An Algorithm for Distributed Heterogeneous Robot-Human Teams

By

S M Al Mahi

Bachelor of Science in Computer Science
Islamic University of Technology
Gazipur, Dhaka, Bangladesh
2011

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
Doctor of Philosophy
May 2020

An Algorithm for Distributed Heterogeneous Robot-Human Teams

Dissertation Approved:

Dr. Christopher John Crick

Dissertation Adviser

Dr. Christopher John Crick

Dr. Johnson P Thomas

Dr. Nohpill Park

Dr. Guoliang Fan

ACKNOWLEDGEMENTS

This dissertation is the result of years of close collaboration with my advisor Dr. Christopher Crick. During this collboration I have learned a lot from him. I sincerely thank Dr. Crick for his contineous support and guidance. I also thank the Cloud Map which is a research collboration among researchers from a very diverse background. This collaboration specially helped to collect data for my studies. I thank all of the members of Cloud Map.

To my parents —
Syed Nurul Alam and Rukhshanara Begum
&
Sisters Shohanee and Anu.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: S M Al Mahi

Date of Degree: May 2020

Title of Study: An Algorithm for Distributed Heterogeneous Robot-Human Teams

Major Field: Computer Science

Abstract:

This dissertation presents a set of three closely related studies conducted by me during my Doctoral studies. The studies focus on two immensely important aspects of Robotics; control of the cooperative multi-robot system and human-robot interactions. I have taken several attempts to understand these two aspects. In particular, I have investigated the autonomous control of the multi-robot system which uses a distributed algorithm for autonomous decision making and also facilitating human interaction with the robots. I have found that it could provide a lot of insights from the field of robotic perception and control. I have also used these examples to apply them in different heterogeneous multi-robot systems. The result of my research work is an integrated control model for a human and multi-robot team system. My study discovered an important knowledge gap. My research also innovated novel tools with a good theoretical foundation to address the research gap. My study also validated the result with real-world data collected during different thoroughly executed experiments. My research can be carefully organized into three major studies that have been well documented and published in renowned scientific venues. These three studies together cover a multitude of dimensions of control of multi-robot systems and human-robot interactions with multi-robot systems. These studies involve extensive research, application design, engineering, and development of heterogeneous multi-robot systems with a focus on human-robot interaction. In this dissertation, I have comprehensively documented my studies. Therefore, I firmly believe my research has been contributed to the field of Robotics and improved our understanding of multi-robot teams and human interaction.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure **Page**

Figure **Page**

Figure **Page**

LIST OF TABLES

Chapter I

Introduction

"I visualize a time when we will be to robots what dogs are to humans, and I am rooting for the machines."—Claude Shannon

What makes a robot autonomous? It is one of the active fields of research in Robotics and Artificial Intelligence. By autonomous it implies it can take its own decision automatically while performing a task. But still one can argue that we are far from reaching truly autonomous robots. Yes, a robot can move from point A to point B very efficiently in an ideal setting. A plethora of eminent research[1, 2] has conducted on this problem in the field of robotics. Research on localization[3, 4, 5, 6] and mapping to the environment has been studied in the literature in-depth as well. Yet robots are still not very good figuring out a good strategy to accomplish a task instructed by humans unlike a human would do. Even when it does it is not good at telling its understanding of its human partner in an intuitive way that human understands. Thus bridges between robots and human understanding of a problem and coming up with a solution is an important research question in the field of human-robot interaction (HRI). For many good reasons, these robots should be autonomous for most of the time and heterogeneous of type. The term heterogeneous implies that there is more than one robot with different sensors and actuators.

In addition to humans, an autonomous or semi-autonomous robot may also need to communicate and coordinate with other robots. That is when we come across the problem of coordination of the multi-robot team. This requires the robots to communicate over and over again with one another, to make sure they are not being used in the same place or have been used in the same place for more than a short period of time and that the robots know how to avoid collision with each other accomplishing a task. It's also very important that the robots know how far apart things have to be so they can coordinate better and avoid each other. An autonomous or semi-autonomous robot needs to have safety measures in place so that it will not trip or otherwise damage itself (such as a broken arm or a broken window). When the robot is being used, it also needs to be ready to move out of its intended area if damage occurs. When the robot is not being used, it must do everything safely without stopping and be ready to move away from another robot or obstacle if necessary. One may ask why do need multiple robots to accomplish tasks in the first place? The answer is because even a fully autonomous robot may often lack the capability to complete a complex task. We find in nature intelligent beings that communicate with each opt to coordinate and help each other to accomplish and objective.

Finally, a coordinated autonomous multi-robot requires human intervention and interaction as well. For example, a team of robots capable of performing a task in a trained environment may not be able to do the same in a completely new environment. To avoid such situations an intelligent robot program needs to recognize the requirements of each task and provide the necessary information and resources. Secondly, and equally important, it is the responsibility of the human team to recognize and react to those errors and problems that arise during the mission. Apart from safety, human domain knowledge integration is another important factor which makes human interaction with multiple robot team is essential. Throughout my studies, for my doctoral degree, I researched the above research problems in the light of state of the art research in autonomous multi-robot systems. I have carefully figured out important challenges in solving those problems. I have conducted rigorous experiments to validate my understanding of the problems and also provided elegant solutions to those. In this dissertation thesis, I will describe three of my major studies in the following chapters. Each of these studies has been published peer-reviewed conferences and scientific venues. Although all of the studies are done on human-robot interaction and control of heterogeneous autonomous robot they modularized into individual research works for their own merits.

1.1 Assessing Cognitive Stress of Human Operator

In this study I demonstrated a robotic system that learns to recognize the behavioral indicators that a complex, rapidly-evolving task has exceeded the cognitive capacity of a human partner. Based on that determination, it can act autonomously to reduce the human decision burden, significantly improving task performance. We present a robot that learns to recognize the indications that a complex, rapidly-evolving task has exceeded the cognitive capacity of a human partner to provide helpful direction. The robot learns to associate human directions and task quality metrics in the context of a well-understood task, in this case, the navigation of a maze. The robot is then able to use this learned model to evaluate the behavior of a human partner in a task which it does not understand and cannot execute without human instruction. Even though the context of the new task is beyond the robot's comprehension, it can accurately assess whether it is being given trustworthy directions from an increasingly frazzled human partner.

One of the most challenging obstacles facing human-robot teams is the inherent communication barrier between the two. Human operators, at least once they have received training, have some notion concerning the capacities of their mechanized partners, but the ability of robots to assess the limitations of humans has not received adequate attention. Research in this area often focuses on attempting to observe human behavior and predict what action or actions to perform in the future; this renders such systems incapable of making instantaneous or reactive decisions. In contrast, systems which are capable of making split-second decisions, such as the lane drift detection found in some high-end cars, make no inference concerning the user's abilites or frame of mind; they are reacting to a well-understood world state without consulting their human partners. This is not true interaction; the robot is learning to work *around* the user instead of with them. Human assistance should enable complex multi-robot tasks where the robots themselves are unable to assess their environment fully, but this lays a heavy burden on the operator in a dynamic, dangerous, rapidly-changing environment with many cognitive demands. Fundamentally, human operators and robots each have complementary capabilities and limitations, and they must each be aware of the abilities of the other. Discovering the cognitive capacity of human operators in human-robot teams is essential [7]. Our research allows robots to form models of

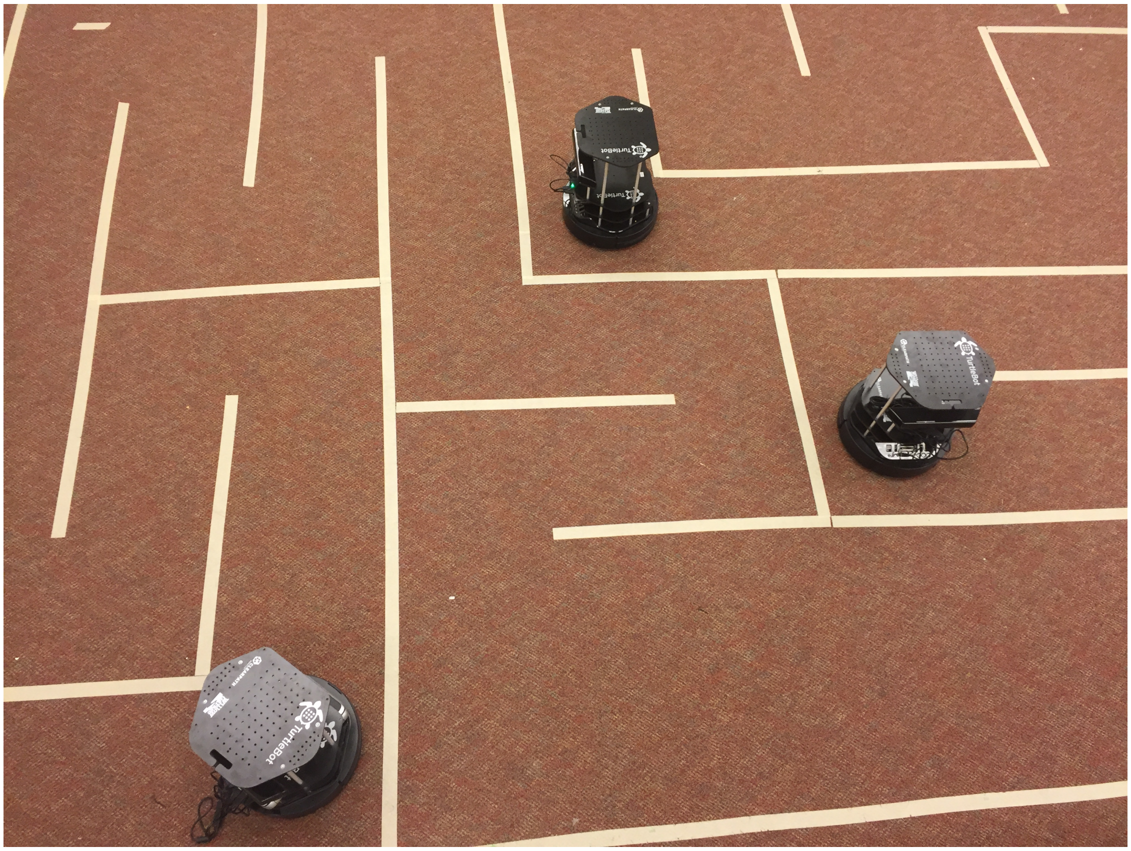


Figure 1.1: Turtlebots navigate a maze while evaluating human task input.

human behavior during well-understood tasks, and then apply these learned models during unknown tasks. We show that these models allow the robot to make an effective determination of the cognitive capacity of a human partner, even when the robot cannot directly assess the task it is being asked to perform. In this way, a robot team member should be able to fall back into a safe autonomous mode whenever the task demands begin to exceed the ability of its human partner to provide effective direction.

Why is teamwork between human and robots so important? Both have advantages and limitations. It is easy to understand that a robot with flawed sensors and actuators may not be able to perform a task, owing to failure to perceive information about its environment. But it is not necessarily the case that improving the sensors and actuators will have a commensurate effect on the robot's ability to accomplish a task – a great many tasks require contextual information far beyond the current ability of any robot to reason about. Thus for the forseeable future, many tasks will require assistance from human teammates, who are able to integrate data, rely on experience to predict the effects of actions on world states, and produce effective long-term plans far better than any current robot.

Human assistance with multiple robots has been demonstrated in different practical situations. For example, they have been used as customer assistants in shopping malls [8, 9], where human operators occasionally assisted robots. Human-robot teams have assisted each other in museum tour scenarios [10] and in warehouse inventory management [11]. Just as robots, regardless of their hardware capabilities, do not necessarily perform well without assistance, humans do not always assist robots as efficiently as they might, despite their superiority in context sensitivity and general intelligence. Several issues arise in this regard [12], including obvious problems such as noisy communication between human operators and robots, accidental damage to robot sensors and so on.

In this paper, we focus on one problem which can cause difficulties in a human-robot team: the cognitive capacity of the human operator. Humans often find their ability to function effectively challenged, due to psychological stress, tiredness, or overwhelming task demands. A robot's ability to participate constructively in a human-robot team will benefit immensely from understanding and accommodating this cognitive stress appropriately. For example, for the pilot of an unmanned aircraft, cognitive stress can be the difference between life and death [13]. If such a robot can detect the emergence of cognitive stress in its operator, it can increase its level of autonomy and reduce its demands on the operator's attention. Hopefully, such a robot would wait safely for a more opportune moment or decide to engage in a less cognitively challenging task, rather than continue to follow the direction of a human who is no longer able to provide appropriate assistance.

In this paper, we have designed robots that learn the correlations between quantifiable behavior metrics and the cognitive capabilities of human operators. We designed a maze game (Fig. 1.1) where multiple robots are given directions by a single human operator. At first, the robots' objective was simply to complete the maze, a task that they were capable of executing without human assistance using autonomous path planning. They were therefore able to determine whether the instructions of their operators were sound or questionable, and associated these outcomes with measurements of their operator's behavior. We observed the output of the learned cognitive stress model in a different task, this time with the robots engaged in a coin-collecting game inside the maze. Although the robots had no knowledge of the rules or objectives of the coin-collecting game, and had no ability to sense the game's context, their learned models enabled them to determine the cognitive stress of their human partner. In this way, they were able to evaluate the trustworthiness of the actions they were being asked to perform. The cognitive stress discerned by the robot correlates with the true stress experienced by the human operators, as quantified by their self-reporting and by expert evaluation.

One of the most challenging obstacles facing human-robot teams is the inherent communication barrier between the two. Human operators, at least once they have received training, have some notion concerning the capacities of their mechanized partners, but the ability of robots to assess the limitations of humans has not received adequate attention.

In our system, the robot learns to model the relationship between human direction and task performance for a well-understood task–in this case, navigating a maze. The robot then participates in a different, more difficult problem, but it can still use its learned model to evaluate a human operator's cognitive load. A robot's ability to participate constructively in a human-robot team will benefit immensely from understanding and accommodating this cognitive stress appropriately [13]. Our work demonstrates robots that can detect the emergence of cognitive stress in their operators, increasing their level of autonomy and reducing demands on the operator's attention.

Human-robot interactions can be evaluated using fundamental metrics [14]. We leverage this data to inform our robots' estimation of a human operator's cognitive capacity. Recent work [15, 16] presented a model for assessing a human's attention level, based on eye contact and gaze detection towards a robot. In our work, the robot learns a general behavior model to identify the operator's cognitive threshold, rather than relying on the specifics of gaze.

Human- robot interactions can be evaluated using fundamental metrics [14] like task effectiveness (TE), neglect tolerance (NT), free time (FT), fan out (FO). Physiological metrics as objective evaluation along with subjective evaluation for cognitive load estimate has been also studied in psychology[17].

1.2 Multi-Modal Multi sensor Interaction between Human and Heterogeneous Multi-Robot System

I introduce a novel multi-modal multi-sensor interaction method between humans and heterogeneous multi-robot systems. I have also developed a novel algorithm to control heterogeneous multi-robot systems. The proposed algorithm allows the human operator to provide intentional cues and information to a multi-robot system using a multimodal multi-sensor touchscreen interface. My proposed method can effectively convey complex human intention to multiple robots as well as represent robots' intentions over the spatiotemporal domain. The proposed method is scalable and robust to dynamic change in the deployment configuration. I describe the implementation of the control algorithm used to control multiple quad-rotor unmanned aerial vehicles in simulated and real environments. I will also present my initial work on human interaction with the robots running my algorithm using mobile phone touch screens and other potential multimodal interactions.

In recent years, robots are becoming more ubiquitous and available to the general public than ever before, thanks to the rapid development of artificial intelligence and technologies like 3D printing, open-source autopilots, mission planners, and so on. Small Unmanned Aerial Systems (UAS) probably are the most common robots that people, in general, have access to today. However, there is a considerable research gap in human-UAS interaction. UAS may have been become popular for their use in photography. They have other useful applications as well. UAS have been used effectively for surveying, agriculture, search and rescue operations and so on. The means of interaction between UAS and humans has usually been joystick controllers, which can communicate control commands using radio communication mostly for the navigation intended by the human operator. Observation of the navigation space from the ground station or the attached camera on the UAS is critical for flight safety. However, UAS can be equipped with other sensor devices as well, to capture information such as temperature, humidity, and pressure. Although this sensory information can be crucial in many applications, it is not available via visual navigation or observation from the ground station, and cannot be performed in the most effective way using just the joystick controller.

I propose using a mobile phone touch screen as an interface between UAS (and robots in general) for these kinds of complex interactions. One of the benefits of choosing mobile phone touch screen is that many people have access to one, and thus it is the most accessible way for people, in general, to interact with robots today. However, there is an enormous research gap here as well. Most of the commercial UAS drones usually provide mobile phone applications to interact with their robots, but these applications are focused on setting up control parameters and communication channels, basic control commands, and executing predetermined maneuvers. Multimodal multi-sensory interactions are often overlooked and thus have tremendous potential for research.

The primary motivation for this work is to collect weather data using multiple UAS in a low-altitude environment. This type of task requires the use of a heterogeneous team of robots where different UAS can be equipped with different sensors. For example, flying fixed-wing UAS, which are fast but neither precise nor maneuverable, can be used to scout a large area of interest. Small quad-rotors with lower velocity but higher precision and maneuverability can be used to survey a smaller area of interest more extensively. They may also need to coordinate with ground robots for efficient deployment. Robots must coordinate with each other in general, ensuring collision avoidance among robots, a suitable spread of the UAS throughout the operating area, and deploying UAS equipped with the right sensors to investigate particular phenomena discovered in particular areas of interest, where the UAS using the specific sensor can collect essential data. I have developed a distributed algorithm to control heterogeneous teams of robots using factor graphs and loopy belief propagation. This algorithm also facilitates arbitrary human input to influence the behavior of a particular robot which is then propagated to other robots. I propose a user interface that directly visualizes the robots' intention in the space and allows the human operator to interact with it. I have already developed the fundamental functionality of the interface using a mobile web application.

1.3 Distributed Control of Heterogeneous Team of robots and Humans

In this work, I introduced a novel, scalable, distributed decision-making algorithm using factor graphs and the sum product algorithm to control the coordination of a heterogeneous multi-robot team in exploration tasks. In addition, our algorithm supports seamless participation of human operators at arbitrary levels of interaction. This dissertation presents experimental results performed using both simulated and actual teams of unmanned aerial systems (UAS). Our experiments demonstrate effective exploration while facilitating human participation with the team. At the same time, my research shows how robots with differing capabilities coordinate their behaviors effectively to leverage each other's individual strengths, without having to explicitly account for every possible joint behavior during system design. My research demonstrate my algorithm's suitability for tasks such as weather data collection using a heterogeneous robot team consisting of fixed- and rotary-wing UAS. In particular, during 60 flight hours of real-world experiments collecting weather data, my research shows that robots using our algorithm were about seven times more efficient at exploring their environment than similar systems which flew preplanned flight profiles. One of our primary contributions is to demonstrate coordinated autonomous control and decision-making among robots operating in very different flight regimes.

In many applications, it is advantageous to have a heterogeneous group of agents cooperating in their execution of tasks and their search for solutions. For example, a search and rescue robot team might incorporate unmanned aerial systems (UAS) to survey and map the affected area, small ground robots to infiltrate and explore rubble and pipes, large ground robots with manipulators to clear heavy objects, and friendly-looking social or medical robots to locate and make contact with the injured. Even similar robots might have very different kinds of capabilities and sensors, such as mapping sensors, hazardous chemical sensors, or sonic sensors for voids and places where people could be trapped. A heterogeneous robot team like the one described above is often difficult and expensive to engineer correctly, especially if the agents possess very different physical characteristics and abilities. In this paper, I am motivated primarily by the problem of using UAS to measure various aspects of weather systems. A large number of different kinds of UAS with different capabilities and sensors may be employed. For example, the system may incorporate large, fixed-wing UAS with a variety of sensors, smaller and less expensive rotary-wing UAS, disposable single-use sensors deployed with parachutes, and many airframe and sensor variations within these categories. Additionally, as projects like this develop, new varieties of agents may be added after the system is already in place.

The conventional way to approach this kind of problem would be to individually tailor the software for each type of agent to produce cooperation and other desired behavior. For

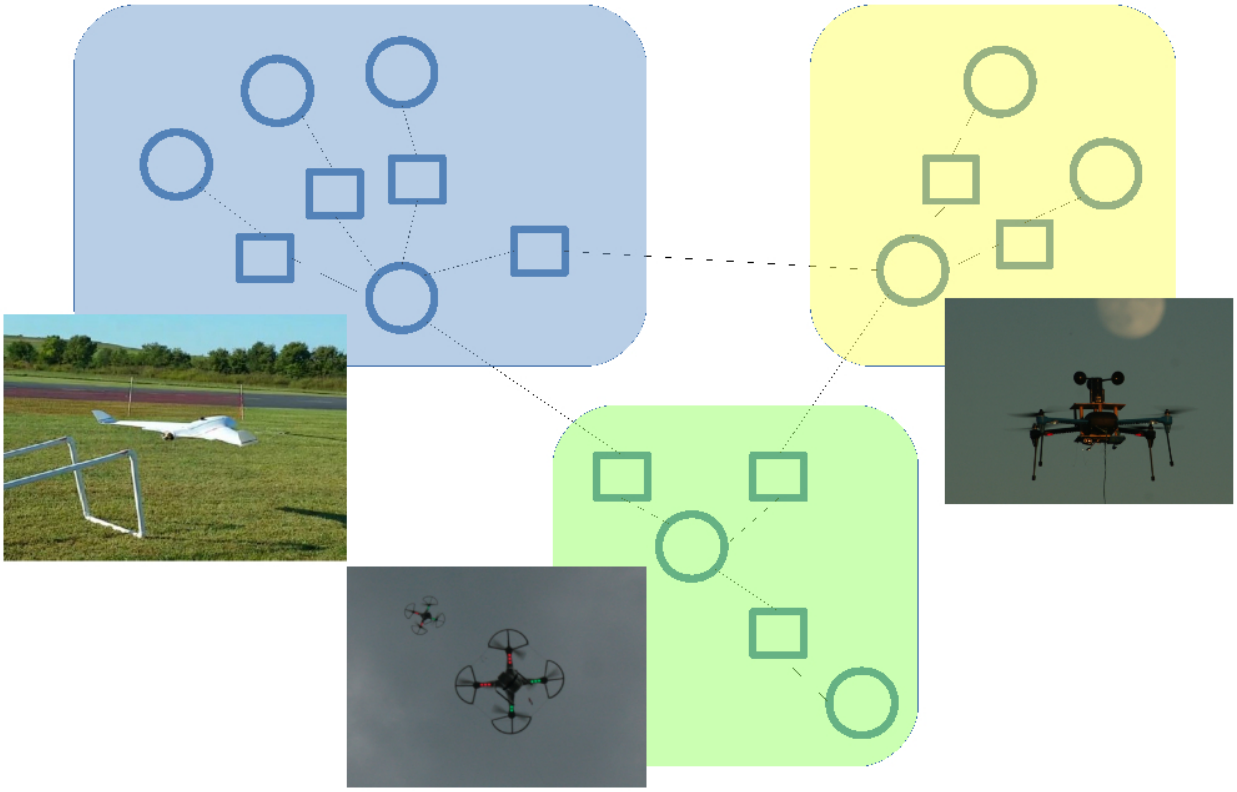


Figure 1.2: A team of heterogeneous UAS with meteorological sensors share a single factor graph representation of their shared intentions.

example, [18] in search and rescue (SAR) tasks, the logic of these two types of robot tasks are designed separately. This approach is often adequate when the different types of agent are not intended to change across missions and the desired collaborations are simple. However, as the number of agent types increases and the problems become more complex, this approach rapidly becomes unsustainable. The complexity of the entire system grows quadratically with respect to the number of agent types, potentially resulting in an unmanageable error-riddled code base and vastly increased cost and development time. Furthermore, such an approach becomes completely unmanageable when new varieties of agents are added after the system is deployed, since it requires significant modification to the logic of all other agent types every time a new variety is introduced. Our approach places these heterogeneous sensor and actuation modalities within a uniform multidimensional belief manifold represented by a factor graph shared among all of the agents in the system. The dimensions of the manifold can (and in many use cases, probably will) represent positions in physical space, but can also encode agents' joint intentions over any arbitrary space of action potentials. Actions to take with respect to other agents, obstacles, unexplored areas and points or gradients of interest can all be encoded in these intentional beliefs. The agents communicate in decentralized and asynchronous fashion, using loopy belief propagation to update the team's joint intentional state. Consensus beliefs are then acted upon by each agent in the local area of the manifold using simple gradient descent.

Many other similar tasks like SAR might require coordination of heterogeneous multi-robot teams with arbitrary human intervention. The task I have particularly investigated is surveying atmospheric data in the lower altitude (under 1000 feet) boundary layer[19]. Although satellite images, weather towers and balloons are used to collect weather data, they are not able to provide coverage in this important area of the atmosphere, even though sudden developments within the boundary layer can significantly contribute to severe weather formation. UAS have been used by meteorologists to collect data in the boundary layer, but they are manually controlled by human operators and usually follow a predefined simple mission plan, such as vertical profiles or circular orbits. Such an approach has drawbacks in efficiently deploying multiple robots and in fielding sufficient trained human operators. Proper coverage requires a multi-robot deployment, and spatiotemporal data can change dynamically, requiring multiple sensors spread out in space to properly sample time-varying data. Assigning human operators to each UAS is not practical, as robot participants must coordinate closely with one another and make data-dependent decisions across the entire robot team in real time. My research has addressed all of these problems using our factor graph algorithm. The application of our proposed algorithm is not limited to these particular tasks, but also can be generalized to many other tasks requiring heterogeneous multi- robot teams cooperating with human operators.

