = 3.15}$, $\boldsymbol{p = 0.0489}$
	B	$F(2,24) = 2.28$, $p = 0.1245$	$F(2,24) = 0.79$, $p = 0.4663$	$F(2,15) = 3.56$, $p = 0.0543$	$\boldsymbol{F(2,69) = 3.98}$, $\boldsymbol{p = 0.0231}$
Button Reflex Time	A	$\boldsymbol{F(2,248) = 3.33}$, $\boldsymbol{p = 0.0373}$	$\boldsymbol{F(2,224) = 5}$, $\boldsymbol{p = 0.0075}$	$F(2,140) = 1.46$, $p = 0.2353$	$\boldsymbol{F(2,618) = 5.83}$, $\boldsymbol{p = 0.0031}$
	B	$\boldsymbol{F(2,271) = 5.76}$, $\boldsymbol{p = 0.0036}$	$\boldsymbol{F(2,235) = 3.57}$, $\boldsymbol{p = 0.0298}$	$\boldsymbol{F(2,143) = 4.12}$, $\boldsymbol{p = 0.0182}$	$\boldsymbol{F(2,655) = 11.56}$, $\boldsymbol{p = 1.1633E-5}$
TLX	A	$F(2,24) = 0.02$, $p = 0.9827$	$F(2,24) = 0.76$, $p = 0.4782$	$F(2,15) = 0.21$, $p = 0.8115$	$F(2,69) = 0.41$, $p = 0.6674$
	B	$F(2,24) = 0.01$, $p = 0.9949$	$F(2,24) = 1.08$, $p = 0.3555$	$F(2,15) = 0.01$, $p = 0.991$	$F(2,69) = 0.4$, $p = 0.6718$

Table 3.5: p-values for Multiple Comparisons in Tukey's Honest Significance Test

Measurement Type	Task Env.	Group	p-value		
			Control/ Supervised	Control/ Autonomous	Supervised/ Autonomous
Avg. Task Assembly Time	B	Overall	$\mathbf{0.0277}$	0.0601	0.9457
Avg. Abs. Val. of Head Ang. Vel.	A	Overall	$\mathbf{0.0448}$	0.2013	0.7578
	B	Overall	0.1005	$\mathbf{0.0241}$	0.8184
Button Reflex Time	A	Group 1	0.0762	0.9961	0.0685
	A	Group 2	0.1963	0.3199	$\mathbf{0.0045}$
	A	Overall	$\mathbf{0.0082}$	0.9917	$\mathbf{0.0106}$
	B	Group 1	$\mathbf{0.014}$	$\mathbf{0.0059}$	0.9147
	B	Group 2	$\mathbf{0.0297}$	0.1047	0.8389
	B	Group 3	0.5684	$\mathbf{0.0134}$	0.1285
	B	Overall	$\mathbf{0.0004}$	$\mathbf{2.14E-5}$	0.7172

that 50% of subjects would prefer one interface. The resulting p values were 2.77×10^{-4} and 0.064 for environments *A* and *B* respectively. Thus, one can accept the first claim of *Hypothesis 5* and find the second claim to be inconclusive. Note that the second claim is close to the borderline of acceptance, e.g., changing only a single survey response would have brought this second claim to within the 95% confidence interval.

It should be noted that no subject ever responded that they preferred full autonomy in environment *A* until group 3 evaluated the modified learning parameters and *no action* enabled MDP. Qualitatively, the addition of the *no action* command seemed to have substantially reduced the number of unnecessary transitions in environment *A* and resulted in trained behavior that was very similar to the supervised system, i.e., R would transition immediately to observing $\mathcal{I}_3$ and would remain there throughout the trial.

3.2.6.6 Subject Remarks and Observations

A few interesting qualitative observations were gathered during the study through post session discussions with the subjects. The experimenters observed that whenever subject 6 was called to service $\mathcal{I}_1$ (a flashing button) in environment *B*, they tended to look away from $\mathcal{I}_1$ and instead towards $\mathcal{I}_3$ (the work bench) while waiting for the green signal from

$\mathcal{I}_4$. The subject stated that it dawned upon them that they had inadvertently trained R to observe $\mathcal{I}_4$ in this configuration. Rather than attempting to retrain it, the subject felt it was easy enough to glance at $\mathcal{I}_3$ in order to observe $\mathcal{I}_4$ through their AR unit. This was an example of the human directly manipulating the structure of the MDP in a manner that had never been considered by the experimenters. The MDP was designed to model learning by the robot. Thus, it was interesting to observe a human learning new actions in response to the robot's learned actions.

Within group 2, it was interesting to note that the autonomous system initially helped to improve reflex time in environment *A* during early sessions, e.g., sessions A and B, but became less helpful in later sessions. This may be related to some qualitative observations during subject 12's participation. It was noted that the subject would initially reference the instructions quite often while working at the assembly table. This translated into frequent transitions between the MDP states denoted $\{\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_1\}$ and $\{\mathcal{I}_2, \mathcal{I}_1, \mathcal{I}_2\}$. The value function used by group 2 had been trained to monitor the button $\mathcal{I}_3$ in these configurations and had not received much training in other states. As subjects in group 2 gained experience in their assembly tasks, they tended to reference the instructions less often and thus the robot received far fewer commands to observe $\mathcal{I}_3$.

One final anecdotal observation was that subject 13 seemed to have memorized the timing sequence of the red/green signal in environment *B*. The subject was observed counting verbally while walking at a leisurely pace towards the button tasks and rarely engaged in the *spamming* behavior noted earlier. Thus, their reaction times appeared to be relatively consistent across all interfaces.

3.2.6.7 Discussion on Alternative and Related Approaches

An argument can be made that system performance may have improved, particularly in environment *B*, had $\mathcal{C}_{\mathcal{R}}$ been further restricted such that the drone maintained a fixed position (of possibly high altitude) from which unobstructed views of all tasks were accessible via rotational maneuvers. While this would certainly reduce the drone's transition time between views, this approach would not scale well to a larger outdoor environment. The motivation for this study is primarily that of EVAs or field engagements in which a human is required to gather detailed information from a number of discrete locations that may be hundreds of feet apart. A single, eye-in-the-sky, view may not provide the appropriate image fidelity in such applications. However, an eye-in-the-sky approach may be more appropriate for small scale task environments—such as those which were contrived for this study due to the size of the lab space.

This work primarily focuses upon the online learning of camera views that are relevant

to multiple spatially separated tasks. However, a number of prior works have studied the related problem of crew-viewing in EVA. The first of these works relied upon the Secondary Camera and Maneuvering Platform (SCAMP)—an underwater free flyer developed by the University of Maryland for NASA. The McDonnell Douglas Aerospace Underwater Test Facility demonstrated in September of 1993 that SCAMP could effectively follow and photograph an astronaut while they maneuvered around elements of a simulated space station truss [106]. This was accomplished via teleoperation; however, vision-based tracking was later addressed in [107]. In the former work, SCAMP provided external views of the subject engaged in a dexterous servicing task that allowed for ground controllers outside of the facility to maintain greater situational awareness of the task. This study emphasized the importance of crew-viewing by ground controllers so as to provide better auditory instructions in task assistance and troubleshooting.

Crew-viewing during EVA evolved into self-viewing during a 2008 study at the University of Maryland's neutral buoyancy facility: the Space Systems Laboratory. In this study, the test subject had access to a video display inside of the suit helmet and was tasked with the replacement of an electronics unit bolted to an underwater structure. The subject found the use of assistive camera views to be particularly helpful while removing the electrical wiring from the (visually obstructed) underside of the structure as well as during the positioning of their feet into foot restraints. The primary complaint, with respect to the use of assistive camera views, was that of the 2 second time delay [108].

Robot self-viewing, specifically over-the-shoulder viewing, is considered in user studies presented in [109]. The author tasked human subjects with remote piloting a simulated SCAMP through a number of hoops distributed throughout the Space Systems Lab. Subjects were given access to the SCAMP's onboard camera as well as an over-the-shoulder view in which they could see the SCAMP as well. Subjects overwhelmingly preferred this view as it allowed for greater situational awareness. Specifically, subjects tended to collide more frequently with the hoops using the SCAMP's onboard camera as they were unable to see their own proximity to the edge of the hoops as they passed through them. This study, while focused upon teleoperation of robots, illustrates how over-the-shoulder views can assist a spacewalking astronaut while maneuvering around obstacles while may be impeding portions of their body outside of their FOV.

Self-viewing and crew-viewing, as studied by McDonnell Douglas and the University of Maryland's Space Systems Lab, are important related approaches to the utilization of free flying robotic cameras during EVA. However, they address a fundamentally different problem than that of the persistent monitoring of multiple spatially separated tasks as studied in this chapter.

3.2.7 Conclusion

This section presented an automatic system for the learning of task-relevant camera views within unstructured multitasking environments that are occupied by a human collaborating with an aerial robot. The aerial robot carries a camera and continuously streams a live video feed to the human's AR display. In this way, the human's FOV is extended by the aerial assistant.

This system learns the locations of visual interest regions within the environment by tracking human visual gaze, integrating point cloud data that intersects this visual gaze, and then fitting the resulting map to a GMM online via expectation maximization. Having learned a set of potential camera views, the robot is commanded to photograph these regions using either a supervised autonomy interface, i.e., clicker control by the human, or a fully autonomous control mode.

