

Dynamic Mission Planning for Communication Control in Multiple Unmanned Aircraft Teams

by

Andrew N. Kopeikin

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

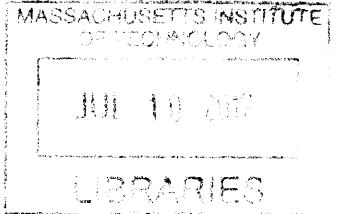
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Jun 2012

© Massachusetts Institute of Technology 2012. All rights reserved.

ARCHIVES



Author
Department of Aeronautics and Astronautics
May 24, 2012

Certified by
Jonathan P. How
Richard C. Maclaurin Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by
Professor Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Dynamic Mission Planning for Communication Control in Multiple Unmanned Aircraft Teams

by

Andrew N. Kopeikin

Submitted to the Department of Aeronautics and Astronautics
on May 24, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

As autonomous technologies continue to progress, teams of multiple unmanned aerial vehicles will play an increasingly important role in civilian and military applications. A multi-UAV system relies on communications to operate. Failure to communicate remotely sensed mission data to the base may render the system ineffective, and the inability to exchange command and control messages can lead to system failures. This thesis presents a unique method to control communications through distributed mission planning to engage under-utilized UAVs to serve as communication relays and to ensure that the network supports mission tasks. The distributed algorithm uses task assignment information, including task location and proposed execution time, to predict the network topology and plan support using relays. By explicitly coupling task assignment and relay creation processes the team is able to optimize the use of agents to address the needs of dynamic complex missions.

The framework is designed to consider realistic network communication dynamics including path loss, stochastic fading, and information routing. The planning strategy is shown to ensure agents support both data-rate and interconnectivity bit-error-rate requirements during task execution. In addition, a method is provided for UAVs to estimate the network performance during times of uncertainty, adjust their plans to acceptable levels of risk, and adapt the planning behavior to changes in the communication environment. The system performance is verified through multiple experiments conducted in simulation.

Finally, the work developed is implemented in outdoor flight testing with a team of up to four UAVs to demonstrate real-time capability and robustness to imperfections in the environment. The results validate the proposed framework, but highlight some of the challenges these systems face when operating in outdoor uncontrolled environments.

Thesis Supervisor: Jonathan P. How

Title: Richard C. Maclaurin Professor of Aeronautics and Astronautics

Acknowledgments

The chance to study aeronautics at MIT was an opportunity I never imagined I would get. The last two years spent with some of the world's most intelligent, entrepreneurial, and hardworking people has been both the most enriching and humbling experience of my life.

I thank my advisor, Professor Jonathan How for mentoring me through my research with inspiring dedication and thoroughness. I also thank several key people: Sam Ponda and Luke Johnson from ACL, Andrew Clare from HAL, and Olivier Toupet from Aurora, who spent countless hours helping me with my research. I simply could not have completed this work without them. Special thanks to Professor Missy Cummings for her mentorship in my first year of research. Thanks to all the members of the Aerospace Controls Lab for showing me the ropes, especially to Tuna Toksoz for always magically getting my computer code to work; Buddy Michini, Mark Cutler, and Josh Redding for teaching me hardware implementation; and Josiah Vandermeijer and Vishnu Desraj for their help with all of the above and their encouragements. Thank you also to the undergrads who helped me with flight testing: Aaron Prindle, Mike Klinker, Will Waste, Sarah Ferguson, and Paul Quimby.

I'm very grateful to Aurora Flight Sciences for sponsoring my second year of grad-school with the Aurora Fellowship, as well as sharing time and expertise with me to complete the research. This work was also sponsored in part by ONR under STTR N00014-08-C-0707 and should not be interpreted as representing the official policies or endorsements of the U.S. Government.

Finally I also recognize my friends and family for their support and for giving me the motivation to take on this challenging endeavor. Many came to visit me in Boston which is really appreciated. Most importantly I thank Ashley, my long time love and girlfriend, for her support, encouragements, and patience in the past two years as I spent long days and nights away from her to complete this work. Thank you.