In many human-robot team tasks, the human operator cannot inform the robots of his or her intentions efficiently because of the different facets of the task exposed to the human operator and the robots. Similarly, it is often challenging to create an interface through which robots can efficiently convey their knowledge and intentions to their human operators, especially when the participant robots are heterogeneous in nature. This paper provides a theoretically neat, practically robust, and generally efficient model for heterogeneous, scalable and dynamic human-robot teams.

In the past decade multi-robot teams have been proposed for various use cases, i.e. weather surveys, rescue missions in natural or accidental disasters, and so on. However, most of these approaches are very task specific and it is almost always the case that robots are programmed very specifically to accomplish a certain type of task. While it is true that given a set of hardware capabilities a robot is capable of performing only a limited number of tasks, intuitively it is also true that deploying a collection of heterogeneous robots should be capable of achieving more sophisticated goals. Thus, I have focused on a distributed control algorithm that allows for such heterogeneity without requiring the constant, delicate design of functionality for each different robot for each new task. There are many recent works that focus on swarming and flocking behavior to demonstrate coordinated control of multi-robot teams, but such approaches are more difficult to apply to specific performance goals in real deployment scenarios. Our approach improves the ability of humans to participate seamlessly in the robots' collaborative goal development and intention formation, without a great deal of on-the-scene manual tweaking and twisting of parameters, configurations and task definitions.

Human operators needs to interact with robots seamlessly. The participation and interaction of human is necessary because because both human and robots have limitations about the information about real state of the world and take optimal actions to take to achieve a goal efficiently. For example, a group of UAVs can be given a task of modeling the presence of a cloud of a gas of interest in the environment. The sensor for detecting that gas can be attached to a some of the UAVs (Unmanned Aerial Vehicle). But the actual presence of the gas may relate to some other sensor as well. For human operator comprehending all the information and manually and control the team of robot becomes impossible with the inclusion of more robots with variety of sensors. Similar situation can be imagined for heterogeneous robot which has different actuators and having different capabilities participating the task. The participation and interaction of human is also very important because human operator might want to dynamically change the control and behavior of the human robot team. Thus we need a algorithm that controls the robots as well as incorporates human operator seamlessly. Moreover, cognitive science is proving to be more important in learning the from any data in general. The information that a robot can collect from the environment of the task might not too large to run and test algorithms that requires learning from huge amount of data and perform extensive computation on that data. Thus human cognition can be an important and often necessary part of learning about the real state of the world.

We have addressed above problems regarding multi-robot team with human operators. Me and my fellow researchers have proposed a factor graph model for robots and human operators where they exchange messages among them using loopy belief propagation. To prove our claim we have used simulated environment for multi robot human team. In my work we have defined the optimum intentions for a robot. In simulated environment our experiment showed that robots in the system can come to consensus among themselves using the algorithm. We have also showed that human can incorporate their intentions seamlessly using the same notion of intention. By calculating the optimum intention using a gradient descent algorithm one the consensus of the intention among robots and also human operator when they participated in the controlling. We have conducted several experiment in simplistic simulated environment. Our experiment showed that the simulate UAVs can survey the simulated environment efficiently. We compared the result with the baseline of simulated environment and we found that it can model the environment fairly. Simulated environment has proven to be very useful when developing this kind of algorithms for robots. Because it give the developer an opportunity to fine tune the algorithm before implementing it in actual hardware. Besides, one of our motivation for this research is to deploy actual robots in harsh environment. Running experiment with actual hardware is often expensive and there are possibilities of damaging the hardware.

Chapter II

Related Work

2.1 Assessing Cognitive Stress of Human Operator

In this section, we will briefly discuss work related to cognitive capacity assessment and human operation of multiple robots. Several research projects recently have investigated human cognitive capacity using different approaches.

Effectively navigating a maze game has been demonstrated in literature [20]. This research showed that robots learn more effectively from human operators if the learning took place in the context of features that the robot can easily understand. Counterintuitively, restricting the information available to a human operator led to better demonstrations and more effective learning. In this research, we show that similar metrics can be employed by the robot for the purpose of learning the cognitive threshold of a human operator.

Recent work [15, 16] presented a model for assessing a human's attention level, based on eye contact and gaze detection towards a robot. Based on the perceived attention level, the robot could generate an appropriate signal to obtain the attention of a targeted human. Attention is an important component of a human agent's cognitive capacity, but in our work, the robot learns a general behavior model to identify the operator's cognitive threshold, rather than relying on the specifics of gaze.

Human-robot interactions can be evaluated using fundamental metrics [14]. Such metrics relate to the cognitive capacity of human operators in obvious ways. For example, task effectiveness (TE) describes how efficiently robots complete a given task under human direction. For example, task effectiveness can be measured using the speed of the robot. In a navigation experiment, it may be the time taken by the robot to reach the goal. It could be defined as the difference between the time taken with and without human assistance. Another important metric is neglect tolerance (NT), which denotes a robot's level of autonomy. In static indoor environments, simple robots such as Turtlebots can easily engage in autonomous navigation. Even in complex environments with dynamic obstacles, clever algorithms [21] can enable such vehicles to navigate autonomously. Thus although it is a very important metric, we have focused on mostly TE because we think that that is the metric which can be exploited more consistently across different problems. Other potentially useful metrics include robot attention demand (RAD), free time (FT), fan out (FO) and interaction time. All of these could conceivably be included as inputs into a learned model such as ours.

Other efforts [22, 14] have presented very similar concepts of metrics for improving the efficiency of human-robot interaction. These principles include implicit mode switching among user interfaces, using human operators' natural cues, directly manipulating the world, manipulating the relationship between the robot and the world, supporting attention management and so on. Using similar terminology, other work [13] discovered how neglect time impacts important properties of human robot teams. They have shown a relation between the neglect time and the maximum number of robot that a single human operator can handle. We leverage this data to inform our robots' estimation of a human operator's cognitive capacity.

In another social experiment [8], where autonomous humanoid robots have been deployed to investigate their social acceptance, a scheduling algorithm has been used to assist the human operator. This experiment developed an algorithm that prioritizes the assitance provided by a human operator for a particular robot within a multi-robot team. The robot's task was to make conversation with interested shopping mall customers and to guide them to particular shelves corresponding to their needs. The shopping mall map was already known. Within critical areas inside the shopping mall environment, the robot needed assistance from humans, for example in unsafe locations or areas with glass walls that confused the robots' sensors. As there were multiple robots, a single operator could not assist them all simultaneously. The operator allocator algorithm therefore picked the robot most likely to encounter a critical region for assistance. In this work, the operator was assumed to be able to assist the robot without considering cognitive capacity. In contrast, we developed a model that enables the robot to make this determination and then perform the task accordingly.

A large amount of work [23, 24] has investigated human operators acting as teachers when interacting with a robot, for example helping a robot in kitchen environment. The human's demonstrations contribute to the robot's reward functions, using a modified reinforcement learning method that is based on the observation that human guidance is able to consider future reward along with past reward. The rich context that a human operator provides and that robots are very poor at reasoning about for themselves leads to improvement in the robot's learned behaviors. This observation only holds, however, as long as the human operator possesses the cognitive capacity to provide good, informative demonstrations. Our work allows robots to make this determination for themselves.

We have mentioned a number of research results that relate to ours in many ways. However, most current work does not explore the fact that, although humans are much more intelligent than robots, they nevertheless face significant limitations in their ability to assist their robot partners. One aspect of these limitations is that a human operator is vulnerable to task overload and psychological stress. Our work focuses particularly on human behavior in the face of this cognitive stress. Although engaging human assistance for the purpose of task learning is valuable, we argue that such a learning task is more effective if the robot simultaneously has the tools to evaluate the trustworthiness of the human's direction.

2.2 Multi-Modal Multi sensor Interaction between Human and Heterogeneous Multi-Robot System

A plethora of work has been done on multi-modal communication between robots and human operators using different methods. For brevity and specificity, I highlight some of the work pertaining to UAS, drones and similar robots. Capturing and recognizing gestures of human operators has been studied in recent works [25, 26]. Limitations and challenges of this approach have been studied in other work[27]. For example, the interaction might look rigid to an observer rather than natural, even though the movement as the gesture is natural. Another major limitation that my approach solves regarding gesture-based control is that using my method one the can provide more flexible control to the robot. Another limitation is the accuracy of detection of the gesture, especially for a novice user. Some of the above-mentioned work also used multimodal interaction with robots. Recent work [28] also presented a framework which compasses Natural User Interface (NUI) including voice, gesture, markers and a Graphical User Interface, i.e., windows, 2D graphics, 2D animations, and images. My work makes use of all these; however, I add a new perspective. In my work, robots' and human operators' belief about the world and their intentions are encoded in the GUI presentation, and I propose interaction in belief and intention space. While my proposed interface currently uses the smartphone touchscreen as the interaction medium, a NUI can also be incorporated using the accelerometer and gyroscope of the smartphone. Depth camera has been used for capturing gesture works[29] for indoor navigation.

Smart phone and tablets have been used to control drones in several works. In [30] a PC tablet was used to recognize and track down visual target for UAS. Other work[31] presented a framework for visualizing and tracking hostile drone in the airspace as an alternative to RADAR. The smartphones touch screen has been used as drone interaction interface [32, 33].

Prior work has used belief networks for coordinated control of multi-robot teams in the context of Markov Random Fields [34, 35, 36]. Gradient-based navigation for multi-robot navigation using potential field has been proposed by other works[37, 38]. However, these works do not incorporate human intentions and interaction. My approach not only addresses this research gap and limitations but also improves the state of the art by extending task complexity by incorporating human direction and advice.

2.3 Distributed Control of Heterogeneous Team of robots and Humans

A great deal of research has been conducted on multi-robot navigation, exploration and surveillance in different applications. Several recent works [39, 40] aimed at monitoring lower altitude atmospheric variables and sampling weather data using UAS and ground robots. However, very little work has been done to effectively coordinate multiple collaborating robots for such tasks, which demand fault tolerance, scalability, autonomous decision-making and human incorporation together. Our research is motivated by collecting weather data in the low-altitude (under 1000 feet) atmospheric boundary lay er in coordinated fashion. This requires a heterogeneous team of robots equipped with various sensors. Because of the large permutation of capabilities and constraints of these robots, human operators are subject to intense cognitive load while operating them. Teams of robots must often make autonomous decisions, even when attempting to satisfy conflicting mission goals. In this paper, we have proposed a loopy belief propagation [41] algorithm within a shared factor graph model. In our decentralized approach, no particular node in the factor graph is essential as long as the network maintains redundant communication pathways, and every robot continually updates its own intentional model with messages it happens to receive from robots within communication range. Thus our approach is fault tolerant as well as scalable. The computation required in loopy belief propagation can be distributed among different agents and each robot's computational requirements are much simpler compared to other methods[**?**] based on techniques such as Markov decision processes. Thus there are no technical barriers to adding arbitrary numbers of additional robots to the team. It is scalable in the spatial sense as well; our live experiments have been conducted in volumes as large as several kilometers across.

Using belief networks as coordination tools for multiple robots has been proposed in the past[**?**], usually in the context of Markov random fields [34, 35, 36]. However, our factor graph representation provides several advantages. First of all, the functions defined within the factor graph are often very simple to engineer. For example, collision avoidance is one of the commonly desired behaviors in a multi-robot exploration task. Instead of explicitly designing collision avoidance mechanisms, such as in [42, 43], we merely design a simple factor graph function which reacts to obstacle positions. Using our approach, complex team behaviors can be specified, tested and changed quickly. In addition, our formulation explicitly allows for seamlessly injecting human directives and advice into the robots' shared intentional framework, as well as additional expected behaviors, at arbitrary levels of specificity and timeliness. Moreover, many variants of MDPs, notably Partial Observable MDPs (POMDPs) are known to be intractable in larger domains.

In our approach, the robots can make autonomous decisions by following the gradient in the joint belief over the space. Gradient-based multi-agent navigation has been studied in other works[37, 38]. However, those studies focused on designing specific goals for a particular task, facilitating expansion of the behaviors. We have extended the idea by designing joint beliefs which can be devised to achieve any expected behavior from the robot team: following human intentions, for example. Additional complex behaviors can easily be introduced by designing new functions as factors in the factor graph.

The problems we consider involve heterogeneous communicative robots [44]. Such robots can be seamlessly incorporated into teams which can then form joint plans and task allocations around each participant's capabilities. This resource allocation problem has been studied extensively [45**?** , 46] from the underlying network topological perspective. Distributed robot teams are commonly considered to be nodes in a network, with connections among the various robots represented as edges. These edges represent communication links between robots or express other relationships of interest. This work demonstrates the graph-theoretic properties necessary in task allocation and team configuration of heterogeneous multi-robot teams. Our distributed probabilistic graphical model easily incorporates heterogeneity because a robot only adds sensor data that it is capable of gathering and only forms intentions over actions it is physically able to take, while continuing to update the factor graph and propagate the information provided by other agents with different capabilities.

For the simulation of multi-agent systems, ROS- Gazebo[47, 48], an open source simulation system, has been used by many academic researchers. It can be used for simulating motion planning in indoor or outdoor environments for teams of ground robots and UAVs [49]. However, it does not have built-in capability for simulating real-world outdoor environments, with varying winds, visibility, air density and turbulence. Moreover, one of our research focuses is to achieve a distributed control system that can be used even in harsh environments such as during severe storms. Simulating many of these environmental phenomena requires a huge amount of weather simulation work if it must be developed from scratch. A few other common problems with other simulators, with respect to our research interest, are extendability and availability. For example, AgentFly[50] a popular multi-agent flight simulator, facilitates control and planning in constrained environments. However, its control system is not customizable and it is not free. Many other commercially available simulators are expensive and out of reach of research community in general. We have used Flight Gear, which is an open source, freely available flight simulation platform mostly for fixed wing aircraft and helicopters. We have extended and customized it with various UAS flight models. Importantly, it also provides a customizable weather and visibility engine, which we have used to simulate specific boundary layer atmospheric phenomena, as well as

3-dimensional gas plumes from methane or carbon dioxide releases – phenomena which we are also able to generate and test in real-world applications. We report results in this paper from both our simulated system and a very extensive set of real-world experiments.

Chapter III

Technical Approaches

3.1 Assessing Cognitive Stress of Human Operator

Our experiments consisted of two games, maze navigation [20] and coin collection. All the games were played in two configurations, using either one or two robots, and with an interaction duration of two minutes. In the maze game, the robot collects data needed to build a model g for evaluating human cognitive load based on input H. In the subsequent coin game, the robot is placed in a different scenario where it has no access to success measures or even rules. Even so, with no independent means of measuring task success, it can still calculate $\hat{s} = g(H)$, and can therefore evaluate the quality of instruction, and hence the cognitive capacity, of its human partner.

In the maze game, the vector of environmental measurements E consists of the following components: e_0 is the *disparity* term, the distance between the navigation directions provided by a human and the route that the robot would have planned for itself, e_1 is the *collision* term, which penalizes collisions with walls, and e_2 is the *time delay* term, the amount of time taken for the human to guide the robot through the maze, compared with the robot's estimate of the time it would have taken under its own power. The computation of $s = f(E)$, the function for measuring success of the human directions, is a normalized summation of the elements of E.

By computing this value s, the robot can label its own data in order to train a supervised learning algorithm which will relate the success of a human-directed task with a set of measured behaviors H: h_0 is the *decision interval* term, which measures the time elapsed between the robot reaching a navigation goal and the human providing a new one, h_1 is the *error correction* term, which measures the tendency of a human operator to provide a navigation goal and then subsequently provide another before the task is complete, and h_2 is the *franticness* term, which characterizes erratic behavior for the control inputs. The robot's model incorporates the data learned from all participants.

3.1.1 Experimental design

In keeping with the problem statement, experiments were carried out in two parts: the Maze Game and the Coin Game experiments. In the first phase, the robot collected the data needed to build a model g for evaluating the trustworthiness of user input H. The robot is able to do this for the first phase because in the case of the Maze Game it has access to f and E and can calculate s. It understands the problem sufficiently to make such judgements; the robot requires no human aid to solve this problem.

In the subsequent Coin Game, the robot is placed in a different scenario, one in which it had no access to success measures or even rules; beyond the fact that it was moving in a similar environment to the Maze Game, it is wholly reliant on human direction to succeed in the task. Even so, with no independent means of measuring task success, it can still calculate $\hat{s} = g(H)$, and can therefore evaluate the quality of instruction, and hence the cognitive capacity, of its human partner.

Communication between operators and robots was achieved using the Robot Operating System (ROS)[51]. Users sent navigation goals to the robots by using a web interface developed for the project, which allowed the experiment to be conducted remotely (Fig 4.2).

3.1.2 Problem statement

Without loss of generality, take $H = [h_1, h_2, \cdots, h_m]$ to be a vector of ecologically valid measurements of human behavior relevant to the problem space. Assume a task for which a robot participant can independently calculate s, a scalar metric of success, which is a function of a vector of measurable environmental features $E = [e_1, e_2, \cdots, e_n]$. Thus, $s = f(E)$, where f is a task-specific function known to the robot. Using f and calculating s, a robot can build its own supervised training set for a learning task, where the human input

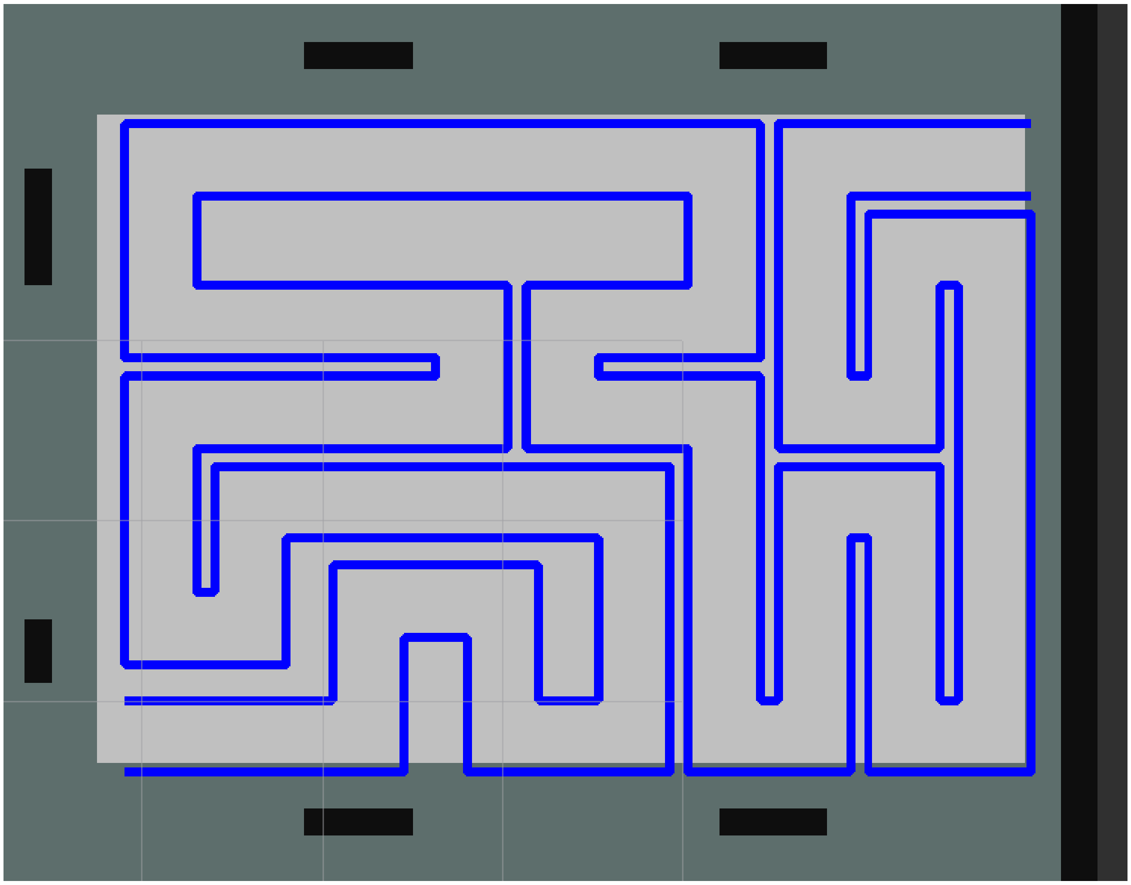


Figure 3.1: Representation of the path through the maze in RViz.

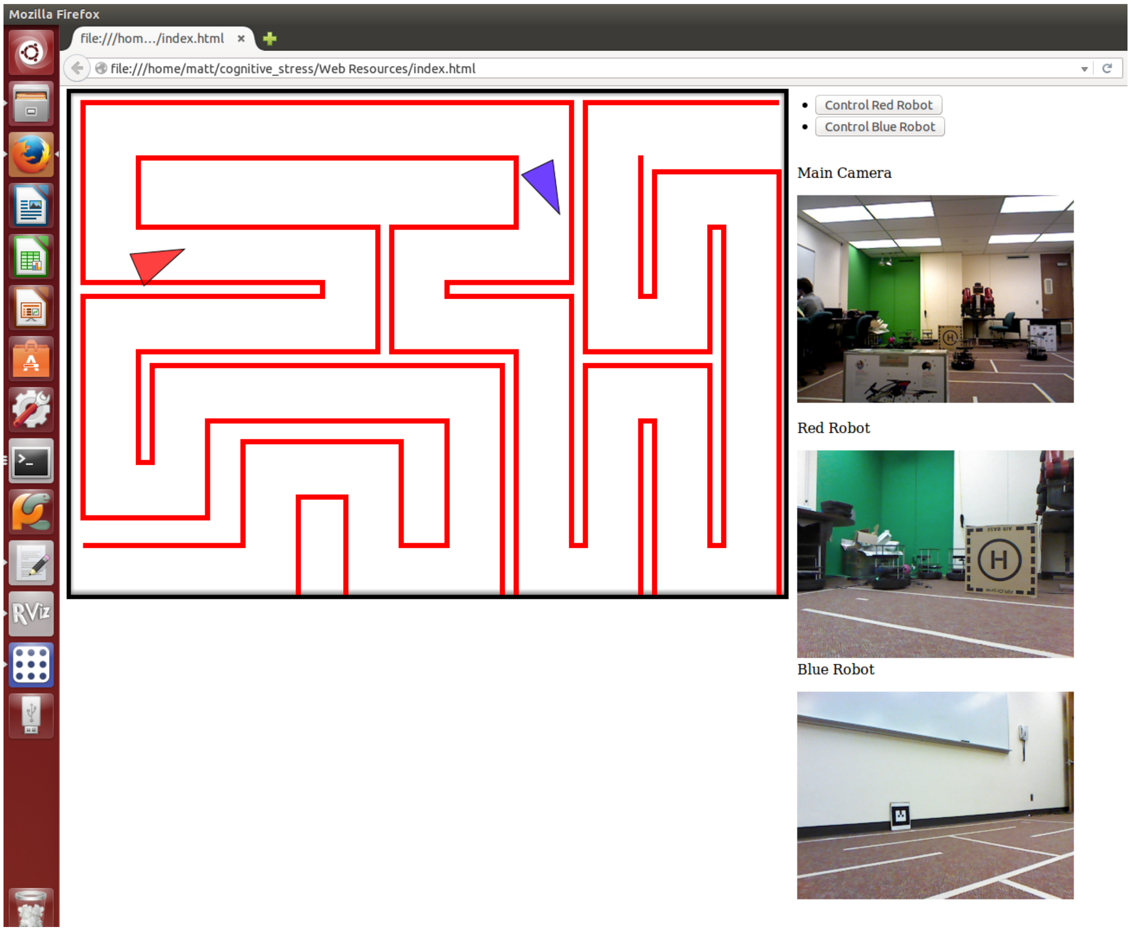

Mozilla Firefox
file:///hom.../index.html
file:///home/matt/cognitive_stress/Web Resources/index.html
Control Red Robot
Control Blue Robot
Main Camera
Red Robot
Blue Robot

Figure 3.2: The Rosbridge Web Interface.

H is associated with s through a learned function g. Thus, the robot learns to associate the human behavioral metrics H with task success s within a known task, so the output of g is a learned *estimate* of the true success ($\hat{s} = g(H)$). Now, assign the robot a task which requires human input for success, i.e., the robot has no access to an analogue to f or s in this new task. However, it can still measure the components of H, and it has access to its learned model g. We show that computing $\hat{s} = g(H)$ in this new environment allows the robot to estimate not the task success (about which it has no information), but the cognitive load on its human partner and an estimate of the quality of the human's direction.

3.2 Multi-Modal Multi sensor Interaction between Human and Heterogeneous Multi-Robot System

3.2.1 HRI Problem Description

I propose a new interaction method between humans and a heterogeneous multi-robot team using my algorithm described in Section **??**. I wish to project the joint belief space of the robots (constructed using my distributed algorithm) onto a phone's touchscreen to visualize the current beliefs of the robots and let the human interact with them in a meaningful way. This simple idea can produce an elegant solution for human interaction with a heterogeneous multi-robot team. Although the robots we are dealing with are heterogeneous (that is, they have differing sensing and actuation capabilities), we are interacting with abstract joint beliefs in intention space. This allows interaction with robots without worrying much about reprogramming the whole system to accommodate heterogeneity. The algorithm is distributed and theoretically can be scaled up to the maximum number of robots that the communication configuration can support. As mentioned earlier, research showed that touchscreen-based interaction performs better than gesture interaction. Nevertheless, all the positive aspects of gesture-based and many other multimodal sensory interaction methods can be easily integrated into the smartphone. Moreover, smartphone touchscreens are extremely accessible to today's human users.

Here I will lay out my initial work on the human interaction application. The current belief of the robot is constructed from the robot's sensory inputs and the propagated messages received from neighboring robots, and it interprets these messages based on its intention.