In the fully autonomous control mode, the human's current visual interest region is estimated online with motion capture. The human's sequence of K most recent visual interests is encoded as a state within an MDP. This sequence of human visual interests provides temporal context as to what region the robot ought to observe in any given MDP state. Each time the human switches their visual focus, a robot action is triggered. The robot is trained on which action it ought to execute in any given MDP state via Q-learning.

In the user study, clear indications were found that the system reduces physical effort during multitasking as measured through average head angular velocity. The proposed supervised interface improved the reaction times of subjects to time-sensitive tasks, which arrive at random times, in the simpler task environment containing 3 regions of interest. The improvements to reaction time could not be replicated in the more complex environment containing 4 tasks; however, this may be in part due to the red/green cycling period of this environment as described earlier. In addition, subjects generally took longer to complete their assembly task using these interfaces. Although, there can be some argument that the reductions in physical effort may be worth the increased assembly times.

Subjects generally preferred to use the supervised control interface in task environment *A* and the fully autonomous interface in task environment *B* with the latter result having marginal statistical significance. Group 3, which tested a modified version of the MDP augmented with the *no action* command, was the only test group containing any subjects that preferred the autonomous system in environment *A*. In addition, group 3 was also the only group for which the autonomous interface actually improved task assembly times. This occurred in environment *A* (see Fig. 3.9). Group 3 was a relatively small sample size; however, the changes introduced to the algorithm in this group are quite promising. A potential future study is advised to adopt the parameter values used by group 3, as well

as the *no action* command, and evaluate whether statistically significant results can be gathered from a larger sample size, i.e., 30 subjects.

The task environment under consideration was highly specific and the hypotheses that were studied here may not necessarily generalize to other environments. The results do, however, provide insight as to how this system might improve abstract performance metrics, such as situational awareness and physical/mental effort, in other unstructured multitasking environments.

CHAPTER 4

Conclusions and Future Work

4.1 Conclusions

In summary, this dissertation serves primarily to prescribe a set of control and estimation strategies to extend the situational awareness of multi-agent networks. These networks can consist of either multiple robots, as in chapter 2, or mixed teams of humans and robots, as in chapter 3.

Chapter 2 presents a set of dynamic coverage control and estimation strategies. Agents are required to patrol a domain while sweeping their sensing footprints across points of interest. In section 2.2, all points must be covered, at one time or another, up to a satisfactory level. In section 2.3, the coverage level at select points decays in accordance with the estimated trajectories and arrival statistics of stochastic intruders. In that work, agents are attempting to recover a decaying coverage level by exploring in the vicinity of the predicted impact points of these intruders. This approach differentiates itself from related works, such as [11], in that the statistics of the intruders are encoded into the coverage level itself and not solely in the agent control laws. Thus, the effects of these intruders are not forgotten the moment they exit the domain. Throughout the majority of chapter 2, agents operate with limited power resources and must periodically return to charging stations. These power constraints are considered in the scheduled deployment of agents and in their assignments to dynamically reallocated energy-aware domain partitions. The work described above is demonstrated both through simulations and experimental studies. Chapter 2 concludes with a brief case study on gesture-based supervised coverage. The contribution here is a coverage protocol whereby a human can direct a robot to engage in local coverage at specific locations prescribed by the human's hand gestures. This essentially replaces the global coverage logic of section 2.2 with direct human commands.

Chapter 3 extends the concept of situational awareness to networks consisting of humans and aerial vehicles operating in close proximity. A set of algorithms is presented that

allow for an aerial robot to infer and sequence the regions of visual interest to a human collaborator within a multitasking environment. The aerial vehicle computes the optimal configuration for viewing each of these visual interest regions and streams these camera views to the human's AR headset so as to extend the FOV of the human. This essentially allows for the human to cover a greater portion of the multitasking environment at any given time thus increasing their situational awareness. These algorithms are experimentally demonstrated through an IRB approved (HUM00145810) user study.

4.2 Future Work

With respect to dynamic coverage, one of the most important avenues for future work lies in the design of control strategies that consider the dynamics of the vehicles rather than simply the kinematics. Within the lab environment, it was often necessary to over approximate the radii of the vehicles so as to increase the safety distance margins between vehicles in close proximity. This is because physical quadrotors, unlike the kinematic models typically seen in coverage literature, cannot null out their velocities instantaneously. Coverage formulations that account for vehicle dynamics will allow for collision avoidance guarantees to hold under more aggressive flight patterns. In addition, it is recommended that future works consider more detailed models of battery dynamics. Battery lifespans tend to reduce over the course of many charging cycles. The accurate estimation of a battery's current lifespan, given the evolution of its discharge characteristics, is an active research area that should be exploited so as to maximize an agent's flight time.

With respect to HRI with aerial/space co-robots, there are a number of potential research avenues. The first of which would be to create a more space-like environment for the user study of section 3.2. The primary motivation for this study was to develop a camera control system for free flying space co-robots (e.g., AERCam) that would improve the situational awareness of astronauts conducting EVA. The subjects of the study had relatively unrestricted motion and vision which is not representative of the physically taxing environment in which an astronaut works. In addition, the subjects also had the luxury of Earth's gravity which obscures the difficulty that astronauts have in establishing consistent reference frames through verbal communication.

To address these experimental limitations, future studies ought to introduce additional physical constrictions on the subjects (e.g., a helmet to limit their FOV and garments to restrict motion) so as to more accurately simulate the experience of an astronaut in space. In addition, one should consider testing this algorithm within a neutral buoyancy laboratory and comparing its performance against a teleoperated camera. Such incremental steps may

enable the ultimate goal of deployment and testing aboard ISS.

In addition, the entire concept of using reinforcement learning to train the selection of camera views from a learned set could be superseded by the implementation of an effective brain computer interface (BCI). A number of researchers have already demonstrated that humans can learn to control UAVs with electroencephalogram (EEG) devices using a set of high level commands [110]. In most of these works, the commands are directional, i.e., *up, down, left right, forward, backward*. Thus, the command to select a camera view from a relatively small set, as in this dissertation, is actually a simpler implementation of BCI controlled drones than those in the contemporary literature. Brain commands are typically issued via motor imagery, i.e., visualizing the movement of a body part. However, in one case a team of researchers demonstrated that the imagination of auditory phrases could be used to control a UAV [111]. It would be interesting to study whether thoughts such as *instructions, buzzer, work bench* could be used to control the UAV of section 3.2. Such a study may be directly relevant to recent grant proposals issued by DARPA to develop brain controlled drones for use in the battlefield [112].

4.3 Additional Related Work

In addition to the main work presented in this dissertation, the author also conducted an additional case study on HRI. This work considered a novel method for generating the trajectory of a robot assisting a human in servicing a set of tasks embedded in a convex 2-D domain. The method makes use of Bayesian inference to predict human intent in task selection. Rather than following the optimal trajectory towards a single task, the robot computes a set of potentially optimal tasks each weighted by the humans posterior probability and superimposes them into a cost function that is designed to minimize the weighted Euclidean distance relative to set. The effect is a flexible path human-robot collaborative network that is shown in simulation to complete all tasks in a given domain in less time than existing methods for a certain class of highly impulsive humans, i.e., humans that tend to randomly switch tasks at times generated by a Poisson counting process.

This work was presented at the 2018 IEEE/RSJ Conference on Intelligent Robots and Systems:

- W. Bentz and D. Panagou "Bayesian-inferred Flexible Path Generation in Human-Robot Collaborative Networks", 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, October 2018.

APPENDIX A

Complete Coverage of 3D Environments

A.1 Automaton

The automaton is described by the following entities [113]:

- a set of discrete states: $Z_i = \{\zeta_{i0}, \zeta_{i1}, \zeta_{i2}, \zeta_{i3}, \zeta_{i4}\}$ which represent local coverage, global coverage, subdomain transfer, and waypoint scan modes (1) and (2) respectively,

- a set of continuous states: $\tilde{q}_i = \{x_i, y_i, z_i, \Phi_i, \Theta_i, \Psi_i\}$ representing the position and orientation of agent i in Cartesian coordinates and 3-2-1 Euler angles respectively,

- a vector field: $f(\zeta_{i0}, \tilde{q}_i) = \mathcal{R}\left[\acute{u}_i^{loc}\ \acute{v}_i^{loc}\ \acute{w}_i^{loc}\ 0\ 0\ \acute{s}_i^{loc}\right]^T$,
  $f(\zeta_{i1}, \tilde{q}_i) = \mathcal{R}\left[u_i^{glo}\ v_i^{glo}\ w_i^{glo}\ 0\ 0\ s_i^{glo}\right]^T$, $f(\zeta_{i2}, \tilde{q}_i) = \mathcal{R}\left[u_i^{sub}\ v_i^{sub}\ w_i^{sub}\ 0\ 0\ s_i^{sub}\right]^T$,
  $f(\zeta_{i3}, \tilde{q}_i) = \mathcal{R}\left[u_i^{wps}\ v_i^{wps}\ w_i^{wps}\ 0\ 0\ 0\right]^T$, $f(\zeta_{i4}, \tilde{q}_i) = \mathcal{R}\left[0\ 0\ 0\ 0\ 0\ s_i^{wps}\right]^T$,
  where $\mathcal{R} = \begin{bmatrix} \mathcal{R}_1 & 0 \\ 0 & \mathcal{R}_2 \end{bmatrix}$ and $\dot{\tilde{q}}_i = f(\zeta_{ik}, \tilde{q}_i)$ is the control input for agent i in state ζ_{ik}. Note that ζ_{i0}, ζ_{i1} and ζ_{i2} command both translation and yaw while ζ_{i3} only commands translation and ζ_{i4} only commands yaw. This is because ζ_{i3} commands that agent i transition between the sub-waypoints of waypoint scan mode while ζ_{i4} commands the 360° yaw sweep at each sub-waypoint.