Contents

1	Introduction	13
1.1	Overview	13
1.2	Motivation, Objectives, and Gaps	15
1.3	Challenges	18
1.4	Thesis Contributions	22
1.5	Thesis Organization	23
2	Network Communication in Multi-Unmanned Vehicle Systems	25
2.1	Network Communication Background	25
2.2	Why Communications Matter in a Team	34
2.3	Communication Network Control Methods	38
2.4	Summary	47
3	Distributed Dynamic Planning Algorithm	49
3.1	Task Allocation Planning Strategies	49
3.2	Consensus Based Bundle Algorithm	52
3.3	CBBA with Relays	54
3.4	Summary	61
4	Improving Network Support	63
4.1	Motivation for Improvements	63
4.2	System Overview	65
4.3	Supporting Data-Rate Requirements	76
4.4	Supporting Error-Rate Requirements	83
4.5	Planning with Uncertainty	87
4.6	Performance Evaluation	90
4.7	Summary	99

5	Flight Test Experiments	101
5.1	Overview	101
5.2	Experimental Setup	103
5.3	Experiment 1 - Original CBBA with Relays	106
5.4	Experiment 2 - Updated CBBA with Relays	109
5.5	Summary	114
6	Decentralized Multi-UAV Control Under Human Supervision	117
6.1	Motivation and Overview	117
6.2	Background Information	118
6.3	Flight Test Architecture	121
6.4	Flight Test Implementation	125
6.5	Flight Test Results	130
6.6	Summary	138
7	Conclusion and Recommendations	141
7.1	Conclusions	141
7.2	Future Work	144
	References	160

List of Figures

1-1	Cooperative aerial surveying mission with a multi-UAV team	16
2-1	Centralized and decentralized network architectures	27
2-2	Ad-hoc On-Demand Distance Vector (AODV) routing protocol	33
3-1	Consensus Based Bundle Algorithm	52
3-2	Different task allocation strategies for network support	55
3-3	CBBA with Communication Relays Algorithm	57
4-1	Motivation to improve CBBA with Relays	64
4-2	Wireless channel performance model	67
4-3	Improved CBBA with Communication Relays	69
4-4	Relay creation strategies	72
4-5	System framework block diagram	75
4-6	Probability of messaging error hop to hop	84
4-7	Estimation of communication environment in simulation	91
4-8	Results: Effects of varying incentive to serve as relay	93
4-9	Results: Mission score for different planning strategies	95
4-10	Results: Assignment supportability for different planning strategies	95
4-11	Results: Inter-connectivity and algorithmic convergence comparison	97
4-12	Results: Effects of varying planning conservatism	98
4-13	Results: Adapting to changes in the communication environment	100
5-1	CBBA with Relays flight test architecture	104
5-2	CBBA with Relays visual interface	105
5-3	Flight test systems	106
5-4	Flight testing the CBBA with Relays framework	107
5-5	Results: Baseline CBBA with Relays performance in flight test	109
5-6	Results: Improved CBBA with Relays performance in flight test	112
5-7	Results: Network control performance in flight test	113

5-8	Results: Uncertainty and adaptive planning in flight test	114
5-9	Results: Estimation of communication environment in flight test . . .	115
6-1	Architecture for multi-UAV team with human supervision	122
6-2	Receding-horizon path-planning	124
6-3	Decentralized multi-UAV human interface	125
6-4	Team of four UAVs	126
6-5	Flight test system architecture	127
6-6	Targets used for during field test	128
6-7	Churning in task allocation due to nonlinear voltage discharge	133
6-8	Churning in trajectory planning due to competing deconfliction paths	133
6-9	Results: time performance metrics in flight test	135
6-10	Results: Human operator utilization in flight test	136

List of Tables

4.1	Communication Model Parameters	66
4.2	General Simulation Mission Parameters	92
5.1	Results: Communication Estimation in Flight Test	113
6.1	Results: System Failures in Flight Test	131

Chapter 1

Introduction

1.1 Overview

Unmanned Air Vehicles (UAVs) have become central to a new era of aviation by providing game changing capabilities in recent wars, and showing tremendous promise in numerous civilian applications including surveying, first response, and transportation, to name a few. Current operational UAVs are typically remotely piloted by humans with basic autonomy similar to modern manned aircraft autopilots. For instance, an MQ-1 Predator on an Intelligence, Surveillance, and Reconnaissance (ISR) mission is actively controlled by a pilot and payload operator from a remote ground control station. RQ-11 Ravens used as small tactical assets are controlled either manually or through user specified waypoints to accomplish a mission. Despite the success of these systems, recent progress in unmanned vehicle (UV) control and autonomy is challenging the current concept of remote piloting by allowing teams of multiple unmanned agents to autonomously collaborate and improve system performance under the supervision of fewer human operators [1–3].

The “real-world” situations which will utilize teams of unmanned systems will involve executing complex missions where the number, status, and types of tasks, as well as the environment vary dynamically. As with all management disciplines, the multi-UV system must continuously assess its capabilities and properly allocate resources to overcome changes and maximize performance. A fleet of multiple UAVs

may be heterogeneous, where different systems have different capabilities in sensor configurations or flight performance. The objective of dynamic mission planning is to ensure that tasks are performed effectively by the right UAV, at the right place, at the right time.