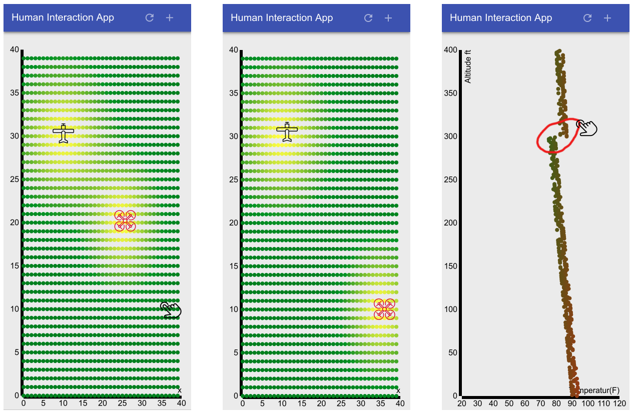


Figure 3.3: (a,b) A toy example where intention over the space is created as a Gaussian distribution and published as a ROS message. Human touch input directs the quadrotor away from the fixed wing aircraft. (c) A meterologist can easily determine an important temperature signature from the vertical profile provided by a quadrotor. She can mark the area of interest by sketching on intention space.

The current belief is streamed as a ROS[51] topic and served using ROS Bridge[52] to a platform-independent Progressive Web Appjoint intentional belief is then rendered using D3 JavaScriptprogrammable customized visualizations and allows our approach to be more flexible in terms of designing user interactions. A snapshot of the toy demonstration of the application is shown in Fig. 3.3.

3.2.2 Research Methods

So far I have implemented my belief propagation algorithm's basic functionality and experimented on simulated and real robots. Right now I am working on incorporating real fixed wing aircraft to coordinate with multiple quadrotors in the same airspace, collecting weather data and avoiding collision. I have started working on multimodal human interaction using a smart phone web application.

Experiment 1 can be described using Figure 3.4 and 3.5. Simulated UAVs A, B, and C here have an exploration task while avoiding collision and remaining within a constrained volumetric boundary – in this case, an 80m cube at 5m resolution, though much larger scales are algorithmically tractable. The scatter plot shows the distribution of intentions of the robots over the exploration space. In this experiment three ϕ factor functions are used for each robot, i.e. $\phi_{unexplored}$, $\phi_{boundary}$ and $\phi_{avoid\_collision}$. $\phi_{unexplored}$ is a delta function applied

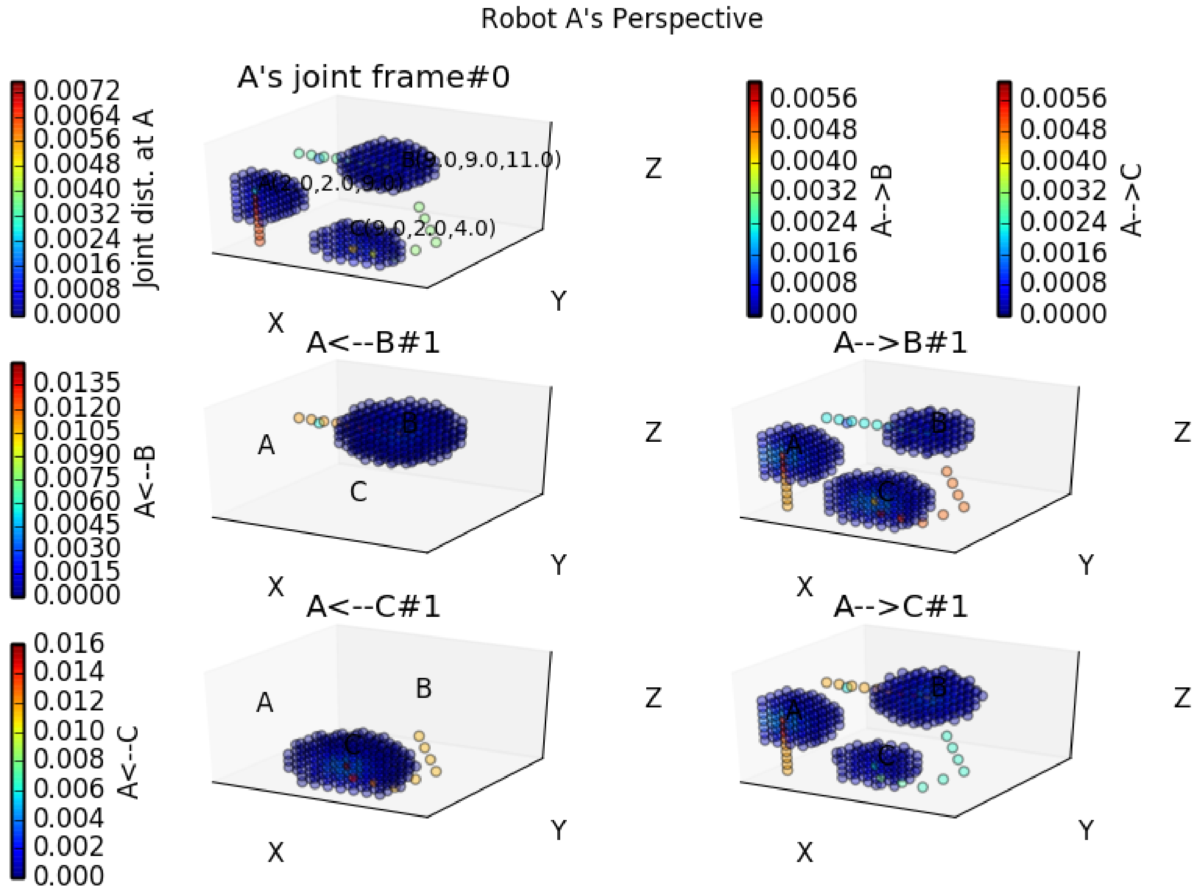


Figure 3.4: Joint belief or navigation intention of simulated UAS over 3D space. Heat map representation of probability distribution of intention. Color bar denotes the probability mass of navigation intention over the space. $A \Rightarrow B$ denotes the intention message passed from robot A to B. Space with higher probability mass value increases likelihood of robot choosing to visit.

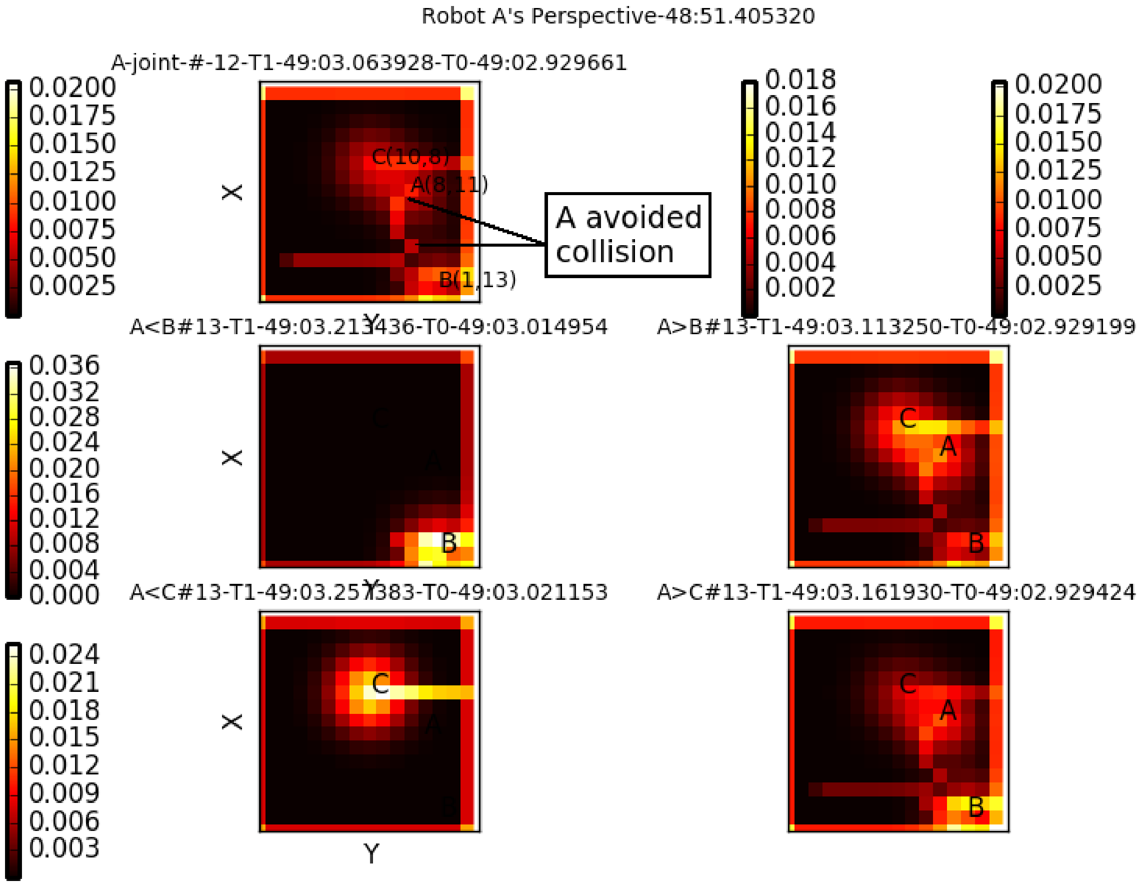


Figure 3.5: Joint belief or navigation intention of simulated UAS over 2D space. Heat map representation of probability distribution of intention. Color bar denotes the probability mass of navigation intention over the space. $A \Rightarrow B$ denotes the intention message passed from robot A to B. Space with higher probability mass value increases likelihood of robot choosing to visit.

to a space whenever a robot visits, making it less interesting. $\phi_{boundary}$ is a function which has a high value at the boundary of the space and zeros everywhere else. $\phi_{avoid\_collision}$ is a Gaussian distribution with mean at a robot's current pose. A robot takes a normalized sum of these functions derived from both its own sensors and the messages from its neighbors to build its intention over the space. The subplot in the bottom left of Figure 3.4 shows the message received by A from C about C's intention. For a exploration task C has a 3D Gaussian distribution with a mean at its current sensor pose estimate. This distribution has been coded in the factor node ϕ_c of the robot. Similarly the message received from B has a distribution having a mean at B's estimated current position in subplot in the middle on the left. The subplots on the right of these two subplots shows that when A sends its belief incorporating its neighbors' (B and C) intentions using the functions defined in factor nodes ϕ_{ab} and ϕ_{ac}. The subplot on the top left shows A's joint distribution of intentions incorporating its neighboring robots' messages. Figure 3.5 shows a similar process at a fixed altitude, i.e. in 2D space. It more clearly shows robot A's intention to avoid collision with robots B and C. My simulated experiment showed that UAS can explore the space without colliding with each other. I have also conducted the experiment with real robots using quadrotor UAVs (see Figure 3.6).

3.3 Distributed Control of Heterogeneous Team of robots and Humans

3.3.1 Problem Formulation

Consider robots A, B, C in an autonomous multi-robot team. Figure **??** shows an example of a factor graph. In this graph A, B, C and H represent four variable nodes of the factor graph. Among the variable nodes A, B, C represent the intentions of the respective robots, while H represents intentions provided by a human operator. A robot builds its own belief from its sets of sensors. For example, robot A builds its belief about the world using its set of sensors μ_a and actuators α_a and so on. These sets are not necessarily the same for all the robots in team, and thus the model naturally includes heterogeneous robots. Using μ_a, μ_b, μ_c robots A, B, C can build their own beliefs about the state of the world and can form their own intentions. This can be done using factor functions, i.e. ϕ_a, ϕ_b, ϕ_c respectively. In Figure **??**, all the robots are connected to each other in the sense that they can communicate

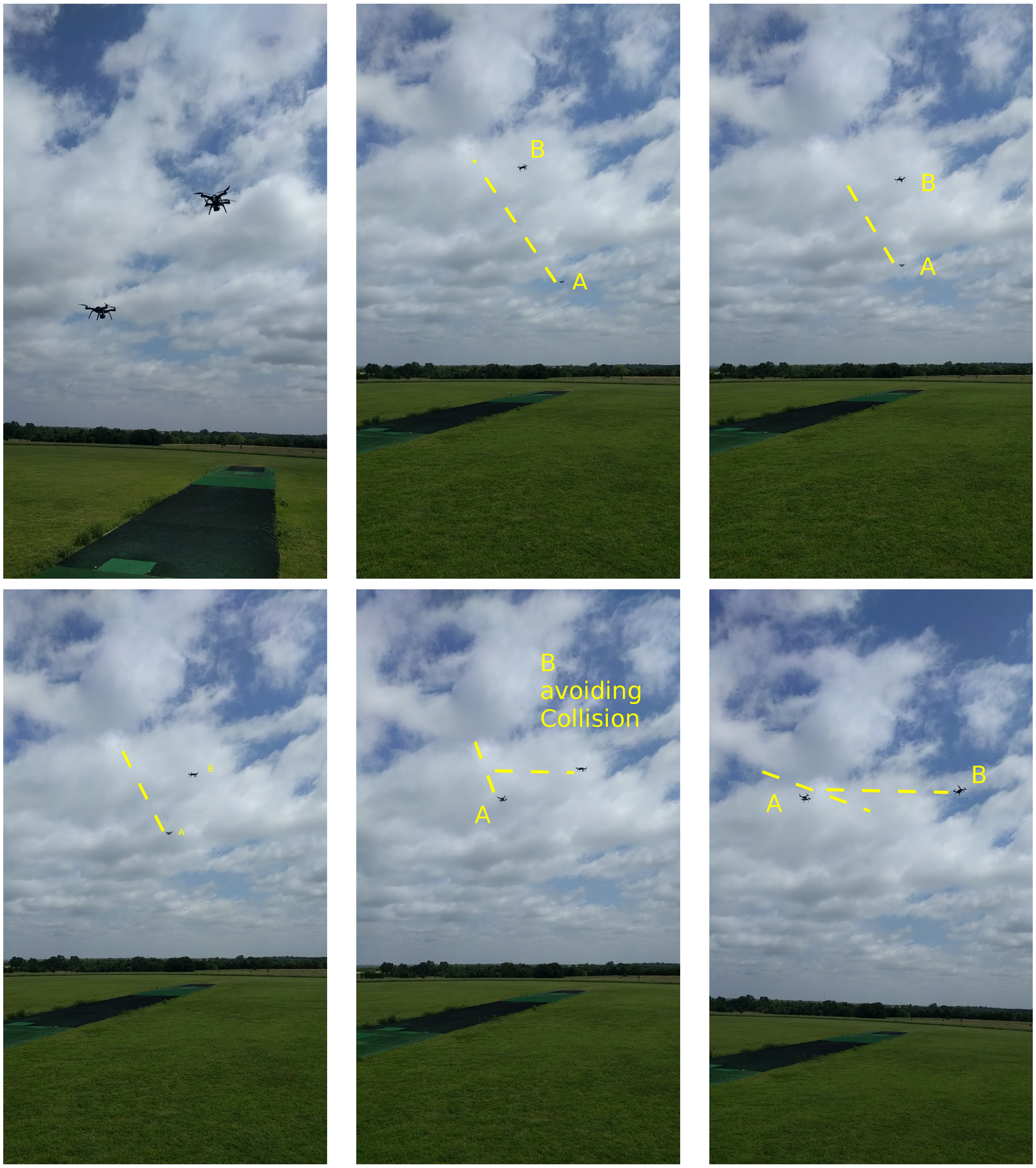


Figure 3.6: Experiment with two UAV robots A and B. (a) A started moving towards B. (b) B started moving away from A's path. (c) B is avoiding collision with A.

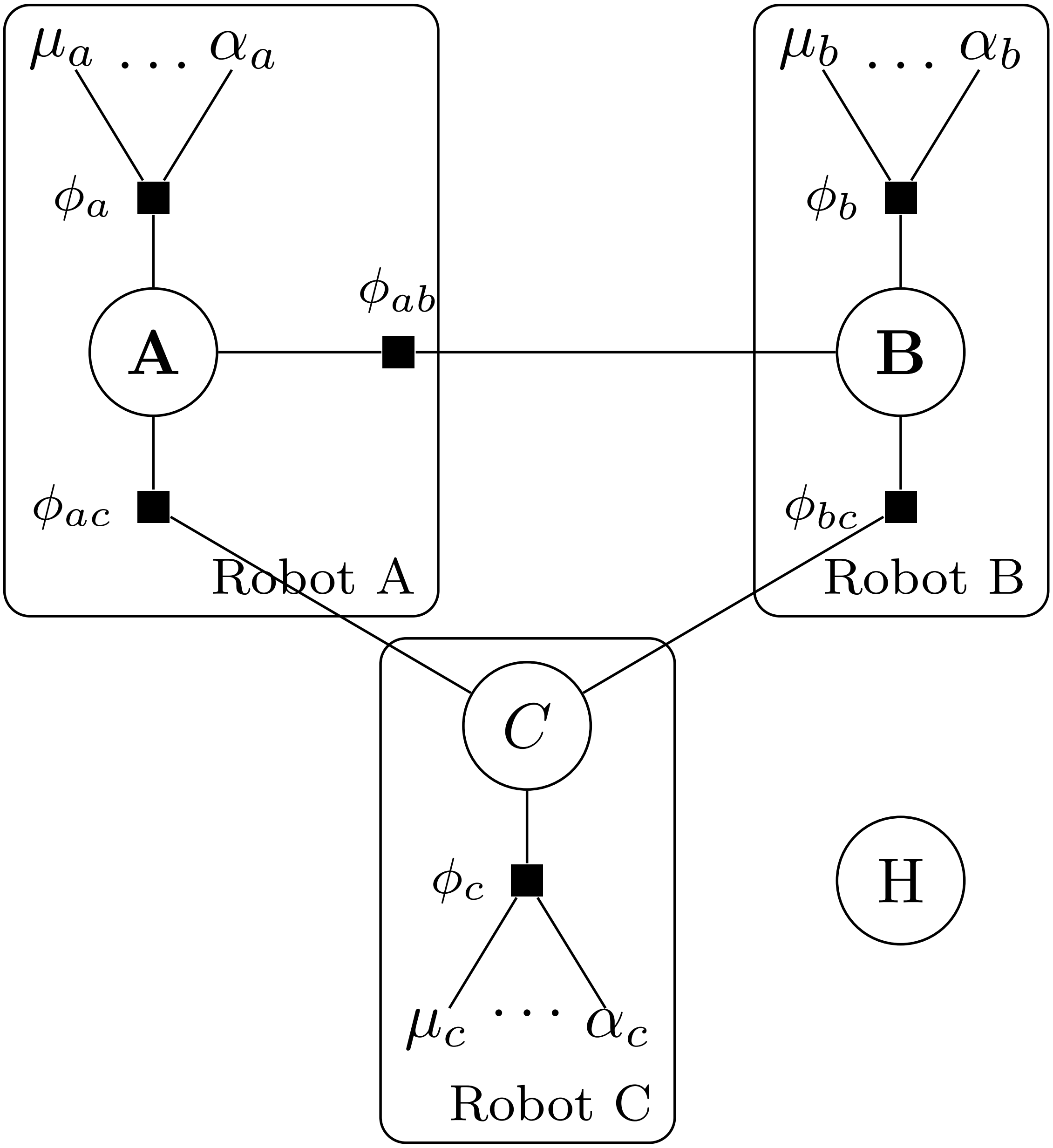


Figure 3.7: A factor graph model of our proposed algorithm with three robots and one human operator. Robots A, B and C compute intentional representations (in circle nodes), maintain sensor and actuator observations (μ and α), and apply ϕ functions to messages passed around the network. A human participant is labeled H. (a) A factor graph where all the robots are connected to each other. No human intentions currently being provided. (b) A is connected to B and C. No connection between B and C. The human operator can communicate additional intentions to B, which are shared throughout the robot team via loopy propagation.

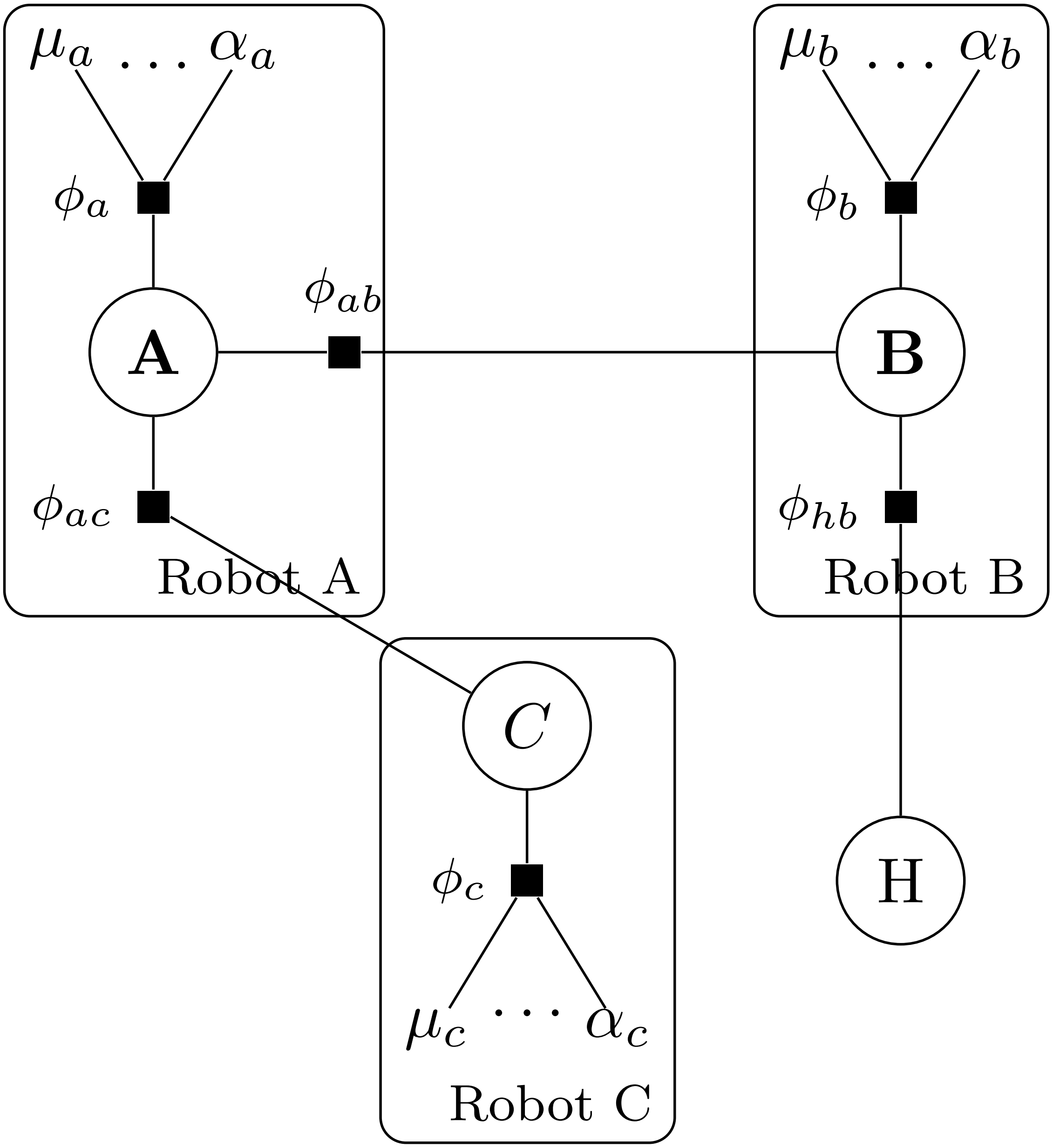


Figure 3.8: A factor graph model of our proposed algorithm with three robots and one human operator. Robots A, B and C compute intentional representations (in circle nodes), maintain sensor and actuator observations (μ and α), and apply ϕ functions to messages passed around the network. A human participant is labeled H. (a) A factor graph where all the robots are connected to each other. No human intentions currently being provided. (b) A is connected to B and C. No connection between B and C. The human operator can communicate additional intentions to B, which are shared throughout the robot team via loopy propagation.

with each other and incorporate each others' intention representation in calculating a joint distribution of belief.

For example, the intention of of robot B is incorporated into A's belief using factor function node ϕ_{ab}. In a joint exploration task for robots A, B and C, the collective objective might be to explore the entire space as quickly as possible while maintaining a collision-avoidance distance from each other. In such a scenario ϕ_{ab}, ϕ_{bc}, ϕ_{ca} can be designed as functions that make a space which has been visited by a particular robot less interesting for the other robots. For example, such a function could compute a time-decaying penalty associated with the robots' various reported location observations, while collision avoidance could be represented as a much stronger penalty function computed from a robot's current position and velocity. Similarly, other types of goals can be achieved using differently-devised ϕ functions. We will demonstrate this in our experiments. Robots can communicate their intention by passing messages to other neighboring robots, and they, in turn, pass that information along to other robots in the team.