- a set of initial states: $\{\zeta_{i0}\} \times \{\tilde{q}_i \in \mathbb{R}^6 \mid \|p_i\| \leq \bar{R}_{\mathcal{D}} \wedge \Phi_i \in (-\pi, +\pi) \wedge \Theta_i \in \left(\frac{-\pi}{2}, \frac{+\pi}{2}\right) \wedge \Psi_i \in (-\pi, +\pi)\}$ encoding that each agent begin in local coverage mode inside of the domain at an orientation within the traditional Euler angle range,

- a domain:

$Dom\left(\zeta_{i0}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid (i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| \leq \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| \leq \bar{R}_{cr}\right)\}$,

$Dom\left(\zeta_{i1}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid (i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| \leq \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| \leq \bar{R}_{cr}\right)\}$,

$Dom\left(\zeta_{i2}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid (i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| > \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| > \bar{R}_{cr}\right)\}$,

$Dom\left(\zeta_{i3}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid (i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| \leq \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| \leq \bar{R}_{cr}\right)\}$,

$Dom\left(\zeta_{i4}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid (i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| \leq \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| \leq \bar{R}_{cr}\right)\}$,

where four of the five discrete states are defined over the set of continuous states for which agent i is inside of its prescribed subdomain $\mathcal{D}^j$. ζ_{i3} (subdomain transfer mode) is defined for agent i outside of its prescribed subdomain,

- A set of edges: $E = \{\left(\zeta_{i0}, \zeta_{i1}\right), \left(\zeta_{i1}, \zeta_{i0}\right), \left(\zeta_{i1}, \zeta_{i2}\right), \left(\zeta_{i0}, \zeta_{i2}\right), \left(\zeta_{i2}, \zeta_{i0}\right) \left(\zeta_{i1}, \zeta_{i3}\right), \left(\zeta_{i3}, \zeta_{i1}\right), \left(\zeta_{i3}, \zeta_{i2}\right), \left(\zeta_{i3}, \zeta_{i0}\right), \left(\zeta_{i3}, \zeta_{i4}\right), \left(\zeta_{i4}, \zeta_{i3}\right), \left(\zeta_{i4}, \zeta_{i2}\right)\}$, each of which is a transition that may be triggered by:

- A guard condition: $G\left(\zeta_{i0}, \zeta_{i1}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \left(|\hat{e}_i(t)| < \varepsilon_1\right) \wedge \big((i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| \leq \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| \leq \bar{R}_{cr}\right) \big)\}$ which transitions agent i from local to global coverage when the agent's contribution to the rate of change of the global coverage error drops below some threshold (the coverage rate is insufficient) so long as the agent operates within its prescribed subdomain,
$G\left(\zeta_{i1}, \zeta_{i0}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \left(\|p_i - p_{\star,i}\| < \varepsilon_2\right) \wedge \left(\|\Psi_i - \Psi_{\star,i}\| < \varepsilon_3\right) \wedge \left(|\hat{e}_i(t)| \geq \varepsilon_1\right) \wedge \big((i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| \leq \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| \leq \bar{R}_{cr}\right) \big)\}$ which transitions agent i from global to local coverage when the desired position and orientation has been achieved to within some threshold, the coverage rate is sufficient, and the agent operates within its prescribed subdomain,
$G\left(\zeta_{i1}, \zeta_{i2}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \left(i \in \{2, ..., N-1\} \wedge \|p_i\| > \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| > \bar{R}_{cr}\right)\}$ which transitions agent i from global coverage to subdomain transfer mode when the agent is not within its prescribed subdomain,
$G\left(\zeta_{i0}, \zeta_{i2}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \left(i \in \{2, ..., N-1\} \wedge \|p_i\| > \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| > \bar{R}_{cr}\right)\}$ which transitions agent i from local coverage to subdomain transfer mode when the agent is not within its prescribed subdomain,
$G\left(\zeta_{i2}, \zeta_{i0}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid (i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| \leq \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| \leq \bar{R}_{cr}\right)\}$ which transitions agent i from subdomain transfer mode to local coverage mode when the agent is within its prescribed subdomain,
$G\left(\zeta_{i1}, \zeta_{i3}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \left(\|p_i - p_{\star,i}\| < \varepsilon_2\right) \wedge \left(\|\Psi_i - \Psi_{\star,i}\| < \varepsilon_3\right) \wedge \left(|\hat{e}_i(t)| < \varepsilon_1\right) \wedge \big((i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| \leq \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| \leq \bar{R}_{cr}\right) \big)\}$ which transitions agent i from global coverage to waypoint scan mode (1) when the desired position and orientation has been achieved to within some threshold, the coverage rate is insufficient, and the agent operates within its prescribed subdomain,
$G\left(\zeta_{i3}, \zeta_{i1}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid wp > 13 \wedge \left(|\hat{e}_i(t)| < \varepsilon_1\right) \wedge \big((i = 1 \wedge p_i \in \mathcal{D}) \vee \left(i \in \{2, ..., N-1\} \wedge \|p_i\| \leq \bar{R}_{i-1}\right) \vee \left(i = i_{cr} \wedge \|p_i\| \leq \bar{R}_{cr}\right) \big)\}$ which transitions agent

i from waypoint scan mode (1) to global coverage when all 13 subwaypoints have been scanned, the coverage rate is insufficient, and the agent operates within its prescribed subdomain,
$G\left(\zeta_{i3},\zeta_{i2}\right)=\{\tilde{q}_i\in\mathbb{R}^6\mid\left(i\in\{2,...,N-1\}\wedge\|p_i\|>\bar{R}_{i-1}\right)\vee\left(i=i_{cr}\wedge\|p_i\|>\bar{R}_{cr}\right)\}$ which transitions agent i from waypoint scan mode (1) to subdomain transfer mode when the agent is not within its prescribed subdomain,
$G\left(\zeta_{i3},\zeta_{i0}\right)=\{\tilde{q}_i\in\mathbb{R}^6\mid wp>13\wedge\left(|\hat{e}_i(t)|\geq\varepsilon_1\right)\wedge\big(\left(i=1\wedge p_i\in\mathcal{D}\right)\vee$
$\left(i\in\{2,...,N-1\}\wedge\|p_i\|\leq\bar{R}_{i-1}\right)\vee\left(i=i_{cr}\wedge\|p_i\|\leq\bar{R}_{cr}\right)\big)\}$ which transitions agent i from waypoint scan mode (1) to local coverage when all 13 sub-waypoints have been scanned, the coverage rate is sufficient, and the agent operates within its prescribed subdomain,
$G\left(\zeta_{i3},\zeta_{i4}\right)=\{\tilde{q}_i\in\mathbb{R}^6\mid\|p_i-p_{\star,i}\|<\varepsilon_2\wedge\big(\left(i=1\wedge p_i\in\mathcal{D}\right)\vee$
$\left(i\in\{2,...,N-1\}\wedge\|p_i\|\leq\bar{R}_{i-1}\right)\vee\left(i=i_{cr}\wedge\|p_i\|\leq\bar{R}_{cr}\right)\big)\}$ which transitions agent i from waypoint scan mode (1) to waypoint scan mode (2) when the agent has converged upon a sub-waypoint position to within some threshold and the agent operates within its prescribed subdomain,
$G\left(\zeta_{i4},\zeta_{i3}\right)=\{\tilde{q}_i\in\mathbb{R}^6\mid\Psi_i=\Psi_i(t_{G(\zeta_{i3},\zeta_{i4})})-d\Psi\wedge\big(\left(i=1\wedge p_i\in\mathcal{D}\right)\vee$
$\left(i\in\{2,...,N-1\}\wedge\|p_i\|\leq\bar{R}_{i-1}\right)\vee\left(i=i_{cr}\wedge\|p_i\|\leq\bar{R}_{cr}\right)\big)\}$ which transitions agent i from waypoint scan mode (2) to waypoint scan mode (1) when the agent has completed a 360° yaw sweep at one of the 13 sub-waypoints and operates within its prescribed subdomain
$G\left(\zeta_{i4},\zeta_{i2}\right)=\{\tilde{q}_i\in\mathbb{R}^6\mid\left(i\in\{2,...,N-1\}\wedge\|p_i\|>\bar{R}_{i-1}\right)\vee\left(i=i_{cr}\wedge\|p_i\|>\bar{R}_{cr}\right)\}$ which transitions agent i from waypoint scan mode (2) to subdomain transfer mode when the agent is not within its prescribed subdomain,

- A reset map: $R=\{\emptyset\}$ which is empty and included for the sake of completeness.

$\hat{e}_i(t)$, defined in (2.12), evolves as a function of the time history of the continuous states and essentially represents how well the local coverage protocol is proceeding. ε_1 is a guard value for $\hat{e}_i(t)$, while ε_2 and ε_3 are guard values for position and yaw proximity with respect to the desired values, $p_{\star,i}$ and $\Psi_{\star,i}$, respectively. wp is an index variable for a set of sub-waypoints which defined in section 2.2.5. $t_{G(\zeta_{i3},\zeta_{i4})}$ is the most recent time at which $G(\zeta_{i3},\zeta_{i4})$ was satisfied. $t_{G(\zeta_{i3},\zeta_{i4})}$ is of relevance because satisfaction of $G(\zeta_{i3},\zeta_{i4})$ requires an agent to yaw around 360° and resume its previous orientation at $t_{G(\zeta_{i3},\zeta_{i4})}$. The variables $\bar{R}_{i-1}$ and $\bar{R}_{cr}$ are the radii of subdomain partitions defined in section 2.2.2.3 along with i_{cr}. The control laws presented in the vector field definition are derived throughout section 2.2.