A significant challenge in this process involves maintaining proper communications to execute the mission. Controlling the network is often just as important as controlling the vehicles themselves as inadequate communications can significantly degrade team performance. There are several general architectures to deploy multiple UAVs as a team. UAVs may be connected only to a central node, such as a human operated control station, which receives information, computes a plan of action, and sends instructions specific to each vehicle (centralized control). Alternatively, UAVs may be connected directly to each other, exchange information, and each formulate their own plan without a designated leader (decentralized control) [4, 5]. Hybrid centralized and decentralized solutions also exist, and UAVs may also cooperate with other manned and unmanned space, air, ground, and sea agents over the network in a joint effort [6–8]. In all cases, team coordination requires agents to exchange state information, observations of the world, and control decisions such as task allocation or motion planning. Furthermore, if the UAVs are being used to gather information, as is often the case, the collected data may need to be communicated to a designated point for analysis, possibly in real-time [9].

Because UAVs are highly mobile vehicles, information is most commonly exchanged across the network using wireless communication. Signals containing encoded messages travel between transmitting and receiving radio modules over wireless channels. The quality of the channel is fundamentally based on the strength of the signal at the receiver compared to noise and interference in the environment, or Signal to Noise Ratio (SNR) [10]. The quality of the link drives the probability that information transmitted will successfully be received, and affects the rate at which information can be exchanged over that channel.

Since multi-UAV team operations are still fairly new in concept, methods to control communications in these systems are actively evolving. The network itself can be

controlled through its topology, which describes the set of interconnections between nodes, and how strong those connections are. There is a large design trade-space in communication systems engineering (coding, modulation, multi-access, antenna design, etc.) which can significantly impact the performance of the network [11] and is beyond the scope of this thesis. Since wireless channels generally degrade with increasing distance and obstacles in the line of sight, the most relevant control method to multi-vehicle systems involves properly positioning agents to support the network. Common ways to do this include planning the motion of agents subject to communication requirements [12–28] or deploying agents designated as mobile communication relays [29–40]. The method described in this thesis differs from previous studies by controlling the network through task allocation. The team cooperates through task allocation to (1) ensure tasks undertaken are supported by the network, and (2) dynamically assign under-utilized agents to serve as communication relays. By explicitly coupling the task assignment and relay planning processes, the team is able to better optimize the use of agent resources to address current mission needs. This leads to improved performance and added flexibility in real-time dynamic mission scenarios. The proposed framework considers the uncertainties associated with wireless channels and the dynamics of information routing protocols which affect network performance. In addition, other relevant elements of multi-UAV cooperative control are also explored. These include human supervisory control, design of a system robust to uncertainties and failures, and general outdoor flight test operations with multi-UAV systems.

1.2 Motivation, Objectives, and Gaps

The primary objective in this thesis is to develop a multi-UAV dynamic mission planning framework which ensures communication requirements are supported during task execution. The goals are illustrated through a motivating scenario representative of envisioned multi-UAV operations (Fig. 1-1). The scenario consists of a complex mission where time-sensitive survey tasks dynamically appear during execution. Each task requires a UAV to travel to the task location, and transmit remotely sensed data