In general, the joint belief g can be calculated using Equation III.1,

$$\begin{aligned} g(x_1, \ldots, x_n) &= \prod_{j \in J} f_j(X_j) \\ &= \frac{1}{Z} \prod_{ij} \phi_{ij}(x_i, x_j) \prod_{ij} \phi_i(x_i) \end{aligned} \tag{III.1}$$

where f is a generic function of the set of all the variables $x \in X$. By definition, a factor graph[53] is a bipartite graph of variables and factor functions. The computation of Eq. III.1 can be performed using a loopy belief propagation (LBP)[54] message passing algorithm on a factor graph. Here $\phi_i(x_i)$ denotes robot i's belief about variable x_i in the world from its sensory information. $\phi_{ij}(x_i, x_j)$ is the belief robot i forms from information received from robot j. All the ϕ functions are the factor functions in the factor graph. The loopy belief propagation algorithm on a factor graph is shown as Algorithm 1.

$$\mu^{a_i \to a_j}(x_j) = \frac{1}{Z} \sum_{x'_i : x'_v = x_v} \phi(x'_i) \prod_{V \in N_i \backslash a_v} \mu^{a_v \to a_i}(x'_v) \tag{III.2}$$

Equation III.2 defines the message μ passed from a variable to a factor, which consists of the normalized product of all of the messages received from the variable's neighboring

Algorithm 1: Loopy Belief Propagation **LBP**

```
Function Main($i$):
    /* at $i^{th} robot$                                        */
    repeat
        $AsynchronousUpdate()$
        $\pi_i^t = \nabla g(x_1, \ldots, x_n)$ /* t signifies time   */
        $ExecutePolicy(\pi)$
    until

Function AsynchronousUpdate($i$, $j$):
    Update belief using Eq.III.2;
    Update intention using Eq.III.3 and broadcast
return
```

factors, except for the recipient factor. Set N_v denotes the neighboring participating robots in the team for a robot a_v.

$$\mu^{a_i}(x_i) = \frac{1}{Z}\phi_i(x_i) \prod_{k \in N_i} \mu^{k \to a_i}(x_i) \tag{III.3}$$

Equation III.3 shows the message μ passed from a factor to a variable, which is the factor function applied to the messages from all other connected variable nodes, marginalized over all of the variables except the recipient's. These messages are passed asynchronously through links that are formed and dropped as the topology of the robot deployment changes, using loopy belief propagation. At a certain time t a particular robot a_i can run an optimization algorithm locally on the joint distribution of intention using its policy π_i.

$$\pi_i^t = \nabla g(x_1, \ldots, x_n) \tag{III.4}$$

Low-level controllers such as PIDs can achieve the goal produced by Eq. III.4. We used a gradient descent algorithm to calculate π_i^t. Other methods can be used as well.

A human operator can be imagined as another factor variable node in the overall graph, although they are not responsible for performing any computation (note that the ϕ_{ha} function applied to the human input in Figure **??** rests within the computational responsibilities of Robot B). Incorporation of human operators in the distributed factor graph coordination is one of the major contributions of this paper. Our approach allows a human

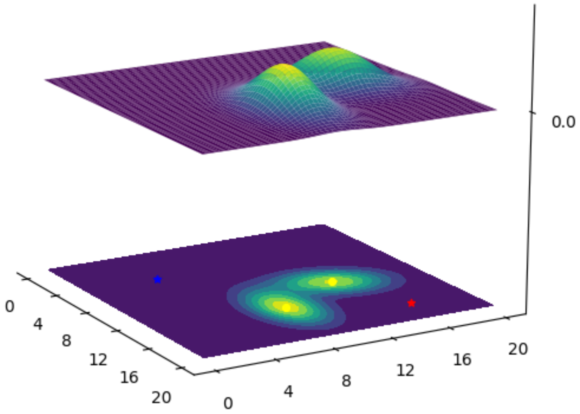


Figure 3.9: (a) Two robots' observations regarding the same obstacle are represented as Gaussian distributions over navigation space

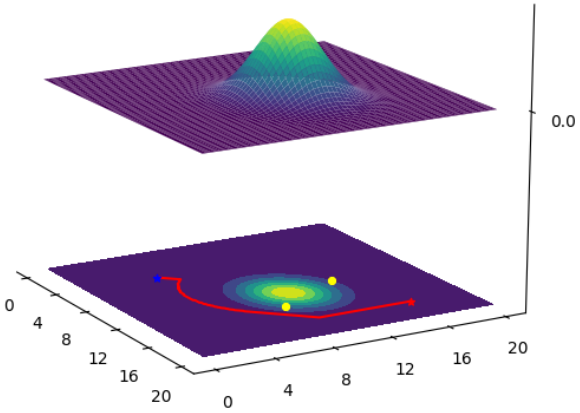


Figure 3.10: (b) After integrating the beliefs according to its ϕ function, one robot (blue star) executes a policy which leads to a goal (red star).

operator to exert an arbitrary amount of control over all of the agents that are indirectly or directly connected to the operator. If no human input is available (for example, if the human operator is task-saturated or does not have a connection to the agent), then the agent and the entire system function autonomously according to the robots' own sensor data and communicated beliefs.

When more guidance is available, the system will continue to propagate messages in identical fashion, but human input will be seamlessly integrated into that process in the form of one or more additional intentions mediated by an additional factor function. Furthermore, if the operator only has an indirect connection to an agent through other agents, these imposed beliefs will still reach the appropriate agent through loopy belief propagation within the whole network. The fusion of the distributed communications architecture, the belief-based information processing, and the optional human interaction allows us to create a general purpose heterogeneous architecture that accommodates smooth changes in robots, hardware, and human input.

One of the known limitations of loopy propagation within factor graphs is that the beliefs occasionally fail to converge in certain cases. This problem is rare, and we went to considerable effort to evaluate its effect on our particular application. We simulated twenty autonomous UAS systems communicating their beliefs about their and each others' observed positions, with many different belief parameterizations. Each robot communicated with all the others, forming the loopiest possible clique. Even so, in hundreds of trials, we were never able to induce the graph to fail to converge. Convergence was always achieved within eighty message iterations.

In our real-world experiments, we have so far involved only as many as five robots, which makes the problem even less likely. In addition, the robots are functioning in real time, with changing conditions, positions and measurements, so if they were to find themselves in a rare non-convergent state, they would quickly emerge from it before it had a chance to produce a substantial performance or safety impact. We can calculate a theoretical upper bound for the probability that the propagation algorithm produces a problematic result. If we assume that nonconvergence induces the absolute worst possible policy selection (itself hugely unlikely), then a collision could happen with probability $p(NC)^n$, where $p(NC)$ is the probability of the network entering a non-convergent state and n is the number of time steps it takes to steer straight for an obstacle (as opposed to away, as the actual computed policy would indicate). Likewise, for a survey problem, the worst policy choice would mean moving in the least interesting direction instead of the most, which could increase the time taken by $2np(NC)$, where in this case n is the number of time steps to conduct the survey. Since our simulations indicate that $p(NC)$ is a number extremely close to 0, these negative effects are negligible.

Chapter IV

Experiments and Results

4.1 Assessing Cognitive Stress of Human Operator

In keeping with the problem statement, experiments were carried out in two parts: the Maze Game and the Coin Game experiments. In the first phase, the robot collected the data needed to build a model g for evaluating the trustworthiness of user input H. The robot is able to do this for the first phase because in the case of the Maze Game it has access to f and E and can calculate s. It understands the problem sufficiently to make such judgements; the robot requires no human aid to solve this problem.

In the subsequent Coin Game, the robot is placed in a different scenario, one in which it had no access to success measures or even rules; beyond the fact that it was moving in a similar environment to the Maze Game, it is wholly reliant on human direction to succeed in the task. Even so, with no independent means of measuring task success, it can still calculate $\hat{s} = g(H)$, and can therefore evaluate the quality of instruction, and hence the cognitive capacity, of its human partner.

Communication between operators and robots was achieved using the Robot Operating System (ROS)[51]. Users sent navigation goals to the robots by using a web interface developed for the project, which allowed the experiment to be conducted remotely (Fig 4.2).

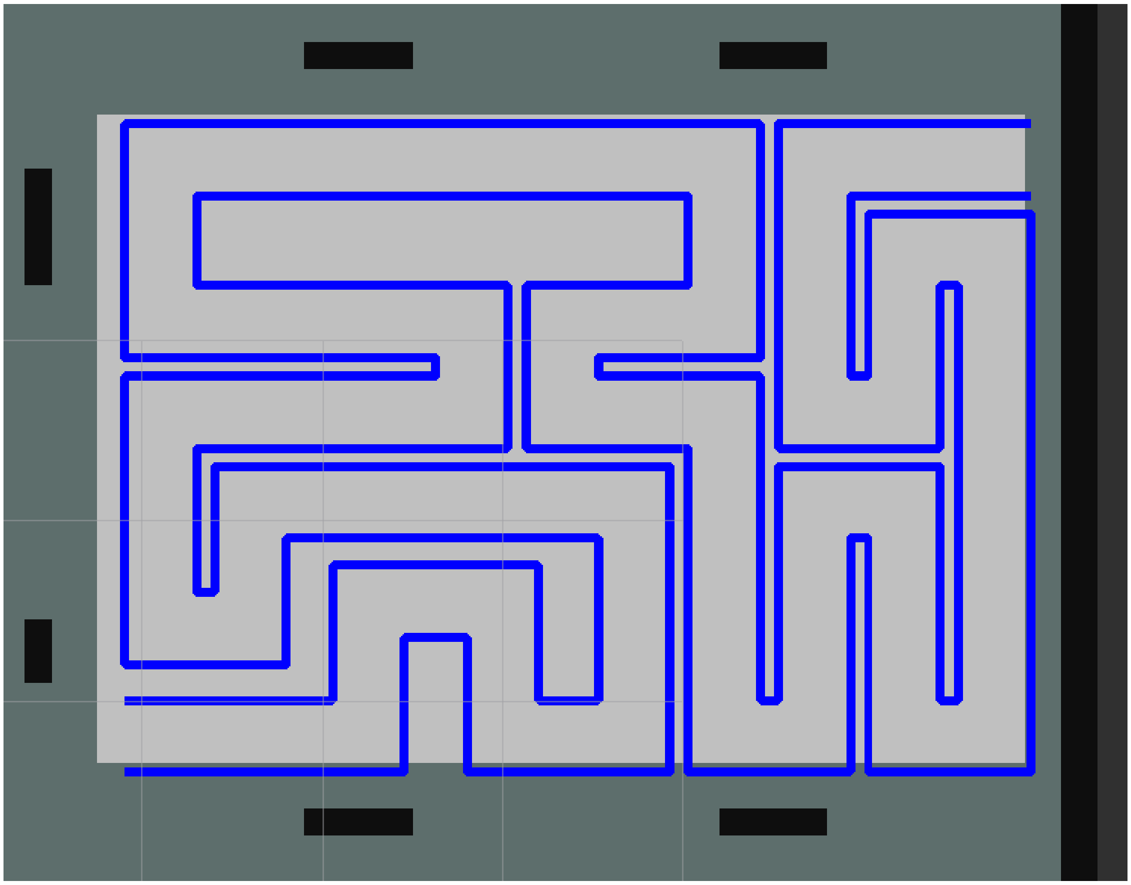


Figure 4.1: Representation of the path through the maze in RViz.

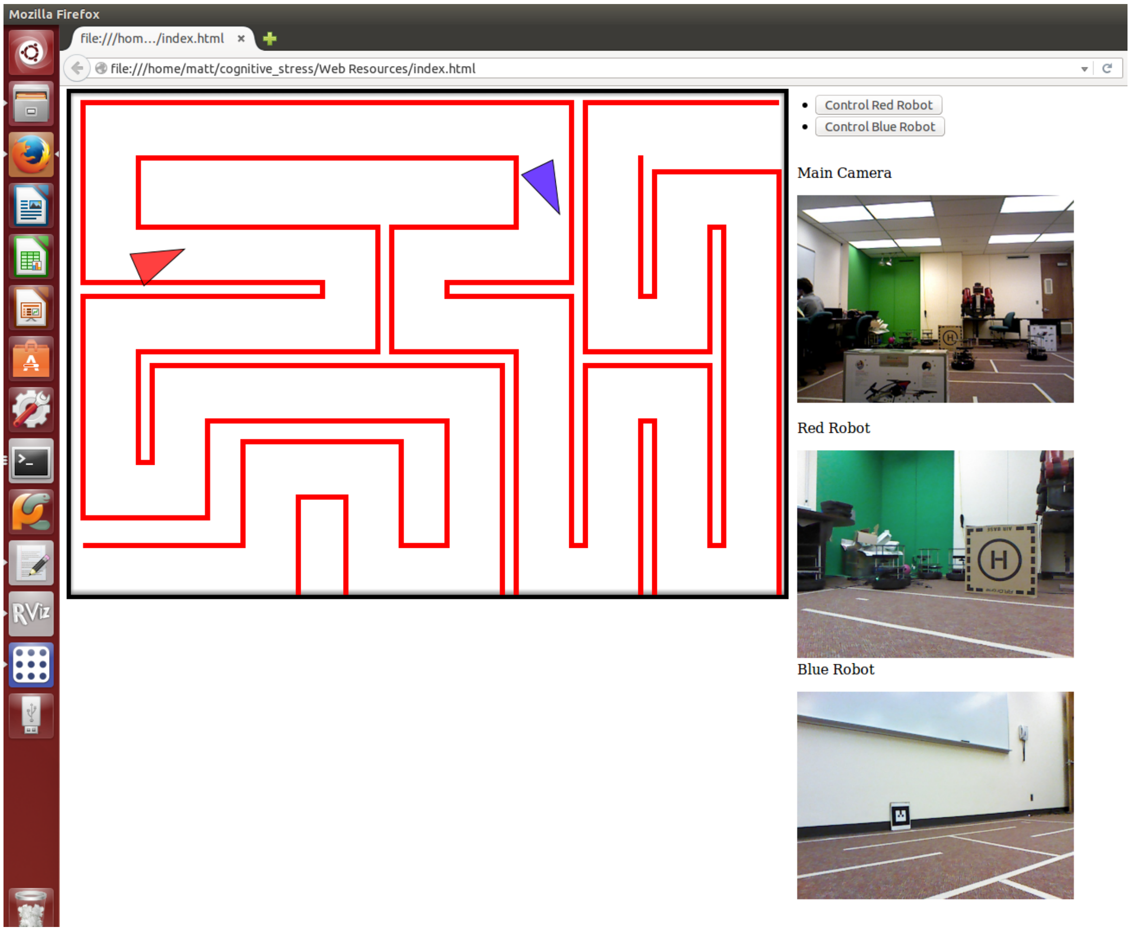

Mozilla Firefox
file:///hom.../index.html
file:///home/matt/cognitive_stress/Web Resources/index.html
Control Red Robot
Control Blue Robot
Main Camera
Red Robot
Blue Robot

Figure 4.2: The Rosbridge Web Interface.

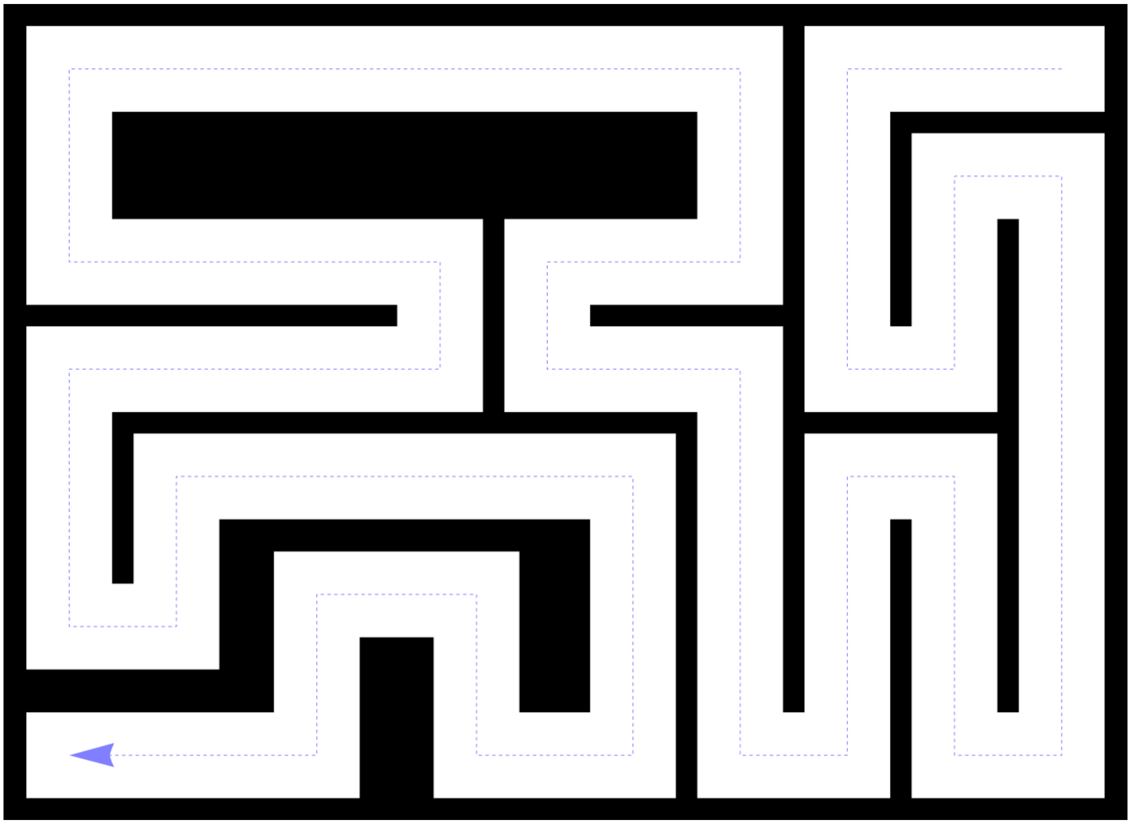


Figure 4.3: Map of the maze. Arrow indicates path direction.

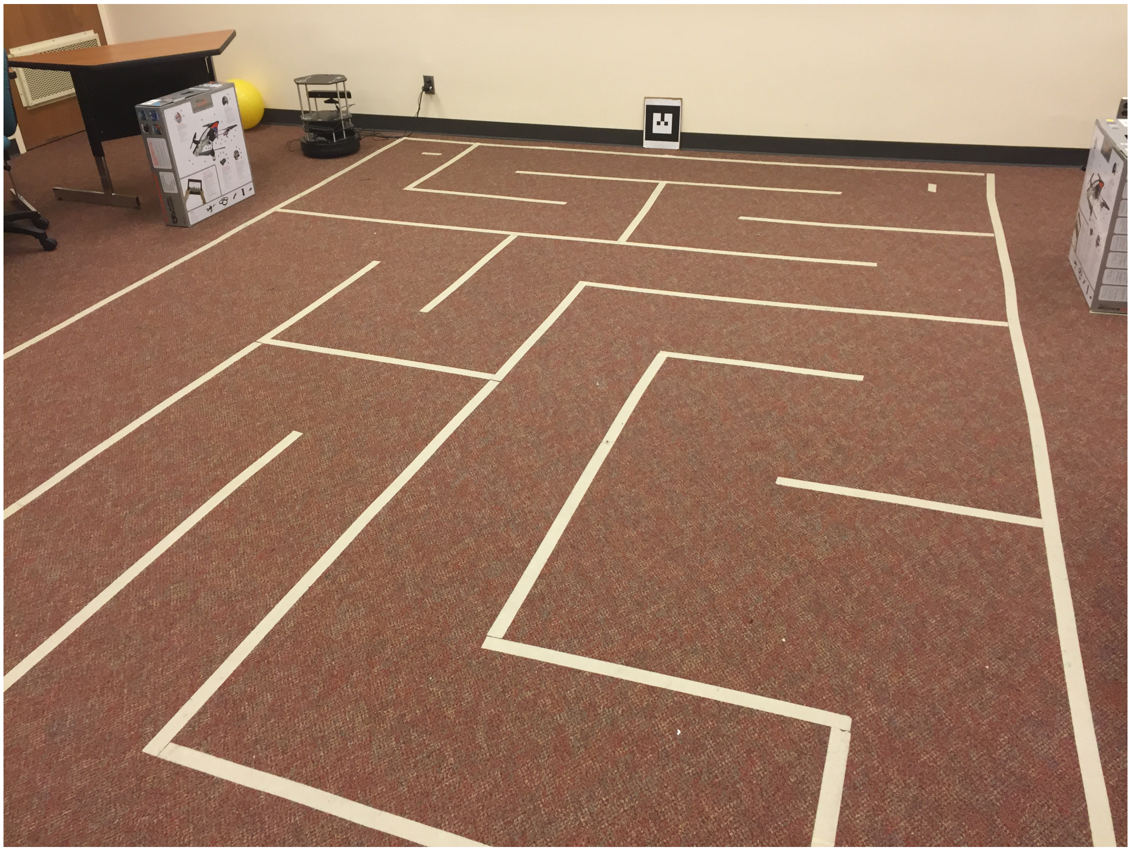


Figure 4.4: The final map laid out on the lab floor.

4.1.1 Maze Game and Training

As previously mentioned, the first step of our experimental design was to identify a task simple enough for the robot to complete unaided, yet complex enough to potentially benefit from human input. The problem also need to be readily scalable to include additional robots. This led us to select autonomous navigation through a maze using Adaptive Monte Carlo Localization (AMCL). The maze consisted of a single path with multiple 90° turns. The final design, seen in Fig 4.4, consists of 32 waypoints and measures $4.63m$ by $3.2m$. It is important to note that the maze traced out on the floor serves only as a convenience for human operators and does not serve as a navigation guide for the robot. In fact, the robot does not treat the walls of the maze as physical obstacles, and will happily pass through them if instructed to do so.

In the Maze Game, the robot is able to evaluate the success of the directions it is given by its human operator. This is a function of a vector of environmental measurements $E = [e_0, e_1, e_2]$, which in this particular context have been defined as follows:

- e_0 is the *disparity* term, the distance between the navigation directions provided by a human and the route that the robot would have planned for itself,
- e_1 is the *collision* term, which penalizes collisions with walls, and
- e_2 is the *time delay* term, the amount of time taken for the human to guide the robot through the maze, compared with the robot's estimate of the time it would have taken under its own power.

The computation of $s = f(E)$ to obtain a success metric is straightforward:

$$s = \frac{1}{Z} \sum_{i=0}^{|E|} e_i \tag{IV.1}$$

where Z is a normalization constant.

We then define a decision function δ based on s. This is intended to indicate whether the provided human assistance is qualitatively above or below that person's useful cognitive threshold θ:

$$\delta(i) = \begin{cases} True, & \text{if } s \leq \theta \\ False, & Otherwise \end{cases}$$

By computing this value s, the robot can label its own data in order to train a supervised learning algorithm which will relate the success of a human-directed task (and, presumably, the cognitive capacity of the human partner) with a set of measured behaviors H.

$H = [h_0, h_1, h_2]$ is a set of human behavioral metrics which are ecologically valid for a navigation direction task. For this particular experiment, these are the following:

- h_0 is the *decision interval* term, which measures the time elapsed between the robot reaching a navigation goal and the human providing a new one,
- h_1 is the *error correction* term, which measures the tendency of a human operator to provide a navigation goal and then subsequently provide another before the task is complete, and
- h_2 is the *franticness* term, which measures the pace of control inputs generally.

These features were chosen because they seem reasonable and ecologically valid in a navigation context, but there is no reason to assume that they are uniquely suitable as components of H. Appropriate features for inclusion in H should be easy to identify in a wide variety of tasks.

4.1.2 Maze Game and Training Results

Now we will show that the robot is able to learn the model $\hat{s} = g(H)$, and that it reflects the cognitive capacity of humans correctly, based on the described metrics. Fifteen test subjects performed runs of the maze game, resulting in approximately 1000 data points used for training.

Fig. 4.5 shows a four-dimensional scatter plot that shows the robot's judgment of human cognitive capacity respect to the humans' behavioral metrics. Because the robot understands the maze game and the details of the performance metrics, it is able to generate the training data labels for a supervised model learning task. We have used the Orange

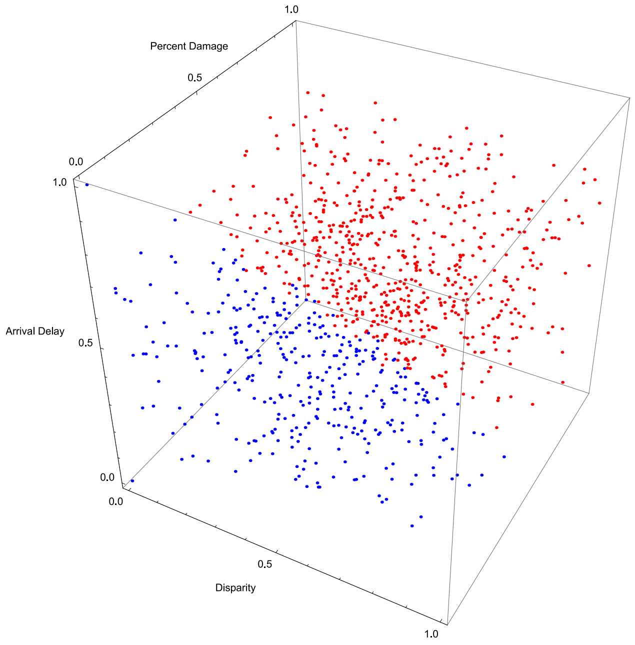


Figure 4.5: Disparity, Collision, Time Delay vs. Task Success. The robot has adjudicated red dots as indicating human behavior at or beyond a useful cognitive threshold, and blue dots as trustworthy.

data mining API[1] for running the regression. The human behavioral metrics H and the robot's self-generated task success metric s are used to train a Support Vector Machine (SVM) regression. SVMs are able to capture regressions and classifications even in high-dimensional spaces. SVMs have been used to learn enormous classifiers in very complex feature spaces including cancer cell detection and spam email classification [55, 56]. The maze game regression operates only in a low-dimensional feature space, so an SVM is able to learn from our experimental data effortlessly, and the approach should scale to much more complex feature spaces. The classifier is trained on H and s. The SVM generates an optimized $g(H)$ by performing a regression on the relationship between s and H. identifies a regression on the test data by minimizing the following function:

$$\tfrac{1}{2}w^T w + C \textstyle\sum_{i=1}^{N} \xi_i$$

with the following constraints,

$$y_i = (w^T x_i + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, i = 1, ..., N$$

[1] http://orange.biolab.si

Table 4.1: Accuracy of the classifier in the maze game experiment.

Training Size	Accuracy of classification
45%	95%
50%	95%
55%	95%

Table 4.2: Confusion matrix of the classification using cross validation method.

		Prediction	
		True	False
Actual	True	72	10
	False	10	72

Here C and b are constants. ξ denotes the non-separability of the input data.