Note that the case in which $i=1$ is included as an exit guard condition from ζ_{i2} to

ζ_{i0} despite no defined transition for $i = 1$ to ζ_{i2}. This is due to the fact that agent i_{cr} may converge upon $\mathcal{O}$ in the state ζ_{i2} and then become instantaneously reindexed as $i = 1$ upon battery exchange.

A.2 Differentiation under the integral sign

The Reynold's transport theorem (2.9), a 3-D generalization of the Leibniz integral rule, is applied to (2.8) in order to compute its time derivative (2.10). Therefore, it is a necessary condition that both $h\left(C^{\star} - Q\left(t, \tilde{p}\right)\right)$, i.e. the integrand of (2.8), and $\frac{\partial}{\partial t} h\left(C^{\star} - Q\left(t, \tilde{p}\right)\right)$ are continuous over t and $\tilde{p}$.

Lemma 2. $h\left(C^{\star} - Q\left(t, \tilde{p}\right)\right)$ is continuous in both t and $\tilde{p}$.

Proof. It is a well known result that if two functions are continuous in t and $\tilde{p}$, then their difference, product and the maximum of the two functions are all continuous in t and $\tilde{p}$ [114]. Note that 0 and $C^{\star}$ are both constants, and thus continuous in any argument. An assumption of the continuity of $Q\left(t, \tilde{p}\right)$ implies continuity of the difference $C^{\star} - Q\left(t, \tilde{p}\right)$ which implies continuity of $\max\{0, C^{\star} - Q\}$. Continuity of products implies that $h\left(C^{\star} - Q\left(t, \tilde{p}\right)\right) = \max\{0, C^{\star} - Q\}^3$ is continuous in t and $\tilde{p}$. It remains is to verify that continuity of $Q\left(t, \tilde{p}\right)$ is a valid assumption.

Continuity of $Q\left(t, \tilde{p}\right)$ in t implies that $\forall \epsilon > 0$, $\exists \delta > 0$ such that $\left|Q\left(t \pm \delta, \tilde{p}\right) - Q\left(t, \tilde{p}\right)\right| < \epsilon$. Note the definition of $Q\left(t, \tilde{p}\right)$ in (2.6) and (2.7) and that $S_i\left(\tilde{q}_i(\tau), \tilde{p}\right)$ is upper-bounded by 1 and lower-bounded by 0, regardless of its arguments, $\forall i \in \{1, ..., N\}$. The boundedness of the integrand in (2.6) implies that the existence of an arbitrarily small δ is guaranteed. Continuity in t holds.

$Q\left(t, \tilde{p}\right)$ may be rewritten in terms of its infinite series right-handed Riemann sum:

$$Q\left(t, \tilde{p}\right) = \sum_{i=1}^{N} \lim_{n \to \infty} \sum_{\ell=1}^{n} S_i\big(\tilde{q}_i\left(\ell t n^{-1}\right), \tilde{p}\big)\left(t n^{-1}\right). \tag{A.1}$$

Continuity of $S_i\big(\tilde{q}_i\left(\ell t n^{-1}\right), \tilde{p}\big)$, $\forall \tilde{p} \in \mathcal{D}$ may now be shown. Note from (2.3) and (2.5) that $S_i\big(\tilde{q}_i\left(\ell t n^{-1}\right), \tilde{p}\big)$ is defined over all of $\mathcal{D}$ but takes nonzero values only within $\mathcal{S}_i$, as parametrized by continuous constraint functions $C_{ki}, \forall k \in \{1, 2, 3\}$, and is defined as zero along the boundary $\partial\mathcal{S}_i$. As $S_i\big(\tilde{q}_i\left(\ell t n^{-1}\right), \tilde{p}\big) \to 0$ as $\tilde{p} \to \partial\mathcal{S}_i$ from any direction in $\mathbb{R}^3$, it follows that $S_i\big(\tilde{q}_i\left(\ell t n^{-1}\right), \tilde{p}\big)$ is continuous $\forall \tilde{p} \in \mathcal{D}$. This continuity holds through summation in (A.1) as well. This concludes the proof. □

Lemma 3. $\frac{\partial}{\partial t} h\left(C^{\star} - Q\left(t, \tilde{p}\right)\right)$ is continuous in both t and $\tilde{p}$.

Proof. In Lemma 2 it is shown that $Q(t,\tilde{p})$ and thus $\max\{0, C^{\star}-Q(t,\tilde{p})\}^3$ are continuous in t and $\tilde{p}$. Taking the time derivative it follows that $\frac{\partial}{\partial t}h\left(C^{\star}-Q(t,\tilde{p})\right) = -3\max\{0, C^{\star}-Q(t,\tilde{p})\}^2\left(\frac{\partial Q}{\partial t}\right)$ is continuous in the first multiplicative term. It remains to show that $\frac{\partial Q}{\partial t}$ is continuous in t and $\tilde{p}$. Recall from (2.6) and (2.7) that:

$$Q(t,\tilde{p}) = \sum_{i=1}^{N}\int_0^t S_i\left(\tilde{q}_i(\tau),\tilde{p}\right)d\tau. \tag{A.2}$$

It is shown in Lemma 2 that $S_i\left(\tilde{q}_i(\tau),\tilde{p}\right)$ is continuous in $\tilde{p}$. Continuity in t is guaranteed as the state of $\mathcal{S}_i$ is determined by the position and orientation of agent i for which discontinuity is physically impossible. Thus (A.2) meets all necessary conditions for differentiation via the first fundamental theorem of calculus:

$$\frac{\partial Q(t,\tilde{p})}{\partial t} = \sum_{i=1}^{N}\frac{\partial}{\partial t}\int_0^t S_i\left(\tilde{q}_i(\tau),\tilde{p}\right)d\tau = \sum_{i=1}^{N} S_i\left(\tilde{q}_i(\tau),\tilde{p}\right).$$

which is continuous in t and $\tilde{p}$ as verified above and in Lemma 2. This concludes the proof. □

A.3 A supporting lemma

Lemma 4. Under the assumption that $x \leq y$ and that $x \geq 2$, one may conclude that:

$$\frac{x^3}{x-1} \leq \frac{y^3}{y-1}.$$

Proof. The assumptions directly provide the following:

$$2 \leq x \leq y.$$

Subtract 1:

$$1 \leq x-1 \leq y-1,$$

invert all terms, and then rearrange into the equivalent statement:

$$\frac{1}{y-1} \leq \frac{1}{x-1} \leq 1.$$

$\{\frac{1}{y-1}, \frac{1}{x-1}\}$ and $\{\frac{1}{x-1}, 1\}$ may be referred to as similarly ordered sets. Chebyshev's Sum Inequality Theorem as presented in [115] may be applied to these similarly ordered sets to yield:

$$\frac{1}{(y-1)(x-1)} + \frac{1}{x-1} \geq \frac{1}{2}\left(\frac{1}{y-1} + \frac{1}{x-1}\right)\left(\frac{1}{x-1} + 1\right).$$

Now simplify and rearrange as follows:

$$\frac{1}{(y-1)(x-1)} + \frac{1}{x-1} \geq \frac{1}{2}\left(\frac{1}{y-1} + \frac{1}{x-1}\right)\left(\frac{x}{x-1}\right),$$

$$\frac{1}{(y-1)(x-1)} + \frac{1}{x-1} \geq \frac{1}{2}\left(\frac{x}{(y-1)(x-1)} + \frac{x}{(x-1)^2}\right),$$

$$\frac{1}{(y-1)(x-1)} + \frac{1}{x-1} \geq \frac{\frac{1}{2}x}{(y-1)(x-1)} + \frac{\frac{1}{2}x}{(x-1)^2},$$

$$\frac{1}{x-1} \geq \frac{\frac{1}{2}x - 1}{(y-1)(x-1)} + \frac{\frac{1}{2}x}{(x-1)^2}.$$

Multiply by $2x^2(x-1)$:

$$2x^2 \geq \frac{x^3 - 2x^2}{y-1} + \frac{x^3}{x-1},$$

and rearrange to form:

$$\frac{x^3}{x-1} \leq 2x^2 - \frac{x^3 - 2x^2}{y-1}. \tag{A.3}$$

Thus, the proof will be completed by verifying that the right hand side of (A.3) is less than or equal to $\frac{y^3}{y-1}$ under the assumptions provided by the lemma. Noting the fact that:

$$2x^2 - \frac{x^3 - 2x^2}{y-1} = \frac{2x^2y - x^3}{y-1},$$

this is equivalent to verifying that $2x^2y - x^3 \leq y^3$ for $y \geq x$. This is straight forward:

$$(x-y)^2 \geq 0,$$

$$x^2 + y^2 - 2xy \geq 0,$$

$$x^3 + xy^2 - 2x^2y \geq 0,$$

and thus for $y \geq x$:

$$2x^2y \leq x^3 + xy^2 \leq x^3 + y^3.$$

This verifies the condition $2x^2y - x^3 \leq y^3$. Therefore, (A.3) may be rewritten as:

$$\frac{x^3}{x-1} \leq 2x^2 - \frac{x^3 - 2x^2}{y-1} \leq \frac{y^3}{y-1},$$

$$\frac{x^3}{x-1} \leq \frac{y^3}{y-1}.$$

This concludes the proof. □

APPENDIX B

Persistent Coverage of 2D Manifolds in Stochastic Environments

B.1 Formal Hybrid Formulation

To provide a compact notation in this section, define $\acute{f}_i = \frac{x_i^2}{\left(x_{\mathcal{C},r}+\aleph_i\right)^2} + \frac{y_i^2}{\left(y_{\mathcal{C},r}+\aleph_i\right)^2} + \frac{z_i^2}{\left(z_{\mathcal{C},r}+\aleph_i\right)^2}$. The coverage strategy for agent i is represented by the hybrid automaton in Fig. 2.24, described by the following entities [113]:

- A set of discrete states: $Z_i = \{\zeta_{i0}, \zeta_{i1}, \zeta_{i2}, \zeta_{i3}, \zeta_{i4}\}$,
- A set of continuous states: $\tilde{q}_i = \{x_i, y_i, z_i, \Phi_i, \Theta_i, \Psi_i\}$,
- A vector field:
  $f(\zeta_{i0}, \tilde{q}_i) = \mathcal{R}\left[u_i^{\mathcal{C},loc}\ v_i^{\mathcal{C},loc}\ w_i^{\mathcal{C},loc}\ 0\ r_i^{\mathcal{C},loc}\ s_i^{\mathcal{C},loc}\ \right]^T$,
  $f(\zeta_{i1}, \tilde{q}_i) = \mathcal{R}\left[u_i^{\mathcal{C},rtb}\ v_i^{\mathcal{C},rtb}\ w_i^{\mathcal{C},rtb}\ q_i^{\mathcal{C},rtb}\ r_i^{\mathcal{C},rtb}\ s_i^{\mathcal{C},rtb}\ \right]^T$,
  $f(\zeta_{i2}, \tilde{q}_i) = \mathcal{R}\left[u_i^{\mathcal{C},pim}\ v_i^{\mathcal{C},pim}\ w_i^{\mathcal{C},pim}\ q_i^{\mathcal{C},pim}\ r_i^{\mathcal{C},pim}\ s_i^{\mathcal{C},pim}\ \right]^T$,
  $f(\zeta_{i3}, \tilde{q}_i) = \mathcal{R}\left[u_i^{\mathcal{C},ptm}\ v_i^{\mathcal{C},ptm}\ w_i^{\mathcal{C},ptm}\ q_i^{\mathcal{C},ptm}\ r_i^{\mathcal{C},ptm}\ s_i^{\mathcal{C},ptm}\ \right]^T$,
  $f(\zeta_{i4}, \tilde{q}_i) = \mathcal{R}\left[u_i^{\mathcal{C},stm}\ v_i^{\mathcal{C},stm}\ w_i^{\mathcal{C},stm}\ 0\ 0\ 0\ \right]^T$ where $\mathcal{R} = \begin{bmatrix} \mathcal{R}_1 & 0 \\ 0 & \mathcal{R}_2 \end{bmatrix}$,
- A set of initial states: $\{\zeta_{i3}\} \times \{\tilde{q}_i \in \mathbb{R}^6 \mid p_i = \mathcal{F}$
  $\wedge\ \Phi_i \in [-\pi, +\pi]\ \wedge\ \Theta_i \in \left[\frac{-\pi}{2}, \frac{+\pi}{2}\right]\ \wedge\ \Psi_i \in [-\pi, +\pi]\}$,
- A domain: $Dom\left(\zeta_{i0}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \acute{f}_i \geq 1 \wedge$
  $\left(i_p \in \{2, ..., N\}\ \Longrightarrow\ \bar{z}_{i_p-1} \leq z_i \leq \bar{z}_{i_p-2}\right)\}$,
  $Dom\left(\zeta_{i1}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \acute{f}_i \geq 1\}$,
  $Dom\left(\zeta_{i2}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \acute{f}_i \geq 1\}$,

$Dom\left(\zeta_{i3}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \acute{f}_i \geq 1 \wedge$
$\left(i_p \in \{2, ..., N\} \implies z_i < \bar{z}_{i_p-1} \vee z_i > \bar{z}_{i_p-2}\right)\},$
$Dom\left(\zeta_{i4}\right) = \{\tilde{q}_i \in \mathbb{R}^6 \mid \acute{f}_i \geq 1\},$

- A set of edges: $E = \{\left(\zeta_{i0}, \zeta_{i1}\right), \left(\zeta_{i0}, \zeta_{i2}\right), \left(\zeta_{i0}, \zeta_{i3}\right), \left(\zeta_{i0}, \zeta_{i4}\right), \left(\zeta_{i1}, \zeta_{i3}\right),$
$\left(\zeta_{i2}, \zeta_{i0}\right), \left(\zeta_{i2}, \zeta_{i1}\right), \left(\zeta_{i2}, \zeta_{i3}\right), \left(\zeta_{i2}, \zeta_{i4}\right), \left(\zeta_{i3}, \zeta_{i0}\right),$
$\left(\zeta_{i3}, \zeta_{i2}\right), \left(\zeta_{i3}, \zeta_{i4}\right), \left(\zeta_{i4}, \zeta_{i0}\right), \left(\zeta_{i4}, \zeta_{i2}\right), \left(\zeta_{i4}, \zeta_{i3}\right), \},$

- A set of guard conditions:

$G\left(\zeta_{i0}, \zeta_{i1}\right) = \{i_p = 1 \wedge t_{i\mathcal{F}} \geq T^\star - \frac{\pi g_{\mathcal{C}_0}}{2U_{max}^{agt}}\left(x_{\mathcal{C}_0,r} + z_{\mathcal{C}_0,r}\right)\},$
$G\left(\zeta_{i0}, \zeta_{i2}\right) = \{(\exists k \mid i = i_k) \wedge \left(\|p_i - \mathit{proj}_{\mathcal{C}_{\mu_i}} p_{\star,i}\| > \varepsilon_4\right)\},$
$G\left(\zeta_{i0}, \zeta_{i3}\right) = \{i_p \neq 1 \wedge \left(z_i < \bar{z}_{i_p-1} \vee z_i > \bar{z}_{i_p-2}\right)\},$
$G\left(\zeta_{i0}, \zeta_{i4}\right) = \{\|p_i - p_j\| \leq R \wedge i_{pr} \neq \text{argmax}_{\bar{j}}\left(t_{\bar{j}\mathcal{F}}\right)\},$
$G\left(\zeta_{i1}, \zeta_{i3}\right) = \{\|p_i - \mathcal{F}\| \leq \varepsilon_4 \wedge t_{i\mathcal{F}} = T^\star\},$
$G\left(\zeta_{i2}, \zeta_{i0}\right) = \{\left(t \geq t_{ck} \wedge\right.$
$\left(\left(i_p \in \{2, ..., N\} \wedge \bar{z}_{i_p-1} \leq z_i \leq \bar{z}_{i_p-2}\right) \vee\right.$
$\left.\left.\left(i_p = 1 \wedge t_{i\mathcal{F}} < T^\star - \frac{\pi g_{\mathcal{C}_0}}{2U_{max}^{agt}}\left(x_{\mathcal{C}_0,r} + z_{\mathcal{C}_0,r}\right)\right)\right)\right) \vee$
$\left(t < t_{ck} \wedge \|p_i - \mathit{proj}_{\mathcal{C}_{\mu_i}} p_{\star,i}\| \leq \varepsilon_4 \wedge \mu_i = 0\right)\},$
$G\left(\zeta_{i2}, \zeta_{i1}\right) = \{\|p_i - \mathit{proj}_{\mathcal{C}_{\mu_i}} p_{\star,i}\| \leq \varepsilon_4 \wedge \mu_i = 0 \wedge t \geq t_{ck} \wedge i_p = 1 \wedge t_{i\mathcal{F}} \geq T^\star - \frac{\pi g_{\mathcal{C}_0}}{2U_{max}^{agt}}\left(x_{\mathcal{C}_0,r} + z_{\mathcal{C}_0,r}\right)\}$
$G\left(\zeta_{i2}, \zeta_{i3}\right) = \{\|p_i - \mathit{proj}_{\mathcal{C}_{\mu_i}} p_{\star,i}\| \leq \varepsilon_4 \wedge \mu_i = 0 \wedge t \geq t_{ck} \wedge i_p \in \{2, ..., N\} \wedge \left(z_i < \bar{z}_{i_p-1} \vee z_i > \bar{z}_{i_p-2}\right)\},$
$G\left(\zeta_{i2}, \zeta_{i4}\right) = G\left(\zeta_{i0}, \zeta_{i4}\right) \vee \{\|p_i - \mathit{proj}_{\mathcal{C}_{\mu_i}} p_{\star,i}\| \leq \varepsilon_4 \wedge \left(\|p_i - p_j\| > R, \forall j \vee i_{pr} = \text{argmax}_{\bar{j}}\left(t_{\bar{j}\mathcal{F}}\right)\right) \wedge \mu_i > 0\},$
$G\left(\zeta_{i3}, \zeta_{i0}\right) = \{i_p = 1 \vee \left(i_p \neq 1 \wedge \bar{z}_{i_p-1} \leq z_i \leq \bar{z}_{i_p-2}\right)\},$
$G\left(\zeta_{i3}, \zeta_{i2}\right) = G\left(\zeta_{i0}, \zeta_{i2}\right),$
$G\left(\zeta_{i3}, \zeta_{i4}\right) = G\left(\zeta_{i0}, \zeta_{i4}\right) \vee \{\|p_i - \mathit{proj}_{\mathcal{C}_{\mu_i}} p_{\star,i}\| \leq \varepsilon_4 \wedge \left(\|p_i - p_j\| > R, \forall j \vee i_{pr} = \text{argmax}_{\bar{j}}\left(t_{\bar{j}\mathcal{F}}\right)\right) \wedge \mu_i > 0\},$
$G\left(\zeta_{i4}, \zeta_{i0}\right) = \{f_i = 0 \wedge |\ln\left(\frac{\|n_i\| - \imath_i}{(\gamma + \mu_i)R - \imath_i}\right)| < \varepsilon_5 \wedge \left(i_p = 1 \vee \left(i_p \in \{2, ..., N\} \wedge \bar{z}_{i_p-1} \leq z_i \leq \bar{z}_{i_p-2}\right)\right)\},$
$G\left(\zeta_{i4}, \zeta_{i2}\right) = \{f_i = 1 \wedge |\ln\left(\frac{\|n_i\| - \imath_i}{(\gamma + \mu_i)R - \imath_i}\right)| < \varepsilon_5\},$
$G\left(\zeta_{i4}, \zeta_{i3}\right) = \{f_i = 0 \wedge |\ln\left(\frac{\|n_i\| - \imath_i}{(\gamma + \mu_i)R - \imath_i}\right)| < \varepsilon_5 \wedge \left(i_p \neq 1 \wedge \left(z_i < \bar{z}_{i_p-1} \vee z_i > \bar{z}_{i_p-2}\right)\right)\}.$