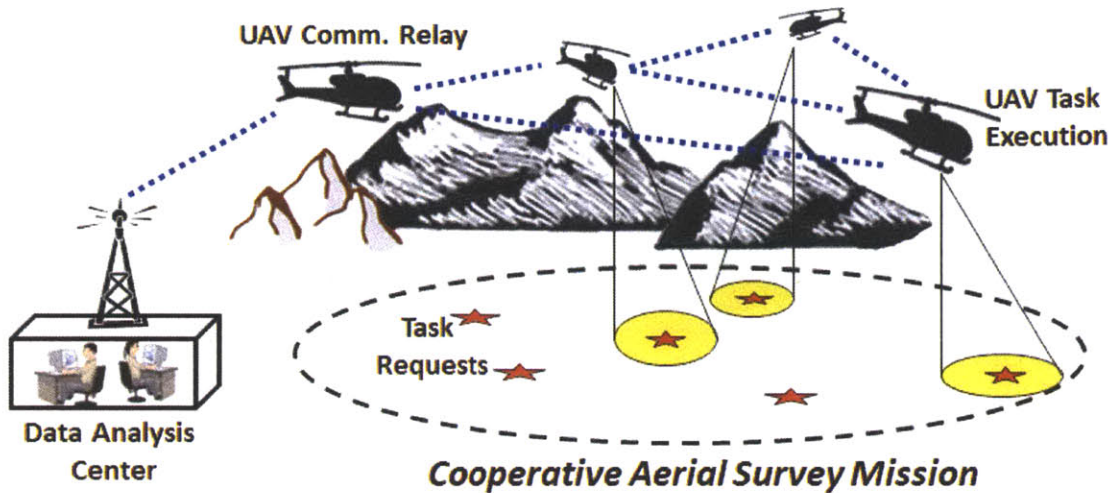


Figure 1-1: Cooperative aerial surveying mission with a multi-UAV team

in real-time to a base station for analysis. Every task completed provides mission reward, which varies based on the importance of that task, how timely it was addressed, and whether the communication network supported its data transmission to the base. Traveling to the task however, incurs cost which counts against reward gained. The objective is to properly coordinate a team of UAVs to achieve high mission performance by maximizing reward and minimizing cost. UAVs can either execute mission tasks or alternatively serve as communication relays to support the network.

The system is controlled in a distributed architecture, where each agent makes its own decisions regarding task assignment. This architecture has several benefits for a multi-agent system. First, it reduces the need to communicate an excessive amount of state information to a ground control station for centralized control. Second, it mitigates latency effects by enabling agents to leverage their immediate local situational awareness in planning the task assignment.

The objectives of this thesis are to design a system that addresses the motivating mission scenario which:

1. Includes a distributed dynamic resource allocation planner which operates in real-time (as opposed to preprocessed or offline) to effectively task agents during the mission.

2. Ensures that the communication network supports data-rate requirements needed to send remotely sensed information at each task to the base.
3. Ensures that the network supports inter-UAV message exchanges to allow UAVs and the base to effectively coordinate plans.
4. Accounts for uncertainty in network performance and can adapt to changes in the communication environment.

Recently, there has been considerable interest in methods to control network communications for a team of mobile agents. Two types of problems are commonly studied. The first consists of controlling a multi-agent system to achieve a defined primary mission objective while staying connected. Such objectives can include maximizing area coverage, tracking a target, rendezvous in space, or formation and flocking control [13, 22, 23, 41]. Network performance depends on the quality of wireless links, the number of interconnected agents, and the ability to route information. These factors are affected by the network topology which can be controlled by properly positioning agents. As such, the problem now becomes controlling team members to achieve the mission objective while simultaneously maintaining a state of connectivity dictated by system requirements. These requirements vary from maintaining connectivity with a specific number of neighbors, to establishing routes with sufficient throughput to send data to a base station. Different motion control strategies have been used which include using potential fields [13] to attract vehicles to each other, reactive control to stop agents from continuing into disconnected states [18, 24], or even adaptive control by changing the motion planning behavior of agents based on sensed communication measurements [28].

The second type of problem consists of controlling a set of agents designated as communication relays to support an underlying network. Here, the explicit primary mission objective is to support connectivity either by maximizing network performance, or minimizing the use of relay resources. Such problems include controlling a chain of relays between two mobile end nodes [32], or optimal relay deployment to support a larger, typically fixed network [40]. Solutions to these problems commonly

use similar motion control strategies as described above, or turn to graph theoretic and network optimization methods to solve a relay deployment plan. These formulations are described in greater detail in Section 2.3.

One of the objectives in this thesis is to build on network communication control strategies proposed in these studies, and include them in a framework which enables distributed execution of complex missions in dynamic environments. Previous work in the literature typically show these methods in simple or static mission scenarios which do not evolve over the course of execution. Furthermore, agent roles are generally predefined and fixed during the mission. Several studies have investigated task allocation for communication control, but the methods presented had centralized architectures [42] and treated network requirements as a constraint to task allocation rather than assigning agents cooperatively to achieve better results [18]. In addition, many studies in the literature assume simplistic communication requirements and deterministic environments which break down in real world operations. As such, the framework in this thesis attempts to simultaneously satisfy several different communication requirements while considering realistic and uncertain networking dynamics. Finally, there is a significant gap in this research field between work performed in simulation or controlled lab environments and real-world operations. To address this, the framework developed in this thesis are implemented in outdoor flight test experiments with a team of three UAVs to show real-time operation in an uncontrolled environment.