The classifier was trained using two different methods; cross-validation and leave-one-out. Both of the data sampling methods provided good classification accuracy. In the cross-validation method we sampled 70% of the data from the maze game to use as training data, which resulted in a 95% accuracy in the classification on the test data sample. In the leave-one-out method, all instances except one are selected for training the classifier. After that, the test data set is classified using the learned model. All of the instances are selected at least once as test data. The total accuracy of the classification is calculated by counting the percentage of data that has been selected and classified correctly against the total data set. This method also resulted in 95% of the data being correctly classified. The result of the classifier on the test data using different sampling size using cross validation method in the first experiment is shown in Table **??**

4.1.3 Coin Game Experiment

The second phase of the project involves a variation on the first experiment dubbed the Coin Game. It uses many of the same principles introduced in the Maze Game, such as the same map, so that the human behavior metric H remains ecologically valid. However,

Table 4.3: Confusion matrix of the classification using leave one out method.

		Prediction	
		True	False
Actual	True	72	10
	False	10	72

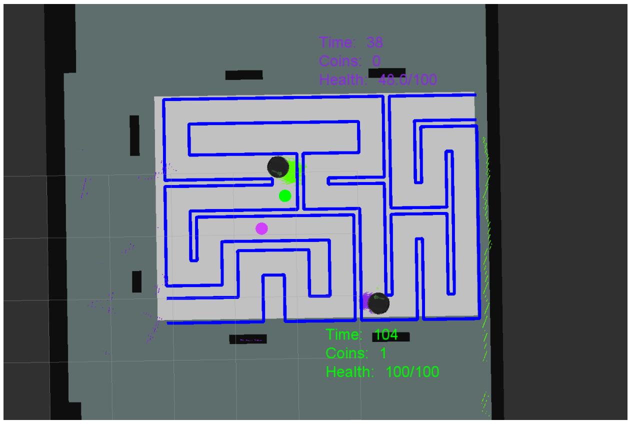


Figure 4.6: Maze map of the Coin Game experiment with coin goals.

in this task, only the human operator has access to the sensory data required to successfully perform the task. Goals appear appear randomly in the maze, and we call these random goals “coins”, with the understanding that the point of the game is to send the robot to the coin locations in order to collect them. Unlike in the previous experiment, where the robot navigated the maze in linear fashion from start to finish, the goals or coins may pop up in any location in the maze, either ahead of or behind the robot. The goal is not represented in the real world; rather the goals are shown on the control interface’s computer screen as colored circles denoting the coins. Coins can pop up between any two key points. The interface for this experiment is shown in Fig.4.6.

The coin game is a very simple game. The human operator points the mouse to a location the map. As the goal is random, the human operator must now switch between robots frequently, in order to marshal the various robots toward their coin goals. Research [14] has shown that this switching time can affect the effectiveness of a task performed by a human-robot team. Thus the second experiment is more challenging in terms of cognitive stress. The task sets a two-minute timer for the human operator and also sets a maximum 5 coin collection limitation. Either condition satisfies the end of the game. Every time the robot collects a coin the timer is shortened by 12 seconds to put more cognitive stress on human operator.

In this instance, the robot has access only to what it can measure about human behavior, in the form of the vector H. It knows nothing about what constitutes success in the Coin Game, as it is not even able to sense the existence of coins. However, it can still compute $\hat{s} = g(H)$, and it can thus calculate an estimate of human cognitive capacity and reliability.

In order to compare the robot's estimate of cognitive capacity with reality, test subjects were asked during the experiment to rate their own cognitive stress levels at 12-second intervals using a Likert scale.

We have set two scenarios in this experiment. The first scenario, is the single human operator one or more robots team. In this scenario only one human operator was allowed to assist the human-robot team. We have set a timer for the human operator as 2 minutes and also set maximum 5 coin collection limit. Either condition satisfies the end of the game. Every time the robot collects a coin the timer is shortened by 12 seconds to put more cognitive stress on human operator. In the second scenario, multiple one or more robots were teamed up with two human operators. The initial timer was set to 1 minute and a maximum of ten coins were allowed to collect. Either condition satisfied he end of the game.

4.1.4 Coin Game Results

Analysis of the results from the coin game experiment are in agreement with the original hypothesis. In general, the robot correctly predicts the cognitive load that its operator was under in every scenario. These findings can be seen in Fig 4.7. Here, the x axis denotes an increase in task complexity over time; the user must issue an increasing number of commands and collect coins in an increasingly short duration. It is critical for interpreting the information in 4.7 to note that while the robot's cognitive load evaluation and the scoring technique for quantifying self-reported user stress both produce a result between zero and one *their magnitudes are not directly comparable.* The points at which stress was identified, and the shape of the curves, however, are appropriate to compare. Whenever cognitive stress occurs or changes, the robot is able to recognize this increase for most cases in the tested scenarios, and the robot's evaluation agrees with self-reported user stress. In the context of this paper, it also supports the original hypothesis: robots can are able to reliably assess the cognitive strain their human partners are under, even in contexts where the actual tasks they are being asked to perform are opaque to the robot.

Six people who participated in first experiment also participated in the second experiment. We have found that the learned model of classifier was able to understand good

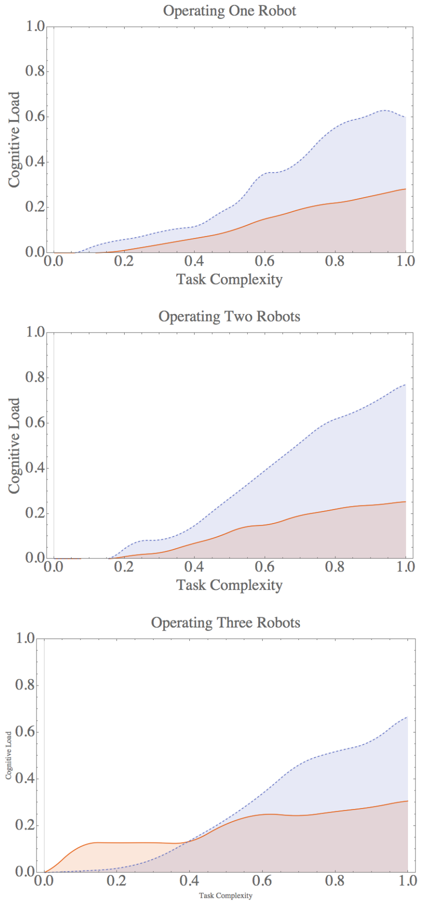


Figure 4.7: Cognitive load of human operators in Coin Game experiments with differing numbers of robots and steadily increasing task complexity. Blue represents subject self-reporting, orange the robot's estimate of $\hat{s}$.

and bad assistance very efficiently. We did not allow the robot to change its behavior based on the trustworthiness; it is just keeping a log of which of the assistance came from the human operator were trustworthy and which were not. While running the second experiment we manually kept a log to indicate which particular human assistance were seemed to be trustworthy in the naked eye. After the experiment we collected the log from the robot and then compared with our human judged interpretation of trustworthiness. We have found that the 95% and 97% of robots prediction matches with the two human interpretations. This is a very promising result to support our hypothesis in Eq. **??**. The result also showed decrease in trustworthiness as the complexity of the task increases. The result on matching classification on different level of complexities are shown in the following Fig4.7:

4.1.5 Cognitive load assessment

In general, the robot correctly predicts an operator's cognitive load. Figure 4.8 shows modest correlation between physiological evidence (breathing rate measured with a Bioharness) of an operator and the robot's estimation of stress. This is suggestive but not conclusive; it may be that physiological stress measures are not precisely indicative of the cognitive load which our robots attempt to predict.

Much more convincing is the learned model's contribution to task success. The coin game requires the operator to navigate the maze collecting coins (visible to the human operator but not to the robot). Delays and errors in successfully collecting coins increase an operator's task penalty score; as time pressure and the number of robots participating in the game grows, the operator's cognitive load is likewise expected to increase. In the manual test condition, the robots continue to act according to human instruction regardless of their model's estimate of cognitive load, while in the autonomous assistance mode, the robots revert to maze navigation behaviors whenever their learned human behavior models detect high cognitive stress. As shown in Figure 4.9, this behavior significantly enhances the overall performance in the game. Robots are able to reliably assess the cognitive load their human partners are under, even in contexts where the actual tasks they are being asked to perform are opaque to the robot.

Fig 4.10. From our experiments we have found that the cognitive load estimate from the robots using our model correlates with the coder evaluated stress level by a factor $\rho = .205$ and $\rho = .25$ for coin game played with one and two robots respectively. Our experiment also captures that posture and breathing rate of human operator as objective metric for estimating cognitive stress level.We could not find any interesting pattern in other physiological objective metrics i.e. heart rate, ECG amplitude using our model and experimental setup.The benefits of this can be seen in Fig 4.9. It is critical for interpreting the information in Fig. 4.10 or 4.8 to note that while the robot's cognitive load evaluation and the scoring technique for quantifying self-reported user stress both produce a result between zero and one *their magnitudes are not directly comparable.*Whenever cognitive stress occurs or changes, the robot is able to recognize this increase for most cases in the tested scenarios, and the robot's evaluation agrees with self-reported user stress.

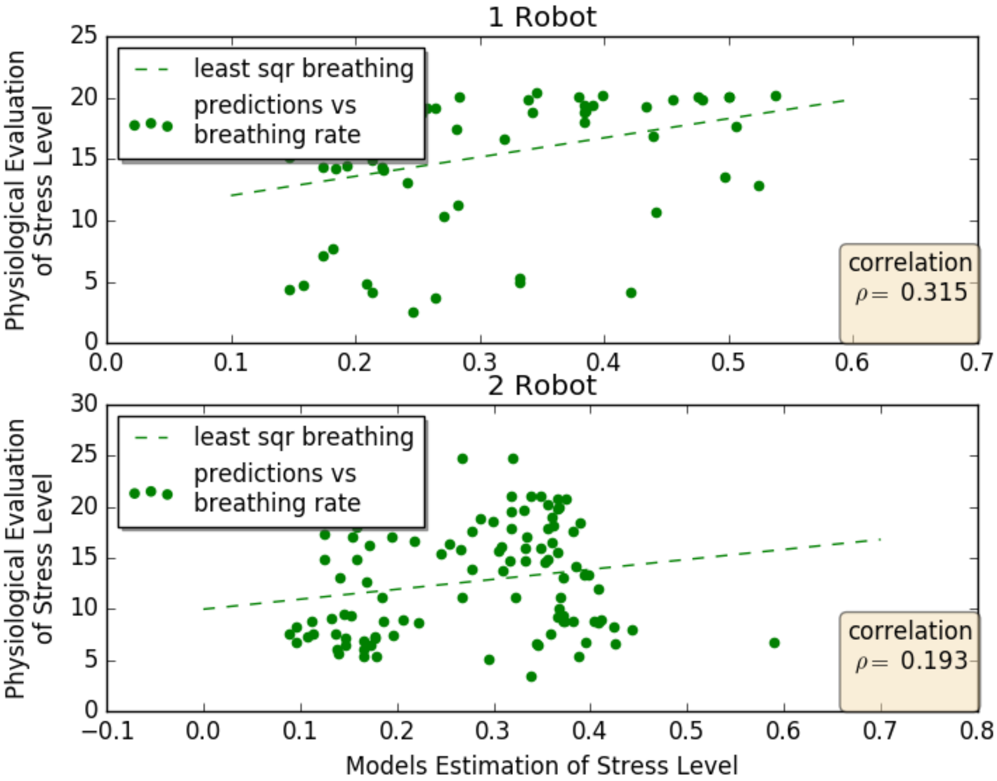


Figure 4.8: Robot's estimated cognitive stress level modestly correlates with physiological metrics.

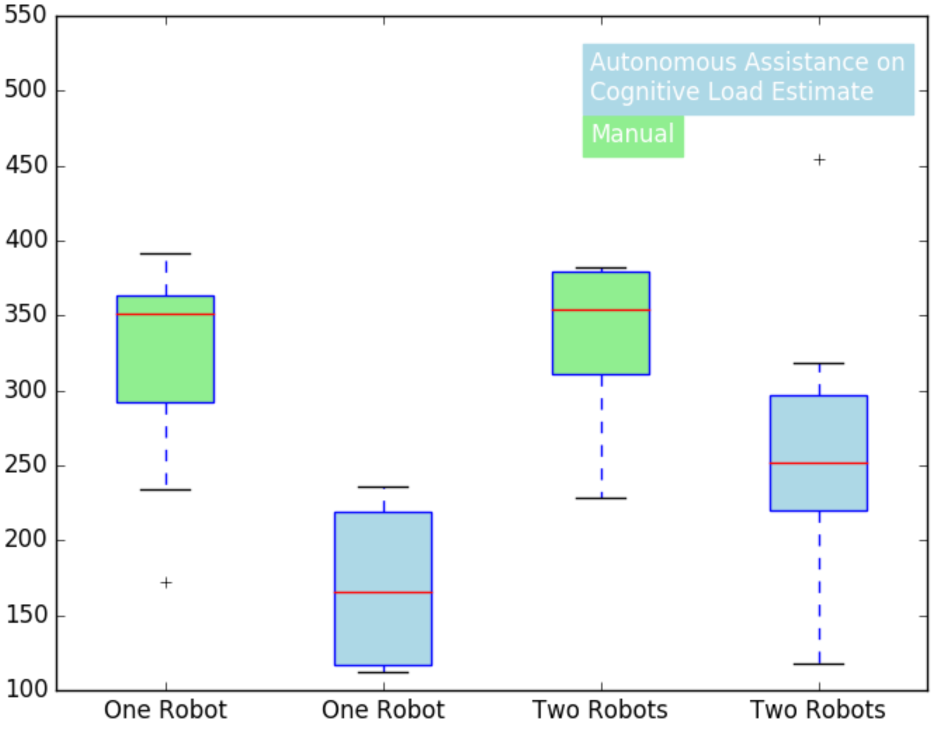


Figure 4.9: Coin game task penalties in manual vs. autonomous assistance modes across 34 test subjects. $p < 0.05$ in both instances.

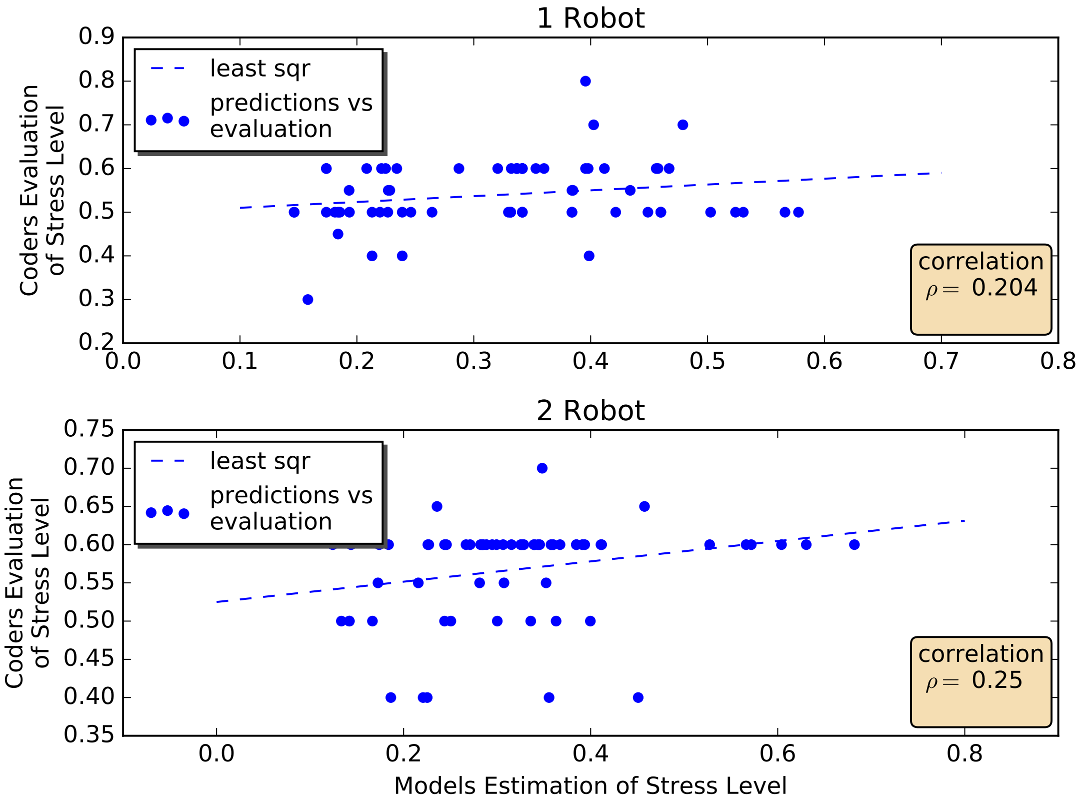


Figure 4.10: Cognitive load of human operators in Coin Game experiments with differing numbers of robots and steadily increasing task complexity.

Table 4.4: Task success comparisons between autonomous assistance mood and manual mood.

Game Mode	Manual		Autonomous Assistance on high Stress level	
Number of Robots Participated	one	two	one	two
Number of Failure/ Total Number of Game Played	3/7	3/7	0/10	1/10
Task Success Score per game	318	355	170	258.2

4.2 Multi-Modal Multi sensor Interaction between Human and Heterogeneous Multi-Robot System

4.3 Distributed Control of Heterogeneous Team of robots and Humans

4.3.1 Flightgear Simulation

We have conducted several experiments using simulated and real robots. In this section we describe three tasks involving exploration of the spatial environment, starting with simple intentions such as exploration and collision avoidance, and demonstrate how additional complex intentions can be introduced easily using our approach.

4.3.2 Exploration Within Boundary Avoiding Collision

In the first experiment we implement the basic functionality of our proposed algorithm for a team of simulated heterogeneous UAVs, conceived as a collection of ROS[51] nodes which pass factor graph messages between each other over ROS topics. It has been implemented in a space discretized system. The messages passed from one robot to other include the UAV's current position, a history of positions it has already visited, and sensory information such as temperature, humidity and air pressure. For this first experiment, the robots do not make decisions based on the sensory information, but they will do so later in the paper. The messages passed by each robot were also timestamped, which can be

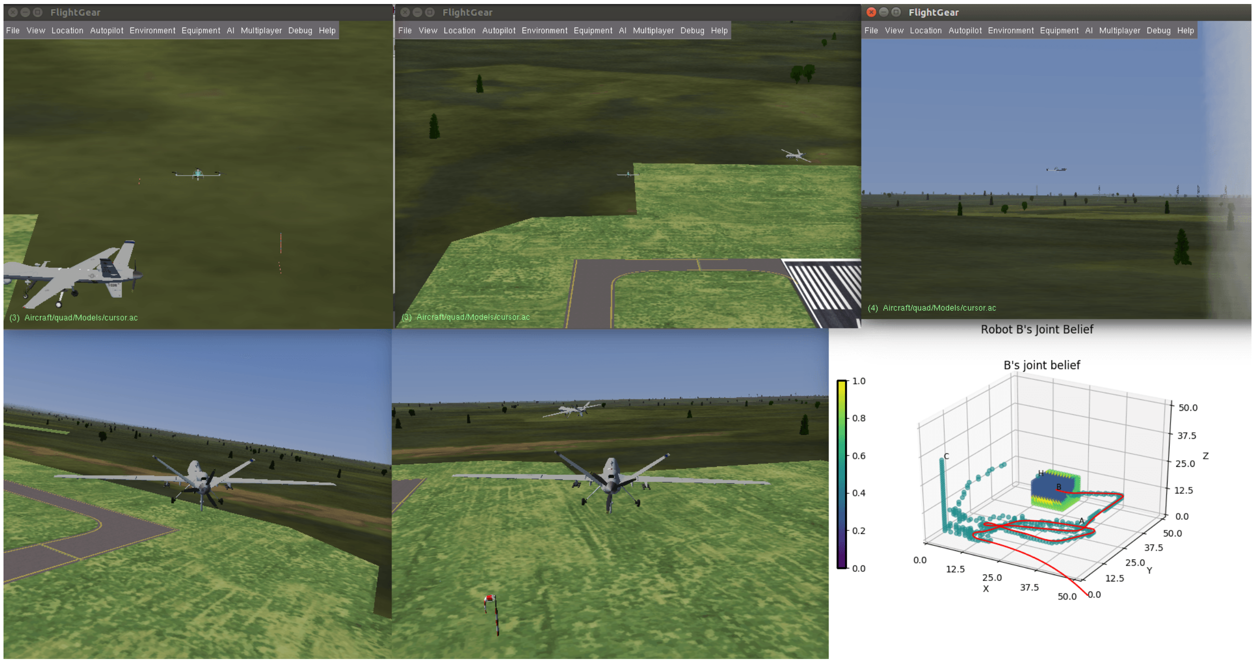

FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
(3) Aircraft/quad/Models/cursor.ac
FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
(3) Aircraft/quad/Models/cursor.ac
FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
(4) Aircraft/quad/Models/cursor.ac
Robot B's Joint Belief
B's joint belief
1.0
0.8
0.6
0.4
0.2
0.0
C
B
A
X
0.0
12.5
25.0
37.5
50.0
Y
0.0
12.5
25.0
37.5
50.0
Z
0.0
12.5
25.0
37.5
50.0

Figure 4.11: Snapshot of heterogeneous team exploration using Flightgear.Quadrotors and fixed-wing aircraft coordinate together to explore, and are each captured in different picture elements. The lower-right element is the joint belief of B. (a) UAS B is moving towards the CO_2 plume. UAS A still exploring far from the plume. (b) UAS B already passed through the plume and sent intention for UAS which are interested in exploring area with high CO_2 density. UAS A is such a robot so it moves towards the plume. B's joint belief in lower-right sub figure also shows the trail of A's path.

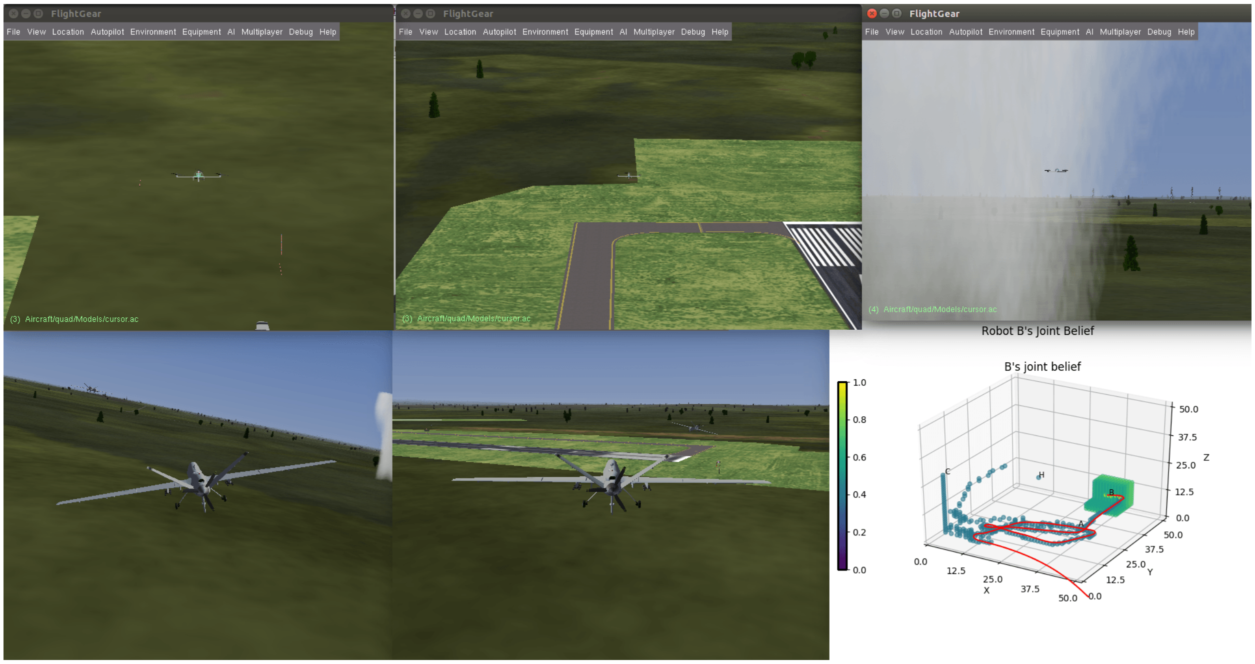

FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
(3) Aircraft/quad/Models/cursor.ac
FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
(3) Aircraft/quad/Models/cursor.ac
FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
(4) Aircraft/quad/Models/cursor.ac
Robot B's Joint Belief
B's joint belief
1.0
0.8
0.6
0.4
0.2
0.0
C
H
B
A
0.0
12.5
25.0
37.5
50.0
X
0.0
12.5
25.0
37.5
50.0
Y
0.0
12.5
25.0
37.5
50.0
Z

Figure 4.12: Snapshot of heterogeneous team exploration using Flightgear.Quadrotors and fixed-wing aircraft coordinate together to explore, and are each captured in different picture elements. The lower-right element is the joint belief of B. (a) UAS B is moving towards the CO_2 plume. UAS A still exploring far from the plume. (b) UAS B already passed through the plume and sent intention for UAS which are interested in exploring area with high CO_2 density. UAS A is such a robot so it moves towards the plume. B's joint belief in lower-right sub figure also shows the trail of A's path.