- Additional parameters include a clock set: $C = \{t_{i\mathcal{F}}\}$, a flag: $f_i \in \{0, 1\}$, an assignment index $\mu_i = \{0, ..., N-1\}$ and,

- A reset map: $R\left(\zeta_{i0}, \zeta_{i2}, f_i\right) = \{1\}, R\left(\zeta_{i0}, \zeta_{i4}, \mu_i\right) = \{\mu_i + 1\}, R\left(\zeta_{i1}, \zeta_{i3}, t_{i\mathcal{F}}\right) = \{0\},$
$R\left(\zeta_{i2}, \zeta_{i0}, f_i\right) = \{0 \text{ if } t \geq t_{ck}; 1 \text{ otherwise}\}, R\left(\zeta_{i2}, \zeta_{i1}, f_i\right) = \{0\},$
$R\left(\zeta_{i2}, \zeta_{i3}, f_i\right) = \{0\}, R\left(\zeta_{i2}, \zeta_{i4}, \mu_i\right) = \{0 \text{ if } \|p_i - \mathit{proj}_{\mathcal{C}_{\mu_i}} p_{\star,i}\| \leq \varepsilon_4$
$\wedge \left(\|p_i - p_j\| > R, \forall j \vee i_{pr} = \text{argmax}_{\bar{j}}\left(t_{\bar{j}\mathcal{F}}\right)\right) \wedge \mu_i > 0; \mu_i + 1 \text{ otherwise}\},$
$R\left(\zeta_{i3}, \zeta_{i2}, f_i\right) = \{1\}, R\left(\zeta_{i3}, \zeta_{i4}, \mu_i\right) = \{0 \text{ if } \|p_i - \mathit{proj}_{\mathcal{C}_{\mu_i}} p_{\star,i}\| \leq \varepsilon_4$
$\wedge \left(\|p_i - p_j\| > R, \forall j \vee i_{pr} = \text{argmax}_{\bar{j}}\left(t_{\bar{j}\mathcal{F}}\right)\right) \wedge \mu_i > 0; \mu_i + 1 \text{ otherwise}\},$
$R\left(\zeta_{i4}, \zeta_{i0}, \mu_i\right) = \{0\}$, and continuous states do not reset between transitions.

BIBLIOGRAPHY

[1] "Wasp AE Small Unmanned Aircraft System," Airforce Technology, https://www.airforce-technology.com/projects/wasp-ae-small-unmanned-aircraft-system/.

[2] "Australian Defence Force Selects AeroVironment Wasp For SUAS Program," Jun 2017, https://uasweekly.com/2017/06/01/australian-defence-force-selects-aerovironment-wasp-suas-program/.

[3] Shanthakumar, P., Yu, K., Singh, M., Orevillo, J., Bianchi, E., Hebdon, M., and Tokekar, P., "View Planning and Navigation Algorithms for Autonomous Bridge Inspection with UAVs," *International Symposium on Experimental Robotics (ISER)*, 2018.

[4] Abbot, R. I. and Wallace, T. P., "Decision support in space situational awareness," *Lincoln Laboratory Journal*, Vol. 16, No. 2, 2007, pp. 297.

[5] Fredrickson, S. E., Duran, S., Howard, N., and Wagenknecht, J. D., "Application of the mini AERCam free flyer for orbital inspection," *Defense and Security*, International Society for Optics and Photonics, 2004, pp. 26–35.

[6] Scharf, R., "ISS and Orion Inspection Capabilities and Challenges," Tech. rep., NASA Johnson Space Center, 2017, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20170000805.pdf.

[7] LeVasseur, D., "The History of SPHERES," NASA, Apr 2015, https://www.nasa.gov/spheres/history.html.

[8] Cortes, J., Martínez, S., Karatas, T., and Bullo, F., "Coverage control for mobile sensing networks," *IEEE Trans. on Robotics and Automation*, Vol. 20, No. 2, 2004, pp. 243–255.

[9] Hussein, I. I. and Stipanović, D. M., "Effective Coverage Control for Mobile Sensor Networks With Guaranteed Collision Avoidance," *IEEE Trans. on Control Systems Technology*, Vol. 15, No. 4, July 2007, pp. 642–657.

[10] Hokayem, P., Stipanovíc, D., and Spong, M., "On persistent coverage control," *Proc. of the 2007 IEEE Conference on Decision and Control*, New Orleans, LA, USA, Dec. 2007, pp. 6130–6135.

[11] Hübel, N., Hirche, S., Gusrialdi, A., Hatanaka, T., Fujita, M., and Sawodny, O., "Coverage control with information decay in dynamic environments," *Proc. of the 17th IFAC World Congress*, Seoul, South Korea, July 2008, pp. 4180–4185.

[12] Yu, J., Karaman, S., and Rus, D., "Persistent monitoring of events with stochastic arrivals at multiple stations," *IEEE Transactions on Robotics*, Vol. 31, No. 3, 2015, pp. 521–535.

[13] Szafir, D., Mutlu, B., and Fong, T., "Communication of Intent in Assistive Free Flyers," *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, ACM, New York, NY, USA, 2014, pp. 358–365.

[14] Erat, O., Isop, W. A., Kalkofen, D., and Schmalstieg, D., "Drone-Augmented Human Vision: Exocentric Control for Drones Exploring Hidden Areas," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 24, No. 4, April 2018, pp. 1437–1446.

[15] Prabhakar, A., "Breakthrough Technologies for National Security," *Defense Advanced Research Projects Agency (DARPA), Tech. Rep.*, 2015.

[16] Las Fargeas, J., Kabamba, P., and Girard, A., "Cooperative Surveillance and Pursuit Using Unmanned Aerial Vehicles and Unattended Ground Sensors," *Sensors*, Vol. 15, No. 1, 2015, pp. 1365–1388.

[17] Kim, K. and Davidson, J., "Unmanned Aircraft Systems Used for Disaster Management," *Transportation Research Record*, Vol. 2532, No. 1, 2015, pp. 83–90.

[18] "Global Hawk Aids in Philippine Relief Efforts," *Cision PR Newswire*, Apr 2014, https://www.prnewswire.com/news-releases/global-hawk-aids-in-philippine-relief-efforts-238865571.html.

[19] Friederike Alschner, Jessica DuPlessis, D. S., "Case Study No 9: Using Drone Imagery for real-time information after Typhoon Haiyan in The Philippines," Tech. rep., The Swiss Foundation for Mine Action (FSD), Aug 2016, https://drones.fsd.ch/wp-content/uploads/2016/06/9Case-Study.Philippine.SearchRescue.3May2016.pdf.

[20] Cruz, H., Eckert, M., Meneses, J., and Martínez, J.-F., "Efficient forest fire detection index for application in unmanned aerial systems (UASs)," *Sensors*, Vol. 16, No. 6, 2016, pp. 893.

[21] Techy, L., Schmale III, D. G., and Woolsey, C. A., "Coordinated aerobiological sampling of a plant pathogen in the lower atmosphere using two autonomous unmanned aerial vehicles," *Journal of Field Robotics*, Vol. 27, No. 3, 2010, pp. 335–343.

[22] Chan, B., Guan, H., Jo, J., and Blumenstein, M., "Towards UAV-based bridge inspection systems: A review and an application perspective," *Structural Monitoring and Maintenance*, Vol. 2, No. 3, 2015, pp. 283–300.

[23] Hallermann, N. and Morgenthal, G., "Visual inspection strategies for large bridges using Unmanned Aerial Vehicles (UAV)," *Proc. of 7th IABMAS, International Conference on Bridge Maintenance, Safety and Management*, 2014, pp. 661–667.

[24] Guerrero, J. A. and Bestaoui, Y., "UAV path planning for structure inspection in windy environments," *Journal of Intelligent & Robotic Systems*, Vol. 69, No. 1-4, 2013, pp. 297–311.

[25] Endsley, M. R., "Design and evaluation for situation awareness enhancement," *Proceedings of the Human Factors Society annual meeting*, Vol. 32, SAGE Publications Sage CA: Los Angeles, CA, 1988, pp. 97–101.

[26] Bernhard, R. P., Christiansen, E. L., and Kessler, D., "Orbital debris as detected on exposed spacecraft," *International journal of impact engineering*, Vol. 20, No. 1-5, 1997, pp. 111–120.

[27] Hayati, S., Balaram, J., Seraji, H., Kim, W. S., Tso, K., and Prasad, V., "Remote surface inspection system," *Proc. of the 1993 IEEE International Conference on Robotics and Automation*, 1993, pp. 875–882.

[28] Micire, M., Fong, T., Morse, T., Park, E., Provencher, C., Smith, E., To, V., Torres, R. J., Wheeler, D., and Mittman, D., "Smart spheres: a telerobotic free-flyer for intravehicular activities in space," *AIAA Space 2013 Conference and Exposition*, 2013, p. 5338.

[29] Ackerman, E., "NASA Launching Astrobee Robots to Space Station," *IEEE Spectrum: Technology, Engineering, and Science News*, Apr 2019, https://spectrum.ieee.org/automaton/robotics/space-robots/nasa-launching-astrobee-robots-to-iss-tomorrow.

[30] Provencher, C., Bualat, M. G., Barlow, J., Fong, T. W., Smith, M. F., Smith, E. E., and Sanchez, H. S., "Astrobee: Space Station Robotic Free Flyer," .

[31] Aziz, S., "Lessons learned from the STS-120/ISS 10A robotics operations," *Acta Astronautica*, Vol. 66, No. 1-2, 2010, pp. 157–165.

[32] Ellis, S. R., "Collision in space," *Ergonomics in Design*, Vol. 8, No. 1, 2000, pp. 4–9.

[33] Carr, C. E., Schwartz, S. J., and Rosenberg, I., "A wearable computer for support of astronaut extravehicular activity," *Wearable Computers, 2002.(ISWC 2002). Proceedings. Sixth International Symposium on*, IEEE, 2002, pp. 23–30.

[34] Doule, O., Miranda, D., and Hochstadt, J., "Integrated Display and Environmental Awareness System-System Architecture Definition," *AIAA SPACE and Astronautics Forum and Exposition*, 2017, p. 5269.

[35] Fong, T., Nourbakhsh, I., Kunz, C., Fluckiger, L., Schreiner, J., Ambrose, R., Burridge, R., Simmons, R., Hiatt, L., Schultz, A., Trafton, J. G., Bugajska, M., and Scholtz, J., "The Peer-to-Peer Human-Robot Interaction Project," *SPACE 2005*, Long Beach, CA, 2005.

[36] Trafton, J. G., Cassimatis, N. L., Bugajska, M. D., Brock, D. P., Mintz, F. E., and Schultz, A. C., "Enabling effective human-robot interaction using perspective-taking in robots," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 35, No. 4, July 2005, pp. 460–470.

[37] Liu, B., Dousse, O., Nain, P., and Towsley, D., "Dynamic Coverage of Mobile Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 2, Feb 2013, pp. 301–311.

[38] Kwok, A. and Martínez, S., "Deployment Algorithms for a Power-constrained Mobile sensor network," *Proc. of the 2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 2008, pp. 140–145.

[39] Zhu, C., Zheng, C., Shu, L., and Han, G., "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, Vol. 35, No. 2, 2012, pp. 619–632.

[40] Liang, J., Liu, M., and Kui, X., "A Survey of Coverage Problems in Wireless Sensor Networks," *Sensors and Transducers*, 2014, pp. 240–246.

[41] Oktug, S., Khalilov, A., and Tezcan, H., "3D Coverage Analysis under Heterogeneous Deployment Strategies in Wireless Sensor Networks," *Proc. of the Fourth Advanced International Conference on Telecommunications*, 2008, pp. 199–204.

[42] Piciarelli, C., Micheloni, C., and Foresti, G. L., "Automatic reconfiguration of video sensor networks for optimal 3D coverage," *Proc. of the 2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras*, IEEE, 2011, pp. 1–6.

[43] Xie, L. and Zhang, X., "3D clustering-based camera wireless sensor networks for maximizing lifespan with minimum coverage rate constraint," *Proc. of the 2013 IEEE Global Communications Conference*, IEEE, 2013, pp. 298–303.

[44] Liu, B., Dousse, O., Nain, P., and Towsley, D., "Dynamic coverage of mobile sensor networks," *IEEE Transactions on Parallel and Distributed systems*, Vol. 24, No. 2, 2013, pp. 301–311.

[45] Stipanović, D. M., Valicka, C., Tomlin, C. J., and Bewley, T. R., "Safe and Reliable Coverage Control," *Numerical Algebra, Control and Optimization*, Vol. 3, 2013, pp. 31–48.

[46] Song, C., Liu, L., Feng, G., Wang, Y., and Gao, Q., "Persistent awareness coverage control for mobile sensor networks," *Automatica*, Vol. 49, No. 6, 2013, pp. 1867–1873.

[47] Smith, S. L., Schwager, M., and Rus, D., "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, Vol. 28, No. 2, 2012, pp. 410–426.

[48] Palacios-Gasós, J. M., Montijano, E., Sagüés, C., and Llorente, S., "Multi-robot persistent coverage with optimal times," *Proc. of the 55th IEEE Conference on Decision and Control*, IEEE, 2016, pp. 3511–3517.

[49] Song, C., Feng, G., Fan, Y., and Wang, Y., "Decentralized adaptive awareness coverage control for multi-agent networks," *Automatica*, Vol. 47, No. 12, 2011, pp. 2749 – 2756.

[50] Cheng, P., Keller, J., and Kumar, V., "Time-Optimal UAV Trajectory Planning for 3D Urban Structure Coverage," *Proc. of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, Sept. 2008, pp. 2750–2757.

[51] Pasqualetti, F., Zanella, F., Peters, J. R., Spindler, M., Carli, R., and Bullo, F., "Camera network coordination for intruder detection," *IEEE Transactions on Control Systems Technology*, Vol. 22, No. 5, 2014, pp. 1669–1683.

[52] Leahy, K., Zhou, D., Vasile, C.-I., Oikonomopoulos, K., Schwager, M., and Belta, C., "Persistent surveillance for unmanned aerial vehicles subject to charging and temporal logic constraints," *Autonomous Robots*, Vol. 40, No. 8, 2016, pp. 1363–1378.

[53] Yang, M., Kim, D., Li, D., Chen, W., Du, H., and Tokuta, A. O., *Sweep-Coverage with Energy-Restricted Mobile Wireless Sensor Nodes*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 486–497.

[54] Mitchell, D., Corah, M., Chakraborty, N., Sycara, K., and Michael, N., "Multi-robot long-term persistent coverage with fuel constrained robots," *Proc. of the 2015 IEEE International Conference on Robotics and Automation*, IEEE, 2015, pp. 1093–1099.

[55] Peters, J. R., Srivastava, V., Taylor, G. S., Surana, A., Eckstein, M. P., and Bullo, F., "Human Supervisory Control of Robotic Teams: Integrating Cognitive Modeling with Engineering Design," *IEEE Control Systems*, Vol. 35, No. 6, Dec 2015, pp. 57–80.

[56] Sarter, N. B. and Woods, D. D., "Team Play with a Powerful and Independent Agent: Operational Experiences and Automation Surprises on the Airbus A-320," *Human Factors*, Vol. 39, No. 4, 1997, pp. 553–569.

[57] Shanker, T. and Richtel, M., "In New Military, Data Overload Can Be Deadly," *The New York Times*, 2011.

[58] Dragan, A. D., Bauman, S., Forlizzi, J., and Srinivasa, S. S., "Effects of Robot Motion on Human-Robot Collaboration," *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, New York, NY, 2015.

[59] Fisac, J. F., Liu, C., Hamrick, J. B., Sastry, S., Hedrick, J. K., Griffiths, T. L., and Dragan, A. D., "Generating Plans that Predict Themselves," *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.

[60] Zhang, Y., Sreedharan, S., Kulkarni, A., Chakraborti, T., Zhuo, H. H., and Kambhampati, S., "Plan explicability and predictability for robot task planning," *Proc. of the 2017 IEEE International Conference on Robotics and Automation*, May 2017, pp. 1313–1320.

[61] Walker, M., Hedayati, H., Lee, J., and Szafir, D., "Communicating Robot Motion Intent with Augmented Reality," *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, ACM, New York, NY, USA, 2018, pp. 316–324.

[62] Hedayati, H., Walker, M., and Szafir, D., "Improving Collocated Robot Teleoperation with Augmented Reality," *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '18, ACM, New York, NY, USA, 2018, pp. 78–86.

[63] McGhan, C. L. and Atkins, E. M., "Human Productivity in a Workspace Shared with a Safe Robotic Manipulator," *Journal of Aerospace Information Systems*, Vol. 11, No. 1, 2014, pp. 1–18.

[64] Hoffman, G. and Breazeal, C., "Effects of anticipatory action on human-robot teamwork: Efficiency, fluency, and perception of team," *2007 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2007, pp. 1–8.

[65] Wang, Z., Deisenroth, M., Amor, H., Vogt, D., Schlkopf, B., and Peters, J., *Probabilistic modeling of human movements for intention inference*, Vol. 8, MIT Press Journals, 2013, pp. 433–440.