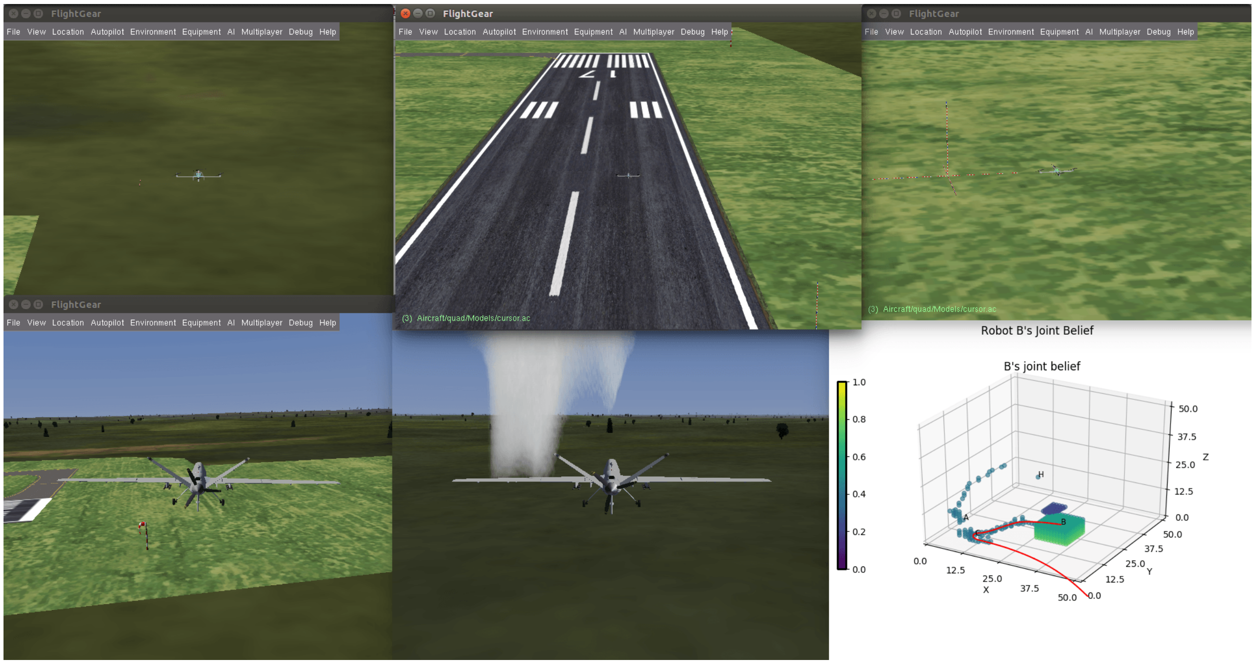

FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
(3) Aircraft/quad/Models/cursor.ac
Robot B's Joint Belief
B's joint belief
1.0
0.8
0.6
0.4
0.2
0.0
H
A
B
0.0
12.5
25.0
37.5
50.0
X
Y
Z

Figure 4.13: Snapshot of heterogeneous team exploration using Flightgear.Quadrotors and fixed-wing aircraft coordinate together to explore, and are each captured in different picture elements. The lower-right element is the joint belief of B. (a) UAS B is moving towards the CO_2 plume. UAS A still exploring far from the plume. (b) UAS B already passed through the plume and sent intention for UAS which are interested in exploring area with high CO_2 density. UAS A is such a robot so it moves towards the plume. B's joint belief in lower-right sub figure also shows the trail of A's path.

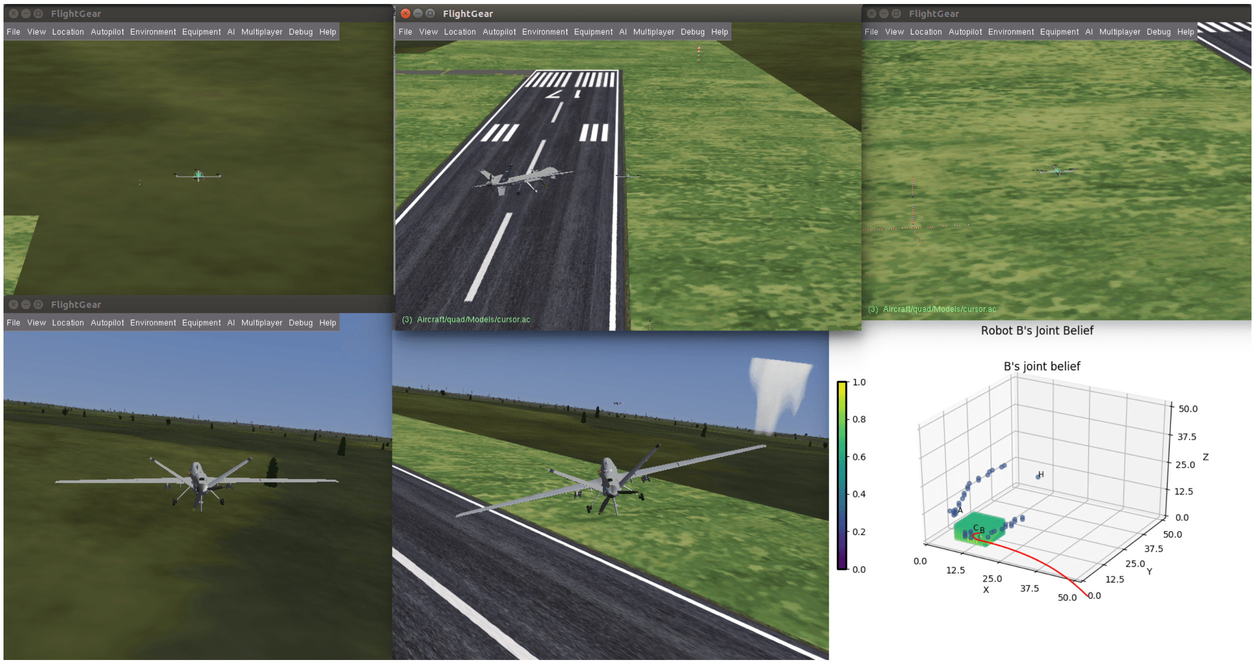

FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
(3) Aircraft/quad/Models/cursor.ac
FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
(3) Aircraft/quad/Models/cursor.ac
FlightGear
File View Location Autopilot Environment Equipment AI Multiplayer Debug Help
Robot B's Joint Belief
B's joint belief
1.0
0.8
0.6
0.4
0.2
0.0
A
C
B
H
0.0
12.5
25.0
X
37.5
50.0
0.0
12.5
25.0
Y
37.5
50.0
0.0
12.5
25.0
Z
37.5
50.0

Figure 4.14: Snapshot of heterogeneous team exploration using Flightgear.Quadrotors and fixed-wing aircraft coordinate together to explore, and are each captured in different picture elements. The lower-right element is the joint belief of B. (a) UAS B is moving towards the CO_2 plume. UAS A still exploring far from the plume. (b) UAS B already passed through the plume and sent intention for UAS which are interested in exploring area with high CO_2 density. UAS A is such a robot so it moves towards the plume. B's joint belief in lower-right sub figure also shows the trail of A's path.

used by ϕ functions which weight information by recency. The messages passed were lists of parameters packed in ROS's message format. For all the experiments, the messages included at least the robots' position, orientation, the identities of neighbors and a limited history of previously broadcast parameters. We have also run experiments using non-parametric methods where the intentions over the whole navigation space were shared as a probability mass function. However, we have not used non-parametric methods in the experiments discussed in this paper. In the experiments discussed later in this paper, the messages also included sensor readings like temperature, humidity, CO_2 density and so on. For fixed-wing aircraft, roll, pitch, and yaw angles along with GPS position and other aerodynamic parameters were passed, which were incorporated into an agent's belief calculations.

We have extended the open source flight simulator Flightgear[2] in order to develop a robotic simulation software suite which supports atmospheric physics phenomena such as turbulence, visibility, temperature, humidity, and the behavior of water vapor and gas plumes such as clouds, smoke, methane, and carbon dioxide. Figure 4.16 shows simulated UAVs which have an exploration task while avoiding collision and remaining within a constrained volumetric boundary – in this case, $500m^2$, between altitudes of 50 and 450 feet. Much larger scales are algorithmically tractable. The whole space was discretized into $50 \times 50 \times 50$ voxels.

In this experiment, three basic ϕ factor functions are used for each robot, i.e. ϕ_u (unexplored), ϕ_b (boundary) and ϕ_{ac} (avoid collision). ϕ_u is a local weight function applied to a space whenever a robot visits, making it less interesting. ϕ_b is a function which has high value near the boundary of the space and zeros everywhere else. ϕ_{ac} is a Gaussian distribution with mean at a robot's current position and a standard deviation of 3.33 and 5 voxel units for quadrotor and fixed wing UAV respectively. The negligible probability mass beyond 3σ is ignored. The parameters were chosen based on the various platforms' maximum airspeed and the size of the voxels, sufficient for belief updating at 10 Hz to allow the robots to take autonomous actions before a collision happens. A robot takes a normalized weighted sum of these functions derived from both its own sensors and the messages from its neighbors to build a joint intention over the space local to its current position. The weight $w_\phi \in (0, 1)$ of each intention ϕ depends on the priority of the task. For example,

[2] http://home.flightgear.org/

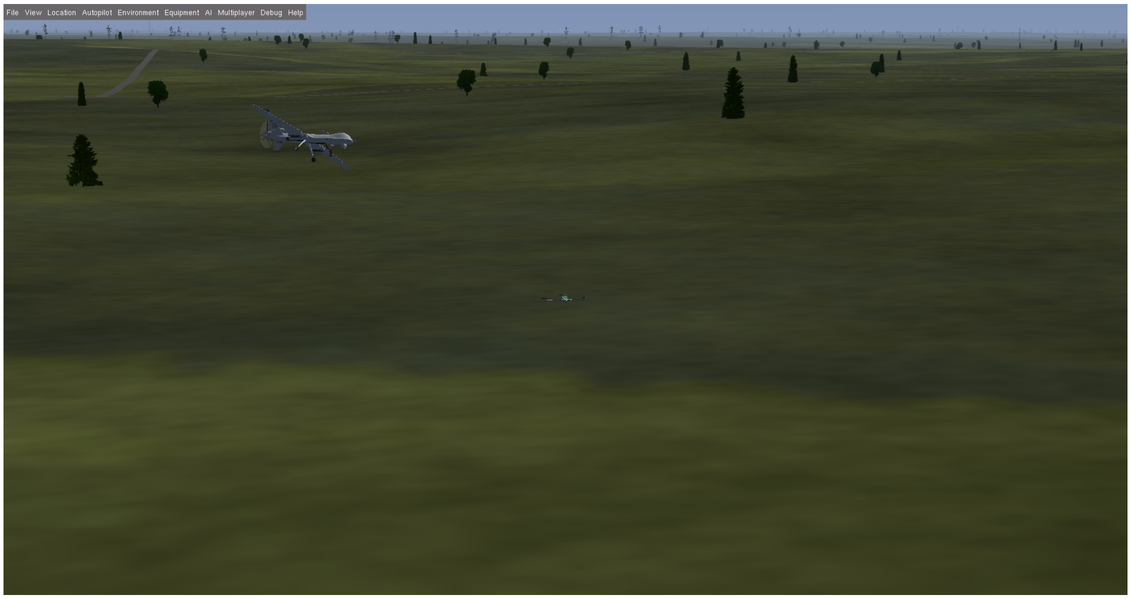

File View Location Autopilot Environment Equipment AI Multiplayer Debug Help

Figure 4.15: Heterogenous robots avoid each other while exploring the space. Simulated fixed-wing UAV (denoted B) avoiding collision with quadrotor (denoted C)

the unexplored function will have less probability mass in unvisited areas, attracting the robot to navigate there. However, if it is close to another robot's current position then then the probability mass from collision avoidance will outweigh the attraction because the ϕ functions associated with avoiding collisions and avoiding boundaries are accorded much higher weight than other behaviors not concerned with safety of flight. These weights also help humans to control the behavior of the the autonomous system. The robot descends the gradient of its joint belief. For all of our applications, boundary weight was set to 1. Weights for direct and indirect collision avoidance were set to 0.85 and .8 respectively. The direct collision avoidance parameter is used for a robot which is directly communicating its position to another robot, whereas the indirect parameter is used to weight the collision avoidance intention from other robots that are propagating their position through an intermediary node or robot.

In this experiment $\phi_{boundary}$ and $\phi_{avoid\_collision}$ were set to 1 to prioritize these intention over $\phi_{unexplored}$ which was set to .5. Intuitively, in this context the intention has been set to prefer avoiding collision with other robots and being within the defined boundary over exploration essentially telling it to explore unexplored area without collision and without leaving the boundary.

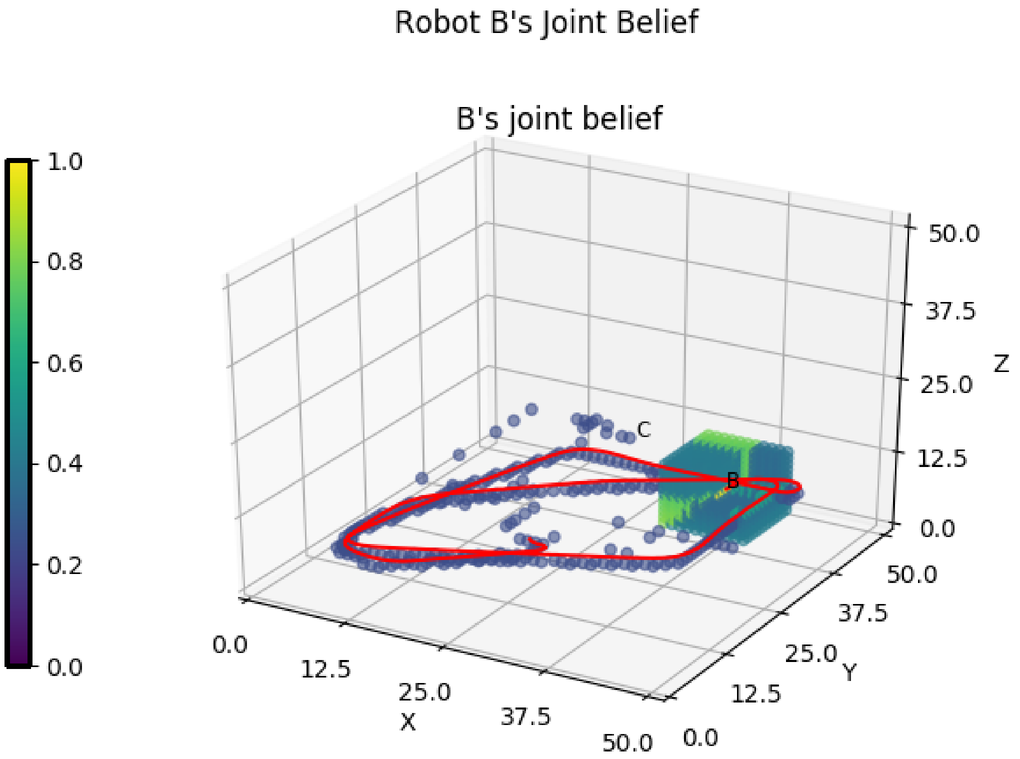


Figure 4.16: Heterogenous robots avoid each other while exploring the space. Joint belief of fixed-wing UAV B. Red line shows path B followed. B only calculates local gradient over the joint distribution of belief.

Fig. 4.16 shows the fixed-wing UAV, denoted as B, avoiding collision with the quadrotor, C, despite the quadrotor being placed directly in the path of the fixed-wing UAV. It also depicts the joint intention that B builds using messages it received from its neighbor C. The scatter plot shows the distribution of intentions of the robots over the exploration space. The color-coded voxels signify the probability of a voxel being less interesting or worth visiting. The ϕ functions and weights described for this experiment have very similar implications for the experiments we will discuss later.

Existing prior work tends to represent the intention plot using 3D surfaces to intuition of the distribution of belief over planar space which is easier to visualize. However, actually for navigation over 3D a scatter plot or point cloud practically useful representation where the color representing the probability mass in 3D space. Moreover, as we are not using any range sensor our sensory information is much coarse over the space thus a we are representing the joint distribution over the space using scatter plots.

4.3.3 Exploiting Heterogeneity

Using our factor graph distributed algorithm, we can exploit the heterogeneous configuration of our robot team. Teams of such robots can accomplish more complex tasks more quickly, and in distributed fashion. In this experiment, we demonstrate this using our simulator. The task is to locate, survey and map a CO_2 plume within a given area. Our heterogeneous team consists of two similar fixed wing UAVs and three quadrotors with slightly different sensory capabilities. All of these simulated UAVs are equipped with GPS, temperature and humidity sensors, but only the fixed-wing aircraft and one quadrotor are equipped with CO_2 sensors.

A fixed-wing aircraft is much faster than a quadrotor, but also far less maneuverable. A team of fixed-wing aircraft will quickly locate traces of CO_2, but they will not be able to carefully map its contours. On the other hand, while a maneuverable quadrotor is better equipped to perform the detailed survey, its slow speed makes the location of the plume difficult to find in the first place.

We have run the simulation in homogenous (all quadrotors) and heterogenous configurations several times with the CO_2 plume situated in different locations. Table 4.5 compares the time taken to find and map the plume.

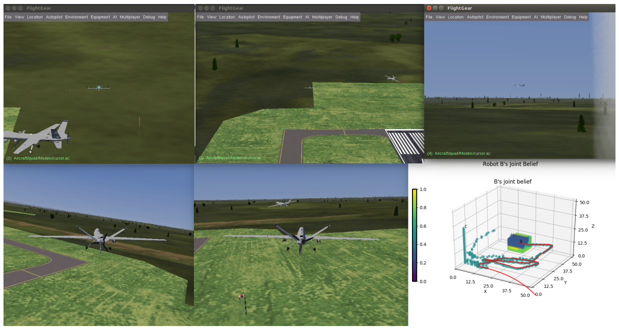


Figure 4.17: Snapshot of heterogeneous team exploration. For visibility we have hidden joint intention of all the robots except for B (a fixed-wing UAS). Quadrotors and fixed-wing aircraft coordinate together to explore, and are each captured in different picture elements. The lower-right element is the joint belief of B. (a) All UAS begin exploring the space. In lower-mid window UAS B and C are captured in the same frame.

Table 4.5: Heterogenous CO_2 plume mapping

Location of Plume			Number of UAVs	Mapping Time (approx)	
X	Y	Z		Homogeneous Team	Heterogeneous Team
40	30	18	5	>20 min	20 sec.
7	28	8	5	3 min	120 sec.
26	34	34	5	>20 min	75 sec

4.3.4 Heterogeneous Team of Robots

In this experiment we simulate robots with different sensor capabilities to demonstrate autonomous coordination of heterogeneous teams of robots. In the weather forecasting application, understanding temperature inversion layers, where the temperature changes drastically, is very important. We simulate temperature sensor data by assigning values throughout the experiment volume, derived from actual measurements taken with (non-autonomous) UAVs. The simulation environment is similar to the previous experiment, but in addition, robot A possesses a thermometer while B does not. Robot A therefore incorporates another $\phi_{temperature}$ when building its joint belief. Figure **??** shows that robot A remains in the vicinity of the sharp temperature gradient in order to map the meteorological phenomenon, while B explores the rest of the space ignorant of the temperature gradient but appropriately ceding that task to the appropriately-equipped robot. This cooperative, coordinated behavior was not coded explicitly. It is also worth mentioning that by modifying the ϕ function we can modify this particular behavior as well. Moreover, a human operator can also exert arbitrary intentions on the robot, as we will see in our next experiment.

4.3.5 Human Intentions and Heterogeneity

This experiment is conducted in similar fashion to the previous ones. However, we allow a human to exert arbitrary intentions during the exploration task. Robot A still possesses its temperature sensor. Robot B, on the other hand, possesses a communication channel to a human operator. The human could send two types of intentions to the robots, telling them to either to explore or avoid some particular area. Two functions $\phi_{interesting}$ and $\phi_{dangerous}$ have been designed as Gaussian distributions centered on particular coordinates. The $\phi_{interesting}$ intention operates at the same time as the previously-specified $\phi_{unexplored}$ function, and gradually exerts less significance as the area of interest is thoroughly explored. In this experiment, the human operator simply sends intentions by publishing coordinates on the command line, but in real world scenarios this would be accomplished by such interfaces as a remote controller, joystick or touchscreen map. Indeed, in our real-world UAV experiments, we use exactly these devices, but so far in these cases, we have not designed explicit human-coordination ϕ functions.

Figure **??** shows how the simulated robots build joint beliefs incorporating dynamic

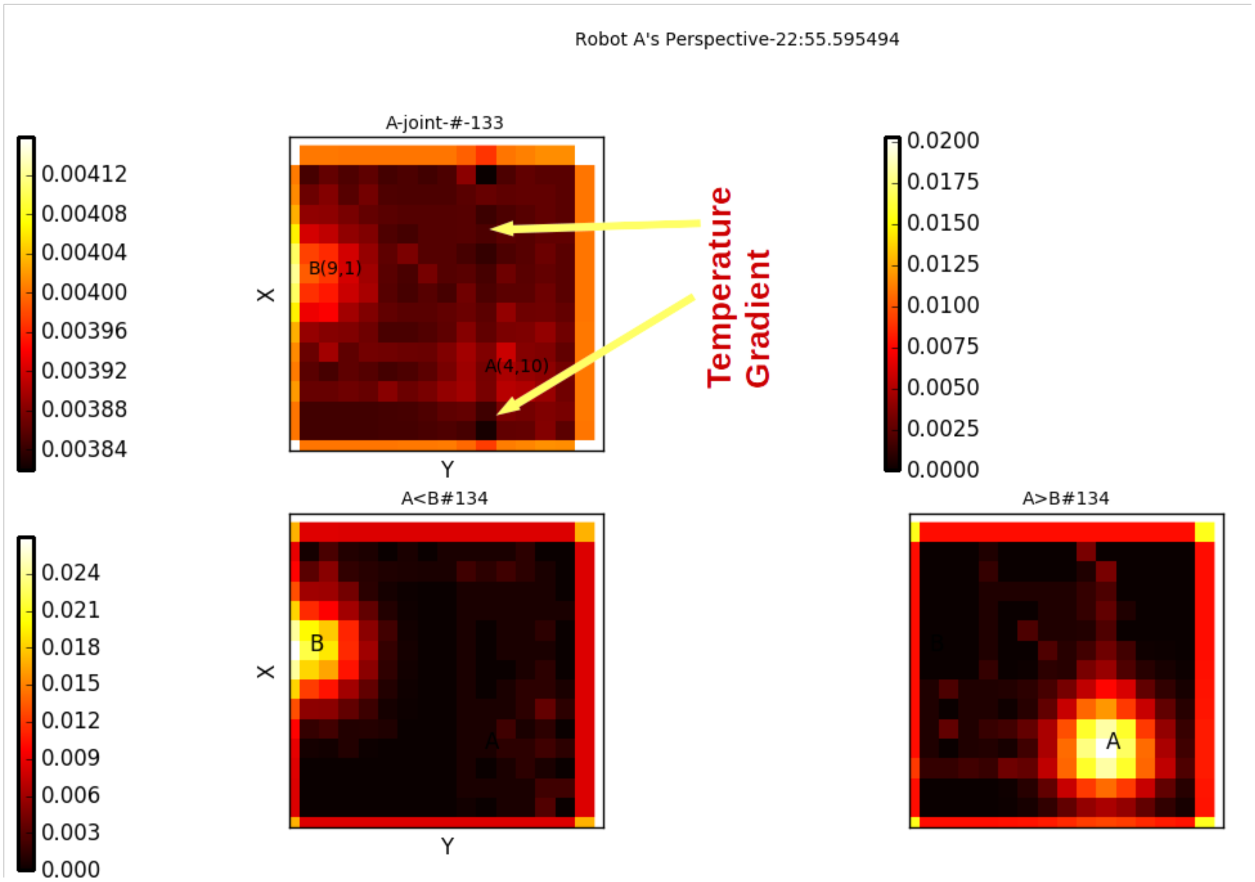


Figure 4.18: Heat map representation of joint belief in heterogeneous team of robots. Robot A explores along temperature gradient.