[66] Baker, C. L., Saxe, R., and Tenenbaum, J. B., "Action understanding as inverse planning," *Cognition*, Vol. 113, No. 3, 2009, pp. 329 – 349.

[67] Liu, C., Hamrick, J. B., Fisac, J. F., Dragan, A. D., Hedrick, J. K., Sastry, S. S., and Griffiths, T. L., "Goal Inference Improves Objective and Perceived Performance in Human-Robot Collaboration," *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, Richland, SC, 2016, pp. 940–948.

[68] Liu, C., Liu, S.-Y., Carano, E. L., and Hedrick, J. K., "A Framework for Autonomous Vehicles With Goal Inference and Task Allocation Capabilities to Support Peer Collaboration With Human Agents," *Proceedings of the ASME 2014 Dynamic Systems and Control Conference*, San Antonio, TX, 2014.

[69] Zhang, J., Zhuang, L., Wang, Y., Zhou, Y., Meng, Y., and Hua, G., "An egocentric vision based assistive co-robot," *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1–7.

[70] Damen, D., Haines, O., Leelasawassuk, T., Calway, A., and Mayol-Cuevas, W., "Multi-user egocentric online system for unsupervised assistance on object usage," *European Conference on Computer Vision*, Springer, 2014, pp. 481–492.

[71] Damen, D., Leelasawassuk, T., and Mayol-Cuevas, W., "You-Do, I-Learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance," *Computer Vision and Image Understanding*, Vol. 149, 2016, pp. 98–112.

[72] Nister, D. and Stewenius, H., "Scalable recognition with a vocabulary tree," *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, Vol. 2, IEEE, 2006, pp. 2161–2168.

[73] Goto, M., Uematsu, Y., Saito, H., Senda, S., and Iketani, A., "Task support system by displaying instructional video onto AR workspace," *2010 IEEE International Symposium on Mixed and Augmented Reality*, Oct 2010, pp. 83–90.

[74] Hitchcock, A. and Sung, K., "Multi-view augmented reality with a drone," *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, ACM, 2018, p. 108.

[75] Wen, M.-C. and Kang, S.-C., "Augmented Reality and Unmanned Aerial Vehicle Assist in Construction Management," *2014 International Conference on Computing in Civil and Building Engineering*.

[76] Papachristos, C. and Alexis, K., "Augmented reality-enhanced structural inspection using aerial robots," *2016 IEEE International Symposium on Intelligent Control (ISIC)*, Buenos Aires, Argentina, Sept 2016.

[77] Nith, R. and Rekimoto, J., "Falconer: A Tethered Aerial Companion for Enhancing Personal Space," 2019.

[78] Sandor, C., Cunningham, A., Dey, A., and Mattila, V.-V., "An augmented reality X-ray system based on visual saliency," *2010 IEEE International Symposium on Mixed and Augmented Reality*, IEEE, 2010, pp. 27–36.

[79] Karami, A., Jeanpierre, L., and Mouaddib, A., "Human-robot collaboration for a shared mission," *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2010, pp. 155–156.

[80] Lasota, P., Nikolaidis, S., and Shah, J. A., "Developing an adaptive robotic assistant for close proximity human-robot collaboration in space," *AIAA Infotech@ Aerospace (I@ A) Conference*, 2013.

[81] McGhan, C. L., Nasir, A., and Atkins, E. M., "Human intent prediction using markov decision processes," *Journal of Aerospace Information Systems*, Vol. 12, No. 5, 2015, pp. 393–397.

[82] Howe, E. J., Buckland, S. T., Desprs-Einspenner, M.-L., and Khl, H. S., "Distance sampling with camera traps," *Methods in Ecology and Evolution*, Vol. 8, No. 11, 2017, pp. 1558–1565.

[83] Buckland, S. T., Rexstad, E. A., Marques, T., and Oedekoven, C. S., *Distance Sampling: Methods and Applications*, Springer, 2015.

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

[85] Hexsel, B., Chakraborty, N., and Sycara, K., "Distributed coverage control for mobile anisotropic sensor networks," Tech. Rep. CMU-RI-TR-13-01, Robotics Institute, Pittsburgh, PA, January 2013.

[86] Atinc, G., Stipanovíc, D. M., and Voulgaris, P. G., "Supervised coverage control of multi-agent systems," *Automatica*, Vol. 50, No. 11, 2014, pp. 2936–2942.

[87] Slattery, J. C., *Advanced Transport Phenomena*, Cambridge UP, 1999.

[88] Saha, B., Koshimoto, E., Quach, C. C., Hogge, E. F., Strom, T. H., Hill, B. L., Vazquez, S. L., and Goebel, K., "Battery health management system for electric UAVs," *2011 Aerospace Conference*, IEEE, 2011, pp. 1–9.

[89] Mansouri, S. S., Karvelis, P., Georgoulas, G., and Nikolakopoulos, G., "Remaining useful battery life prediction for UAVs based on machine learning," *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 4727–4732.

[90] Panagou, D., Stipanović, D. M., and Voulgaris, P. G., "Distributed coordination control for multi-robot networks using Lyapunov-like barrier functions," *IEEE Trans. on Automatic Control*, Vol. 61, No. 3, March 2016, pp. 617–632.

[91] Ma, X., Jiao, Z., Wang, Z., and Panagou, D., "Decentralized prioritized motion planning for multiple autonomous UAVs in 3D polygonal obstacle environments," *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 292–300.

[92] Wills, A. G. and Heath, W. P., "A recentred barrier for constrained receding horizon control," *Proc. of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, Vol. 5, May 2002, pp. 4177–4182 vol.5.

[93] Vincenty, T., "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Survey review*, Vol. 23, No. 176, 1975, pp. 88–93.

[94] Vincenty, T., "Geodetic inverse solution between antipodal points," Scanned by Charles Karney from the copy in R.H. Rapp's library at Ohio State University. The report is a work of the U.S. Government and so is in the public domain.

[95] Ivory, J., "VIII. A New Series for the Rectification of the Ellipsis; together with some Observations on the Evolution of the Formula $(a^2 + b^2 - 2ab\cos\theta)^n$," *Transactions of the Royal Society of Edinburgh*, Vol. 4, No. 2, 1798, pp. 177–190.

[96] Schmidt, R. F., *Fundamentals of Sensory Physiology*, Springer-Verlag Berlin Heidelberg, 3rd ed., 1986, p. 159.

[97] Hunziker, H., *Im Auge Des Lesers [The eye of the reader: foveal and peripheral perception - from letter recognition to the joy of reading] (in German)*, Zürich: Transmedia Stubli Verlag, 2006.

[98] Thrun, S., Burgard, W., and Fox, D., *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, 2005, pp. 80–82.

[99] Hoover, A. W., "Lecture Notes: Particle Filter," lecture notes.

[100] Bishop, C. M., *Patern Recognition and Machine Learning*, Springer, 2006, pp. 435–439.

[101] Miller, G. A., "The magical number seven, plus or minus two: Some limits on our capacity for processing information." *Psychological review*, Vol. 63, No. 2, 1956, pp. 81.

[102] Lewis, F. L. and Liu, D., *Reinforcement learning and approximate dynamic programming for feedback control*, John Wiley & Sons, 2013, pp. 205–209.

[103] Wu, C. J. and Hamada, M., *Experiments: Planning, Analysis, and Optimization*, chap. 2, Wiley, 2nd ed., 2009.

[104] Sauro, J., "Measuring Usability with the System Usability Scale (SUS)," .

[105] Howell, D. C., *Statistical Methods for Psychology*, chap. 5, PWS-Kent Pub. Co., 7th ed., 2010.

[106] Anderson, D. and Buck, C., "Free-flying camera design for Space Station EVA and telerobotic operations," *Space Programs and Technologies Conference and Exhibit*, September 1994.

[107] Atkins, E. M., Lennon, J. A., and Peasco, R. S., "Vision-based following for cooperative astronaut-robot operations," *Proceedings, IEEE Aerospace Conference*, March 2002.

[108] Stolen, M. F., Dillow, B., Jacobs, S. E., and Akin, D. L., "Interface for EVA human-machine interaction," Tech. rep., SAE Technical Paper, 2008.

[109] Hall, S., *Model Following Control Strategies and Human Interface Techniques for the Treatment of Time Delay During Teleoperation*, Ph.D. thesis, University of Maryland, 2004.

[110] Nourmohammadi, A., Jafari, M., and Zander, T. O., "A Survey on Unmanned Aerial Vehicle Remote Control Using BrainComputer Interface," *IEEE Transactions on Human-Machine Systems*, Vol. 48, No. 4, Aug 2018, pp. 337–348.

[111] Coenen, J., *UAV BCI comparison of manual and pedal control systems for 2D flight performance of users with simultaneous BCI control*, Ph.D. thesis, Radboud University Nijmegen, 2015, Bachelor's Thesis.

[112] "Six Paths to the Nonsurgical Future of Brain-Machine Interfaces," *Defense Advanced Research Projects Agency*, May 2019, https://www.darpa.mil/news-events/2019-05-20.

[113] Lygeros, J., "Lecture notes on hybrid systems," https://people.eecs.berkeley.edu/ sastry/ee291e/lygeros.pdf.

[114] Kaczor, W. J. and Nowak, M. T., *Problems in mathematical analysis II*, American Mathematical Society, 2003.

[115] Hardy, G. H., Littlewood, J. E., and Pólya, G., *Inequalities*, Cambridge UP, 1952, pg. 43.