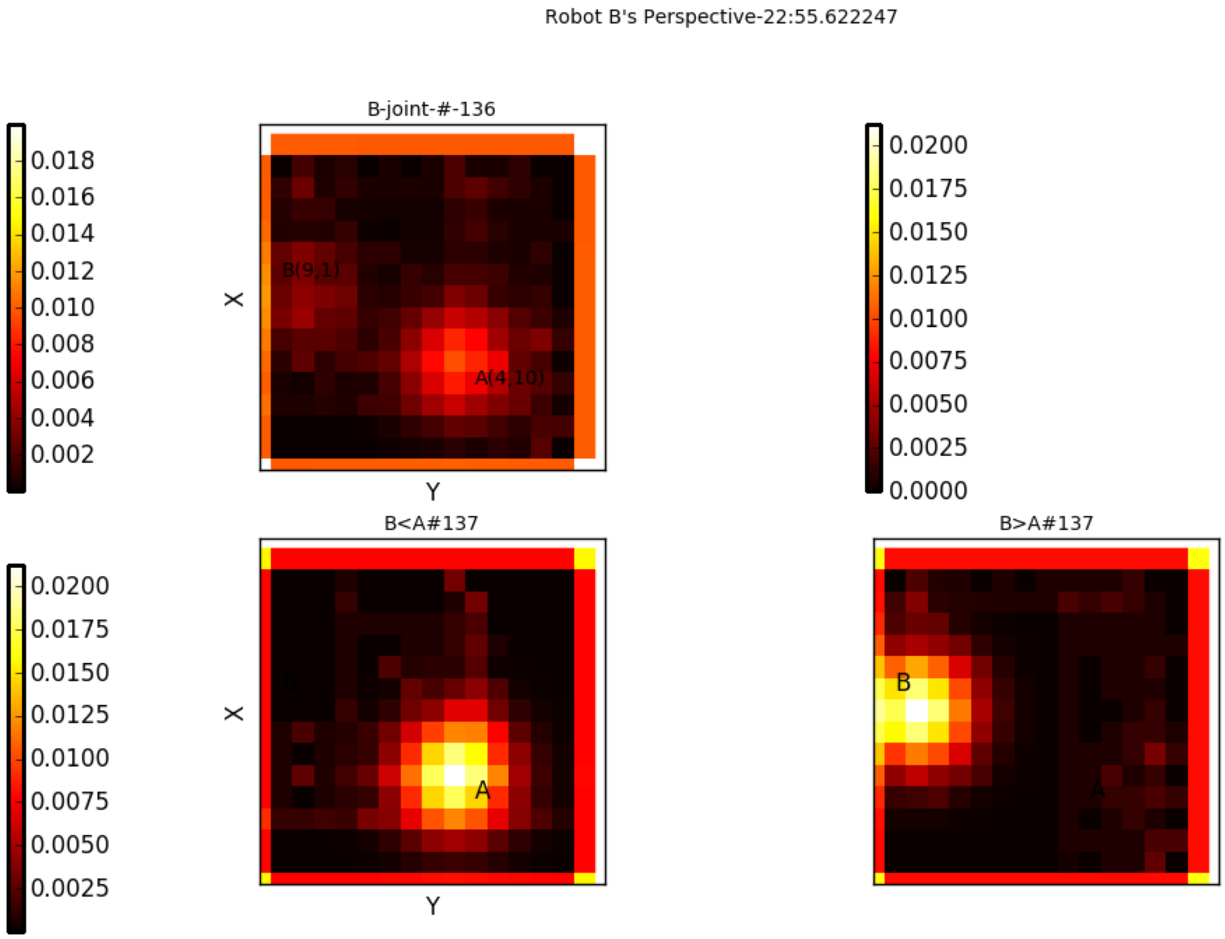


Figure 4.19: Heat map representation of joint belief in heterogeneous team of robots.Robot B explores the remainder of the environment

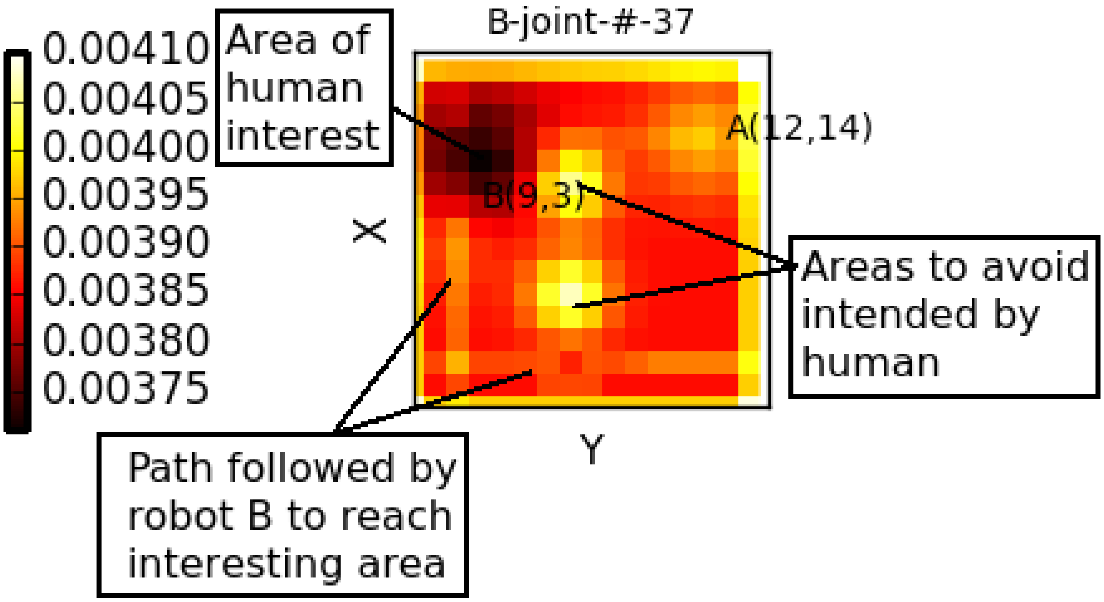


Figure 4.20: (a) Robot B proceeding to interesting area intended by human operator while avoiding dangerous areas.

human intentions. The robots begin their exploration tasks with the ϕ functions described as in the previous experiment. After a while, the human operator sends the intention to avoid two areas within the space, as shown in Figure 4.20, and also inserts an area of interest. Figure 4.21 shows how robot B reached the interesting area, avoiding the specified danger zones and additionally avoiding colliding with robot A. At the same time, the human's desires are promulgated to robot A through loopy propagation from B, but because B is already on the way, and in any case, A has the additional temperature-related exploratory goal, the two robots quickly reach an equilibrium which sends robot B to the new goal and leaves A to map the temperature gradient. After robot B explores the area of interest, the human marks another area of interest, and robot B heads in that direction safely and surely (Figure 4.22).

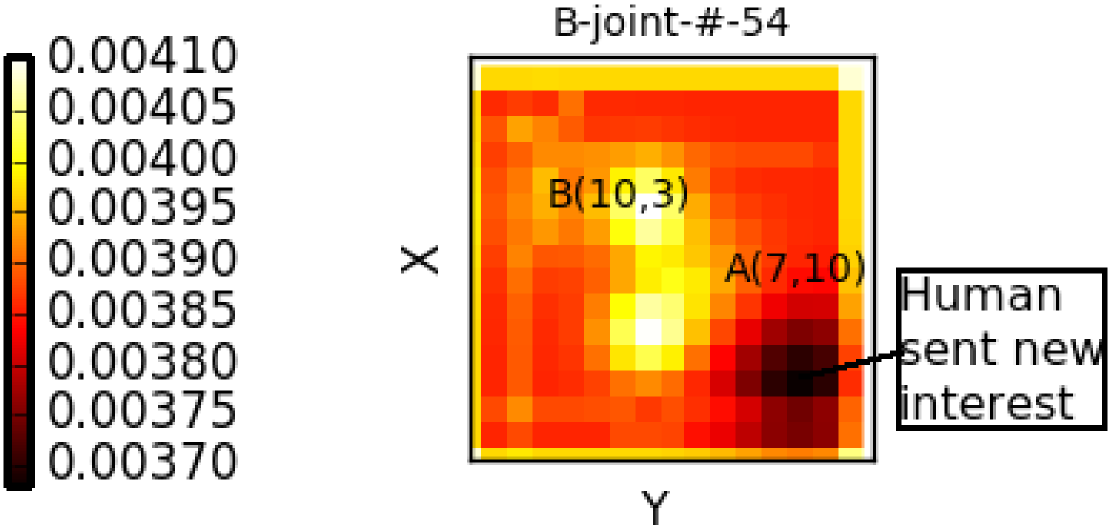


Figure 4.21: (b) B explores the interesting area.

4.3.6 Experiment With Real Robots

We have used two 3DR Solo quadrotors[3] for conducting this experiment. Each is equipped with a DHT22[4] temperature and humidity sensor. To introduce heterogeneity, one of the sensors only reported temperature, while the other sensed humidity. The choice of these particular sensors were motivated by our interest in collecting data for atmospheric physics using UAS. However, other kinds of sensors could be supported using the same approach; our algorithm is software-based and is useful for many different hardware choices. Our experiment was conducted within a 64 m^3 cube of airspace, with a base altitude 5 meters above the ground for safety.

The ϕ functions in place for this experiment were similar to those used on the simulator. A boundary ϕ_b function places a very high value (meaning a strong avoidance intention) at the boundary of the cubic space. The boundary intention and the collision avoidance intention ϕ_{ac} are both weighted very strongly. The collision avoidance intention has been

[3] https://3dr.com/solo-drone/

[4] https://www.adafruit.com/product/385

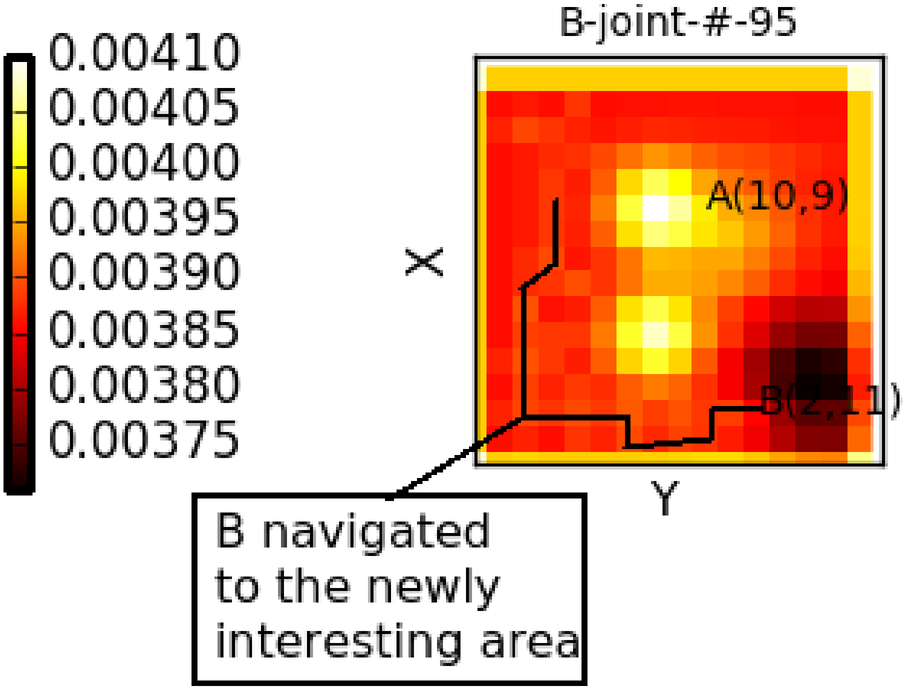


Figure 4.22: (c)Navigating to new interesting area inserted by human operator avoiding dangerous areas. Robot A continues to explore temperature gradient(not visible to B)

Table 4.6: Flight Test Summary

Date	Starting Time	Ending Time	Explored %	Avg Temp °F	Avg Rel Hum(%)	Temp Inversion Point (alt m)	Hum Inversion Point (alt m.)
Nov 14, 2017	10:16:59	10:16:59	30	70	70	45	45
Nov 14, 2017	10:16:59	15	15	70	70	45	45
Nov 10, 2017	16:06:19	15	15	70	70	45	45

designed again as a Gaussian probability mass function having the mean at the current location of a particular robot and a variance depending on relative robot speeds. The unexplored intention ϕ_u assigns higher values to locations that have been visited and are thus less interesting. The above ϕ functions are very similar to the functions we have used in simulation. As this experiment incorporates temperature and humidity sensors, we provide two more intentions, namely, ϕ_t and ϕ_h, much as we did previously. However, we do not know the temperature and humidity over the whole space. We only have the measurements at the current location of the robots and the places they have been before, and those values may change over time. However, we can infer the temperature and humidity gradients over unvisited space using the data we have already collected. The robots will attempt to locate and explore areas of rapid change, as these inversion and boundary layer phenomena are most informative to a meteorologist.

We simulate these sharp temperature and humidity changes in our weather-aware simulation environment. Fig. **??**(a-b) demonstrates that the robots are able to detect and map such sudden changes. Then, in Fig. **??**(c-d), we show the same behavior in a real world experiment. The robots are able to locate a temperature inversion at 45 meters above the ground.

We are also able to demonstrate seamless human intervention using our algorithm. In this particular instance, a human operator takes active control of one of the UAS, overriding the intentions developed by that robot. The robot, however, continues to communicate with the other team members, using the same loopy propagation framework. The other systems modify their intentions accordingly. Fig. 4.27 shows a human intentionally steering robot A toward its neighbor B. This induces robot B to evade, because of the influence of ϕ_{ac}. The human's intention is incorporated smoothly into the overall team behavior, without any explicit commands from the human to any other robot participant beyond the first.

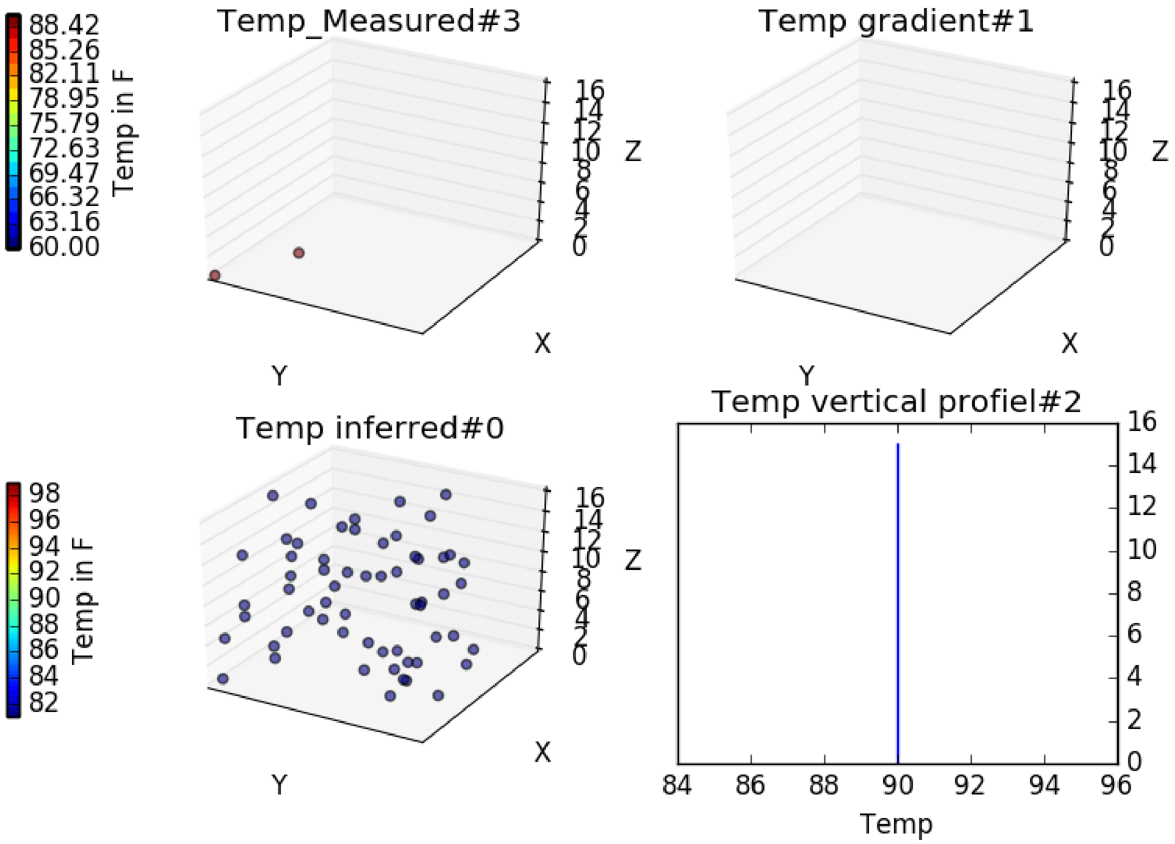


Figure 4.23: T1: In each subfigure, upper left is measured temperature, upper right is the inferred temperature gradient, lower left is randomly sampled temperature predictions drawn from the inferred gradient, and lower right is a temperature vs altitude plot. a-b and c-d are temperature profiles in simulation and the real world respectively. In each case, the first figure is early in the exploration process, and the second is after additional exploration and mapping.

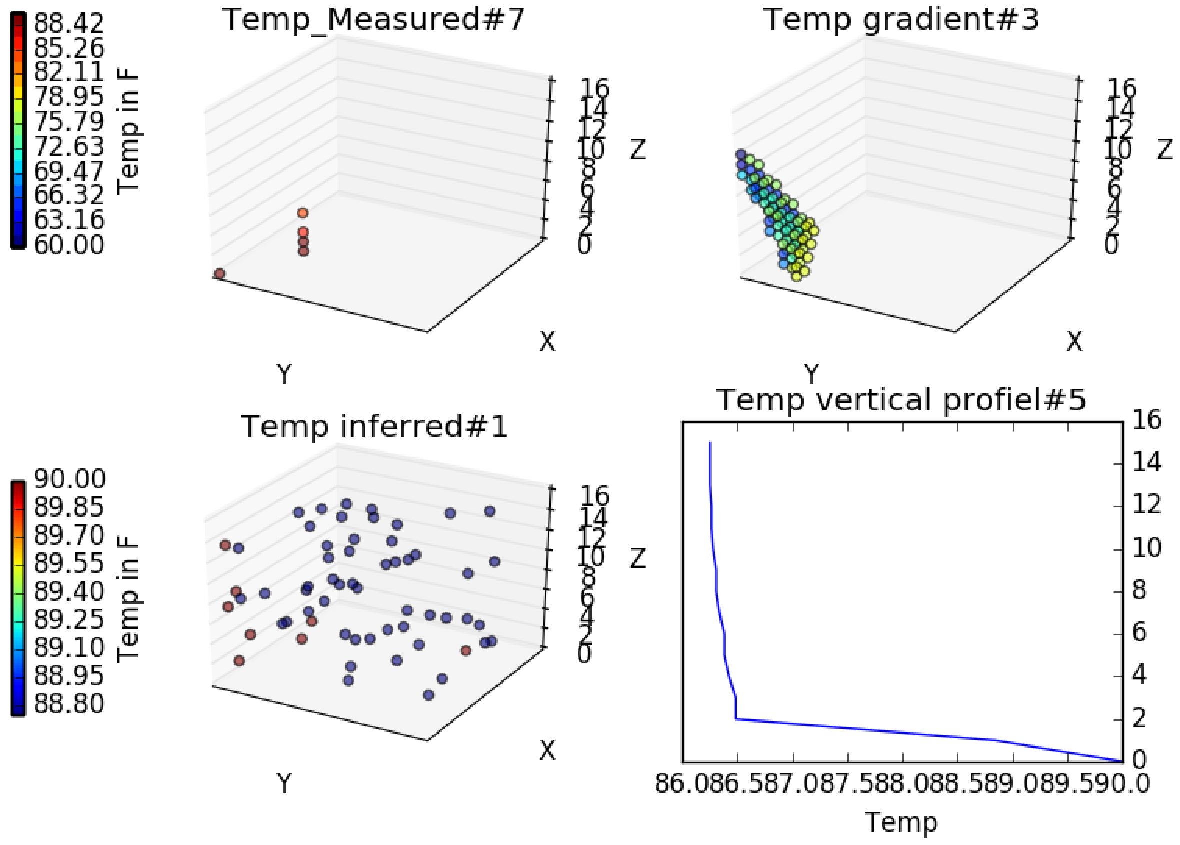


Figure 4.24: T2: In each subfigure, upper left is measured temperature, upper right is the inferred temperature gradient, lower left is randomly sampled temperature predictions drawn from the inferred gradient, and lower right is a temperature vs altitude plot. a-b and c-d are temperature profiles in simulation and the real world respectively. In each case, the first figure is early in the exploration process, and the second is after additional exploration and mapping.

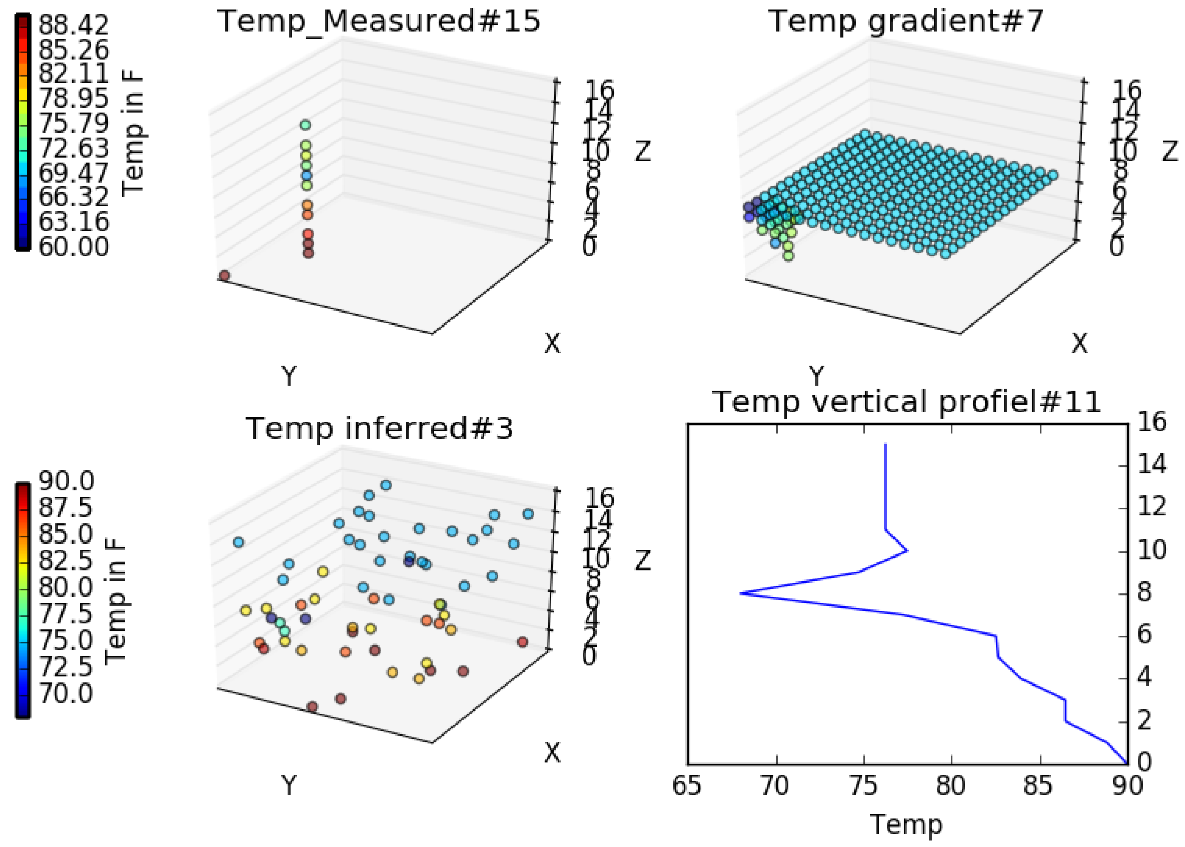


Figure 4.25: T4: In each subfigure, upper left is measured temperature, upper right is the inferred temperature gradient, lower left is randomly sampled temperature predictions drawn from the inferred gradient, and lower right is a temperature vs altitude plot. a-b and c-d are temperature profiles in simulation and the real world respectively. In each case, the first figure is early in the exploration process, and the second is after additional exploration and mapping.

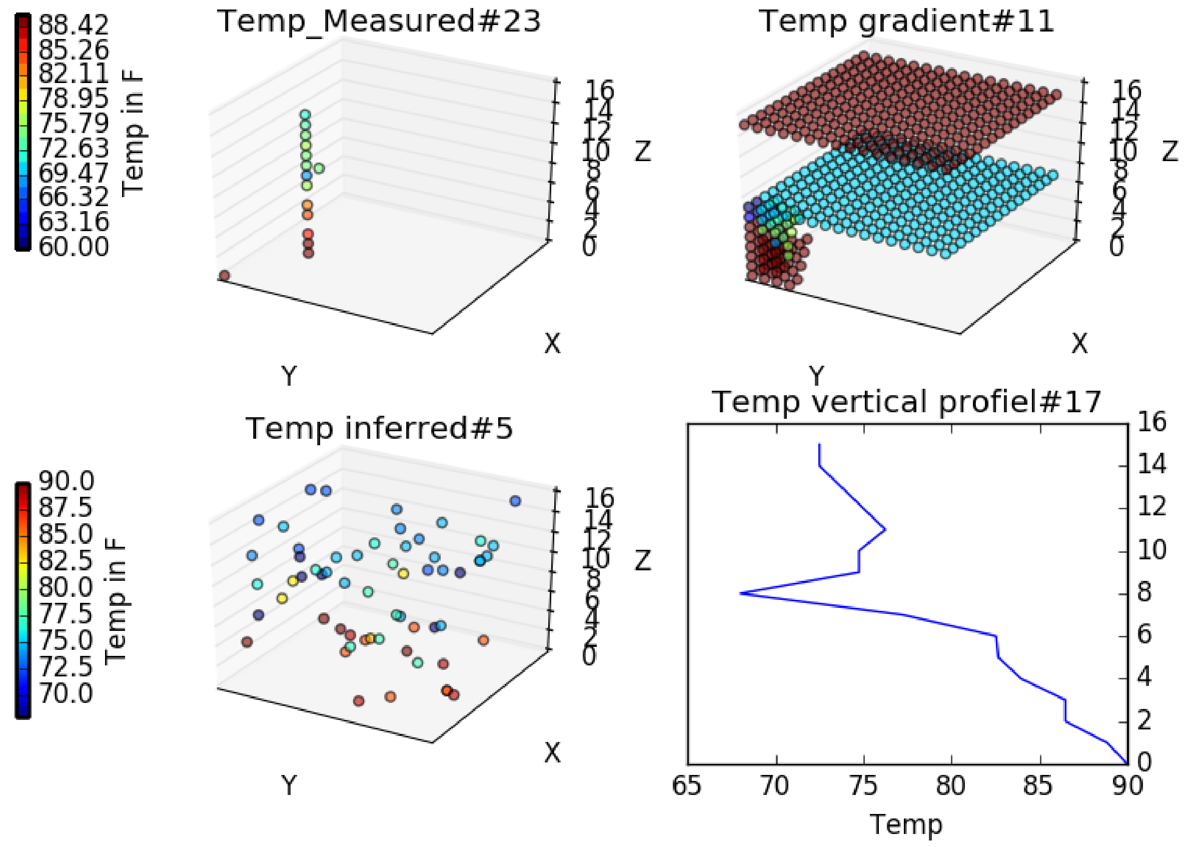
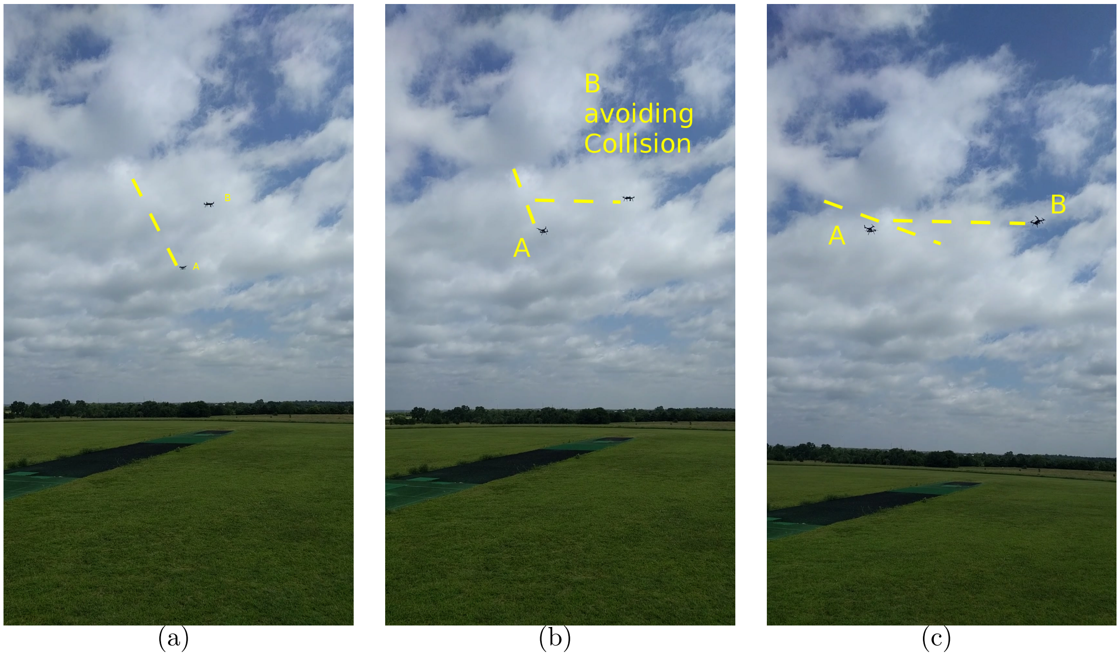


Figure 4.26: T6: In each subfigure, upper left is measured temperature, upper right is the inferred temperature gradient, lower left is randomly sampled temperature predictions drawn from the inferred gradient, and lower right is a temperature vs altitude plot. a-b and c-d are temperature profiles in simulation and the real world respectively. In each case, the first figure is early in the exploration process, and the second is after additional exploration and mapping.



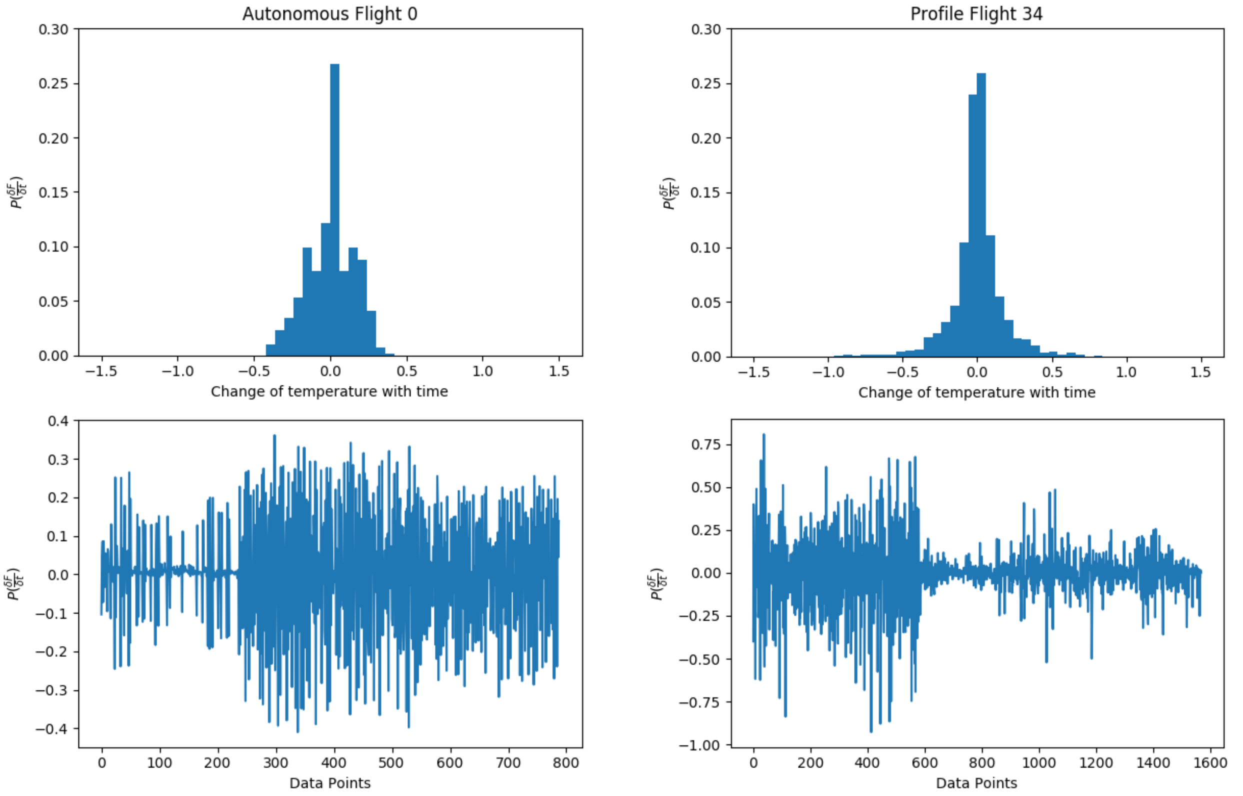
(a) (b) (c)

Figure 4.27: Experiment with two UAV robots A and B. (a) Human commands A to move toward B. (b) B moves to avoid collision with A. (e) A and B flying at safe distance again.

4.3.7 Experiment With *LOTS* of Real Robots

Our work contributed to the ISARRA Lower Atmospheric Process Studies at Elevation–a Remotely-piloted Aircraft Team Experiment[5] (LAPSE-RATE). Up to now, meteorologists and weather experts have used radar, balloon soundings and satellite data to model weather. However, as mentioned in the introduction, such sensor modalities cannot collect data in lower altitude environments effectively, and the atmospheric physics community does not have a good understanding of the boundary layer above the height of weather towers but below the safe operating envelopes of manned aircraft or the line of sight constraints of radar. We, along with more than 50 of our fellow researchers and colleagues from diverse research backgrounds and origins, participated in ISARRA LAPSE-RATE and collaboratively collected weather data using UAS. The experiment was conducted during a flight week campaign in the San Luis Valley in Alamosa, Colorado at several interesting sites from the perspective of atmospheric research. Collectively, 1200 flights were flown in this week by the participants, and our team contributed 215 flights and acquired 2.9 GB of sensor data.

[5] https://isarra.colorado.edu/flight-week



(a) Information Gain 2.257021, Duration 0:15:14, Information/Second=0.002469

(b) Information Gain 2.255881, Duration 0:28:09, Information/Second=0.001336

Figure 4.28: Plots for a representative autonomous flight are on the left; a preplanned profile flight is on the right. Measured temperature change over time ($\frac{\delta F}{\delta t}$) is in the bottom row; the top row collects these data into a histogram for information gain computation.

This is both the largest amount of data collected by UAS for weather measurement in a specific time period and geographic location, and involved more heterogenous UAS platforms in the data collection than ever before.

One of our research focuses for this period revolved around testing the performance of our coordination algorithm and comparing its performance with the current approach, where meteorologists and roboticists collaborate to devise planned waypoint-based missions to map the weather environment, usually taking the form of fixed vertical profiles or transects between two geographic points at various altitudes. For this experiment, we performed 13 autonomous flights and 64 fixed, planned flights using several different quadrotor and fixed-wing UAS.

Table 4.7: Comparison Between Autonomous and Profile Flight

Measure		Temperature	Humidity
Entropy	Autonomous	3.340759	2.997618
	Profile	3.424795	3.193233
Duration	Autonomous	05:17:11	05:06:03
	Profile	39:02:39	10:41:48
Info. Gain/sec	Autonomous	**0.0001755**	**0.0001632**
	Profile	0.0000244	0.0000256

All of these autonomous, profile, transect and other flights were conducted in the San Luis Valley, west of Great Sand Dunes National Park, Colorado. The experiments were conducted following FAA guidelines, with special permission to fly missions involving swarms, night flight and high altitudes. All the flights were required to have a human FAA-licensed UAS pilot in charge of the flight. The autonomous flights were conducted in multiple locations over half-hectare areas of open farm fields or rangeland. The transect flights followed flight paths of approximately 1 KM in length. Although we have access to the data from all 1200 flights, these experiments were performed across sixty flight hours, a small fraction of the total.

The overarching goal of the LAPSE-RATE campaign participants, roboticists and meteorologists alike, was to discover interesting phenomena in weather data, such as sudden changes to temperature or humidity. This information allows atmospheric physicists to understand and model convective activity and other atmospheric behavior. We hypothesized that our algorithm, which is specifically intended to coordinate multi-robot teams in real time in response to heterogenous sensor data, would be much more efficient at detecting and mapping these atmospheric phenomena, compared to UAS following fixed, preplanned profiles. We quantify the quality of the data collected by computing the entropy of the data distribution, and turn that into information gain per unit time. For this experiment we have used factor functions ϕ_t (temperature), ϕ_{ac} (collision avoidance), ϕ_b (boundary), as well as ϕ_h (humidity), which behaves just like the temperature function but is tied to a different sensor. We collected the sensory data into a normalized histogram, which essentially captures the probability density function of the changes of sensory data in the weather over time.

Table 4.8: Comparison Between Autonomous and Profile Flight

Entropy		Duration		Info. Gain/sec	
Autonomous	Profile	Autonomous	Profile	Autonomous	Profile
3.340759	3.424795	05:17:11	39:02:39	0.0001755	2.436553e-05

Table 4.9: Autonomous Flights: July 19 2018

	Entropy		Duration		Info. Gain/sec	
	Autonomous	Profile	Autonomous	Profile	Autonomous	Profile
Mean	1.935458	1.511861	00:21:50	00:13:14	0.003546	0.002759
Median	1.861298	1.476238	00:10:17	00:13:32	0.003322	0.001778
IQR	0.726217	0.664193	00:14:55	00:11:51	0.002566	0.001628

Figure 4.28 shows histograms computed for two representative flights. The overall information gain per flight is almost identical, and this is reflected in the similar shapes of the two histograms. However, as can be observed on the lower figure, the profile flight spent most of the second half of its trajectory in a very low information regime, where nothing particularly interesting was happening to the temperature. Thus the autonomously coordinated flight took half the flight time to obtain the same information.

Even a very poor deployment strategy for a meteorological sensor will continually produce information as it measures its environment over time. A better strategy will produce higher information gain over a specific amount of time. Thus we calculated the the information gain density simply by dividing the entropy by the duration of the flight by seconds. This quantity tells us how well a particular flight worked in gathering interesting information about the weather quantitatively. Table 4.7 shows a comparison between our autonomous flight tests and pre-planned waypoint based profile flights. Autonomous flights following our algorithm have higher information gain and spend lower flight times to collect that information. Thus, they have a higher information acquisition rate on average than fixed profile flights. On average our autonomous approach collected data around seven times more effectively for both temperature and humidity data. Because of the huge number of samples collected using both methods, our results are extremely statistically significant ($p < 10^{-7}$).

Chapter V

Conclusion

5.1 Assessing Cognitive Stress of Human Operator

Robots that are capable of understanding the cognitive load their operator is experiencing are vital to safe and efficient teamwork in complex scenarios where the proper level of autonomy and interaction is fluid. Vital communication cues are embedded in the way we behave in particular circumstances, and these implicit indicators do not have to be lost on our robots. My work's contribution is to demonstrate a quantitative, learnable, generalizable model that allows a robot to determine that a user has succumbed to cognitive stress, even when it cannot independently assess the instructions it is being given.

5.2 Multi-Modal Multi sensor Interaction between Human and Heterogeneous Multi-Robot System

I have presented a novel model for multimodal interaction among heterogeneous multi-robot systems. I have developed the algorithm for controlling such a system and demonstrated its effectiveness using experiments with real robots and in simulation. I have presented a toy example of how this new interaction method may work and also discussed its potential. In future, I intend to implement a fully functional version of this interaction. I will add more multimodal functionality like gyroscopes, accelerometers. I plan to build a multimodal interface framework based on a smartphone touch screen that will allow users to interact seamlessly with robot teams in many possible applications.

5.3 Distributed Control of Heterogeneous Team of robots and Humans

In this work, I have proposed a scheme for heterogeneous multi-agent control that uses factor graphs and loopy belief propagation to abstract intention away from the specifics of hardware capability and sensors, allowing a diverse collection of systems to be controlled with the same software and to interact effectively with each other. Additionally, human operators may insert themselves into the decision-making process to varying extents as desired. Our method enormously simplifies the logic and programming required to solve these kinds of problems. We have demonstrated the effectiveness of the approach in simple real-world scenarios and more complex simulated ones. At present, we have equipped actual UAV robots with real meteorological sensors and have demonstrated the efficacy of this approach in large real-world deployments, improving our understanding of near-surface weather phenomena and our ability to monitor and predict severe weather.

The contribution of this work is the novel algorithm for distributed heterogeneous control of robots with humans in the loop, and a very large-scale experiment which confirms its applicability, performance and robustness. We also illustrate theoretical evidence of its performance and developed a simulated environment which makes it possible accurately to generate real-world weather phenomena for multi-robot UAS testing. Our work facilitates human interaction with heterogeneous multi-robot teams. I have run an enormous and extensive investigation of lower altitude weather using heterogeneous UAS. The immense scale, duration and millions of data points collected demonstrate the capacity of our algorithm to deploy heterogenous robots over hundreds of square kilometers, investigating and mapping meteorological data with a speed and resolution unmatched by UAS whose autonomy is limited to flying pre-planned profiles and transects. This represents a large advance in the state of the art.

5.4 Limitations & Future Research Direction

My research has contributed algorithms and methodologies to coordinate task of heterogenous cooperative multi-robots and enable human robot interaction in efficient way. My research contributed in quantifying humans cognitive load in operating multiple robots. We are still far away from a general purpose algorithm for cooperative autonomous robot and human robot interaction but my work have shown promising result towards the right direction. Our experiments and data proved this consistently. There are few limitations of my research which can be addressed in new research. This brings me to future research direction for anyone who is interested in similar research problem. (1) How can we represent arbitrarily complex intentions of heterogeneous multi-robot teams in the factor graph representation that I have developed? (2) How can the intentions and beliefs of robots be represented to humans and vice versa?

My research results to this point have been obtained from known graph structures, factor functions and weights. The factor functions was carefully engineered to achieve certain objectives, for example, collision avoidance, exploration and efficient data sampling. Although the proposed algorithm and setting is theoretically applicable to any complex multi-robot task, these parameters and factor functions may not be easy to design. For most multi-robot tasks, the easier part in my approach is representing the robots in a graph with nodes representing the intention of each robot and edges representing their communication links. More challenging is to factorize the graph by designing appropriate factor functions and assigning optimal weights to interpret intention. For many expected behaviors, collision avoidance for example, we can engineer a function that explicitly makes a robot avoid intentions which have high probability of collision. However, how this collision avoidance intention should trade off with other expected intentions (for example, exploration) requires learning how to weight the factor functions. I wish to explore how we can initializing a factor function based on intuitive human knowledge, and the learn appropriate weights and parameters using reinforcement learning.

Bibliography

[1] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[2] Jerome Barraquand and Jean-Claude Latombe. Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, 10(6):628–649, 1991.

[3] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer, 1990.

[4] P Cheeseman, R Smith, and M Self. A stochastic map for uncertain spatial relationships. In *4th International Symposium on Robotic Research*, pages 467–474, 1987.

[5] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[6] Andrew H Jazwinski. *Stochastic processes and filtering theory.* Courier Corporation, 2007.

[7] Karoline Malchus, Petra Jaecks, Oliver Damm, Prisca Stenneken, Carolin Meyer, and Britta Wrede. The role of emotional congruence in human-robot interaction. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*, HRI '13, pages 191–192, Piscataway, NJ, USA, 2013. IEEE Press.

[8] Kuanhao Zheng, Dylan F Glas, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Supervisory control of multiple social robots for navigation. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 17–24. IEEE Press, 2013.

[9] Takayuki Kanda, Masahiro Shiomi, Zenta Miyashita, Hiroshi Ishiguro, and Norihiro Hagita. An affective guide robot in a shopping mall. In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, pages 173–180, New York, NY, USA, 2009. ACM.

[10] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Hähnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *IEEE International Conference on Robotics and Automation*, volume 3. IEEE, 1999.

[11] Peter R Wurman, Raffaello D'Andrea, and Mick Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29(1):9, 2008.

[12] Cynthia Breazeal. Social interactions in hri: the robot view. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(2):181–186, 2004.

[13] Jacob W Crandall, Michael Goodrich, Dan R Olsen Jr, Curtis W Nielsen, et al. Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(4):438–449, 2005.

[14] Dan R Olsen and Michael A Goodrich. Metrics for evaluating human-robot interactions. In *Proceedings of PERMIS*, volume 2003, page 4, 2003.

[15] Dipankar Das, Mohammed Moshiul Hoque, Yoshinori Kobayashi, and Yoshinori Kuno. Attention control system considering the target person's attention level. In *Proceedings of the 8th ACM/IEEE International Conference on Human-robot Interaction*, pages 111–112. IEEE Press, 2013.

[16] Mohammed Moshiul Hoque, Tomomi Onuki, Dipankar Das, Yoshinori Kobayashi, and Yoshinori Kuno. Attracting and controlling human attention through robot's behaviors suited to the situation. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, HRI '12, pages 149–150, New York, NY, USA, 2012. ACM.

[17] Jeffrey B. Brookings, Glenn F. Wilson, and Carolyne R. Swain. Psychophysiological responses to changes in workload during simulated air traffic control. *Biological Psychology*, 42(3):361 – 377, 1996. Psychophysiology of Workload.

[18] Zoltán Beck, Luke Teacy, Alex Rogers, and Nicholas R Jennings. Online planning for collaborative search and rescue by heterogeneous robot teams. In *Autonomous Agents & Multiagent Systems (AAMAS)*, 2016.

[19] John Roy Garratt. The atmospheric boundary layer. *Earth-Science Reviews*, 37(1-2):89–134, 1994.

[20] Christopher Crick, Sarah Osentoski, Graylin Jay, and Odest Chadwicke Jenkins. Human and robot perception in large-scale learning from demonstration. In *Proceedings of the 6th international Conference on Human-Robot Interaction*, pages 339–346. ACM, 2011.

[21] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, et al. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.

[22] Michael Goodrich, Dan R Olsen Jr, et al. Seven principles of efficient human robot interaction. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3942–3948. IEEE, 2003.

[23] Andrea Lockerd Thomaz and Cynthia Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *AAAI*, volume 6, pages 1000–1005, 2006.

[24] Andrea L Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6):716–737, 2008.

[25] Jessica R. Cauchard, Jane L E Kevin, Y Zhai, and James A. Landay. Drone & Me: An Exploration Into Natural Human-Drone Interaction. *UbiComp '15*, pages 361–365, 2015.

[26] Tayyab Naseer, Jurgen Sturm, and Daniel Cremers. FollowMe: Person following and gesture recognition with a quadrocopter. In *IEEE Int. Conf. Intell. Robot. Syst.*, pages 624–630. IEEE, nov 2013.

[27] Meghan Chandarana, Erica L. Meszaros, Anna Trujillo, and B. Danette Allen. Challenges of Using Gestures in Multimodal HMI for Unmanned Mission Planning. pages 175–182. 2018.

[28] Ramon A. Suarez Fernandez, Jose Luis Sanchez-Lopez, Carlos Sampedro, Hriday Bavle, Martin Molina, and Pascual Campoy. Natural user interfaces for human-drone multi-modal interaction. In *2016 Int. Conf. Unmanned Aircr. Syst.*, pages 1013–1022. IEEE, jun 2016.

[29] Andrea Sanna, Fabrizio Lamberti, Gianluca Paravati, and Federico Manuri. A Kinect-based natural interface for quadrotor control. *Entertain. Comput.*, 4(3):179–186, aug 2013.

[30] Jeongwoon Kim, David Hyunchul Shim, and James R Morrison. Tablet PC-based visual target-following system for quadrotors. *J. Intell. Robot. Syst. Theory Appl.*, 74(1-2):85–95, 2014.

[31] Sanjay K. Boddhu, Matt McCartney, Oliver Ceccopieri, and Robert L. Williams. A collaborative smartphone sensing platform for detecting and tracking hostile drones. volume 8742, page 874211. International Society for Optics and Photonics, may 2013.

[32] Danielle Cummings, Stephane Fymat, and Tracy Hammond. Sketch-based Interface for Interaction with Unmanned Air Vehicles Work-in-Progress. 2012.

[33] D Cummmings, S Fymat, and Tracy Hammond. RedDog : A Smart Sketch Interface for Autonomous Aerial Systems, 2012.

[34] Jesse Butterfield, Odest Chadwicke Jenkins, and Brian Gerkey. Multi-robot markov random fields. In *Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2008.

[35] Ryan K Williams and Gaurav S Sukhatme. Cooperative multi-agent inference over grid structure Markov random fields. In *Intelligent Robots and Systems (IROS)*, 2011.

[36] Shayegan Omidshafiei, Christopher Amato, Miao Liu, Michael Everett, Jonathan P. How, and John Vian. Scalable accelerated decentralized multi-robot policy search in continuous observation spaces. In *International Conference on Robotics and Automation (ICRA)*, 2017.

[37] Seng Keat Gan, Robert Fitch, and Salah Sukkarieh. Real-time decentralized search with inter-agent collision avoidance. In *International Conference on Robotics and Automation (ICRA)*, 2012.

[38] Pablo Lanillos, Seng Keat Gan, Eva Besada-Portas, Gonzalo Pajares, and Salah Sukkarieh. Multi-UAV target search using decentralized gradient-based negotiation with expected observation. *Information Sciences*, 282:92–110, 2014.

[39] Alessandro Renzaglia, Christophe Reymann, and Simon Lacroix. Monitoring the evolution of clouds with UAVs. In *International Conference on Robotics and Automation (ICRA)*, 2016.

[40] Christophe Reymann, Alessandro Renzaglia, Fayçal Lamraoui, Murat Bronz, and Simon Lacroix. Adaptive sampling of cumulus clouds with UAVs. *Autonomous Robots*, pages 1–22, 2017.

[41] Christopher Crick and Avi Pfeffer. Loopy belief propagation as a basis for communication in sensor networks. In *Uncertainty in Artificial Intelligence (UAI)*, 2003.

[42] Zachary N. Sunberg, Mykel J. Kochenderfer, and Marco Pavone. Optimized and trusted collision avoidance for unmanned aerial vehicles using approximate dynamic programming. In *International Conference on Robotics and Automation (ICRA)*, 2016.

[43] Yu Fu, Xiang Yu, and Youmin Zhang. Sense and collision avoidance of unmanned aerial vehicles using Markov decision process and flatness approach. In *International Conference on Information and Automation (ICIA)*, 2015.

[44] Peter Stone and Manuela Veloso. Multiagent systems: a survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[45] Waseem Abbas and Magnus Egerstedt. Distribution of agents with multiple capabilities in heterogeneous multiagent networks - A graph theoretic view. *Proceedings of the IEEE Conference on Decision and Control*, pages 290–295, 2012.

[46] Philip Twu, Yasamin Mostofi, and Magnus Egerstedt. A measure of heterogeneity in multi-agent systems. *Proceedings of the American Control Conference*, pages 3972–3977, 2014.

[47] Radu Bogdan Rusu, Alexis Maldonado, Michael Beetz, and Brian Gerkey. Extending player/stage/gazebo towards cognitive robots acting in ubiquitous sensor-equipped environments. In *ICRA Workshop for Networked Robot Systems, 2007, Rome, Italy*, 2007.

[48] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.

[49] Jeff Craighead, Robin Murphy, Jenny Burke, and Brian Goldiez. A survey of commercial & open source unmanned vehicle simulators. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 852–857. IEEE, 2007.

[50] Michal Pechoucek and David Sislak. Agent-based approach to free-flight planning, control, and simulation. *Intelligent Systems, IEEE*, 24(1):14–17, 2009.

[51] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, volume 3, page 5, 2009.

[52] Christopher Crick, Graylin Jay, Sarah Osentoski, Benjamin Pitzer, and Odest Chadwicke Jenkins. Rosbridge: Ros for non-ros users. In *Proceedings of the 15th International Symposium on Robotics Research*, 2011.

[53] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[54] Judea Pearl. *Probabilistic reasoning in intelligent systems.* Morgan Kaufmann, 1988.

[55] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[56] Harris Drucker, Donghui Wu, and Vladimir N Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

Appendix A

An Appendix Explained

There is no appendix for this dissertation.

VITA

S M Al Mahi

Candidate for the Degree of:

Doctor of Philosophy

Dissertation: An Algorithm for Distributed Heterogeneous Robot-Human Teams

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for Doctor of Philosophy in Computer Science at Oklahoma State University, Stillwater Oklahoma in May, 2020.

Completed the requirements for Bachelor of Science in Computer Science at Islamic University of technology, Gazipur Bangladesh in 2011.

Experience:

Employed by Oklahoma State Univeristy in the position of Research/Teaching Assistant in Stillwater, Oklahoma from August 2014 to date.

Professional Memberships:

Student Member of IEEE, ACM as of 2017, 2018, 2019.