1.3 Challenges

1.3.1 Network Control Challenges

There are numerous and often competing challenges in controlling network communications in a multi-UAV system. The design must properly balance ensuring communication requirements are met without over-constraining system operation. The control of network communications follows the same feedback control principles used

for many other types of systems. Network topology measurements are used as state feedback. A model of the network dynamics is then used to select control inputs for packet transmission and information routing. Modeling and accurately predicting the performance of wireless channels is a significant challenge in this process. In general, links degrade with increasing inter-node distance due to path loss. Signal shadowing occurs due to obstacles in the environment such as buildings, mountains, or even the platform vehicle itself. Furthermore, multipath replicas of the signal reflecting and scattering off these obstacles constructively or destructively interfere with each other at the receiver. Path loss, shadowing, and multipath fading effects are very detrimental and difficult to predict in congested operating environments [10, 43, 44]. Fortunately, large and medium scale UAVs often operate in open skies, which mitigates many of the detrimental effects. However, actively studied concepts of mini/micro-UAV teams deployed below the tree line, inside buildings, or in urban settings will be significantly affected by these dynamics [45]. Similarly, noise and interference from other emitting sources in the environment can be difficult to predict and can change both in time and by location [46]. In some contested environments, another source may actively seek to interfere or “jam” communications to degrade the system performance as a counter-measure [1]. For similar reasons, measurements of the signal strength, while accurate, may also fluctuate significantly at time-scales of only tens or hundredths of a second. The uncertainties in the model and the rapid variation in the measurements pose a significant challenge in the network feedback control loop, and lead to limited performance guarantees of the system [10].

Given the ability to control the network topology, the problem of efficiently routing information between nodes is a challenge in itself [10]. Different types of data have different communication requirements, but in general the typical objective is to route data to its destination with minimal delay, over links with sufficient data-rate capacity and with minimal number of packets dropped due to error. A wireless network typically operates on one or more communication channels in a specific range of the electromagnetic spectrum. As such, nodes transmitting on the same or neighboring channels interfere with each other, which generates a design trade [47] best illustrated

through a simple example. Assume that a UAV needs to transmit time sensitive data to a distant base. One option is to increase its transmitter power enough to send the data directly to the base. However, this higher transmit power now interferes more with other UAVs also trying to communicate. Furthermore, the long link may be strong enough to reach the base under nominal conditions, but may not be robust to channel fading. Another option then is for the UAV to transmit over a more robust link and at a lower power to another less distant UAV, which can then relay the message, possibly through additional hops, to the base. This reduces interference from that UAV, but may cause delays in end-to-end delivery of the data, and cost overhead in the network to establish the multi-hop route. The challenge of selecting the best route and power settings is further exacerbated as the number of agents increases, as wireless link qualities continuously fluctuate under fading, and because the network topology is dynamic as agents move around to execute the mission [10]. This is especially relevant for multi-UAV systems, which can involve vehicles traveling rapidly over large distances.

Multi-agent networks of all types face the communication control challenges described above. The degree to which these challenges affect the system changes based on the domain of operation. For instance, Unmanned Ground Vehicles (UGVs) typically operate in environments with obstacles to line-of-sight, resulting in channel fading, but they have the ability to stop when a suitable location to transmit is encountered [48]. Autonomous Underwater Vehicles (AUVs) have limited underwater communication capabilities, and may only be able to exchange coordinated control information once every few hours upon surfacing [3]. Unattended Ground Sensors (UGS) forming a fixed Wireless Sensor Network (WSN) may be less susceptible to dynamic changes in topology, but need to be power conscious when transmitting because their battery lifetime is limited [36]. There are a number of challenges especially relevant for UAVs. First, the size, weight, and power (SWaP) of the UAV radio device can be limited by design constraints of the vehicle and compete with requirements imposed by other on-board systems. SWaP constraints may also limit on-board computing capability which can limit the complexity of the algorithm used

to control communications. Next, unless a vertical flight aircraft is used, the antenna position and orientation, which affects channel quality [49], is subject to vehicle dynamics. Finally, interestingly some light weight materials commonly used for UAVs, such as carbon fiber, have conductive properties which shield radio signals and can cause loss of link [50].

1.3.2 Decentralized Planning Challenges

The goal of the mission planning algorithm in this framework is to allocate the right UAV to execute the right task at the right time. Because there are multiple UAVs, multiple tasks, and many different orders in which to execute them, the interdependencies result in a large number of possible assignments. This combinatorial problem of finding the “optimal” assignment quickly becomes difficult to solve computationally in a reasonable amount of time. The problem is further aggravated when considering communication requirements. Each configuration of agents results in a different network topology which may require different relay support configurations to provide extended range or relieve bottlenecks. This creates inter-task coupling and coupling with assigning agents to support the network. In addition to these complications, the mission and environment vary dynamically, and it is therefore paramount that the planner quickly compute an effective task assignment to adapt to these changes.

Whether a decentralized algorithm is planning task assignment, vehicle motion, or even information routing, the goal is for the decentralized agents to quickly converge to a common solution to execute a coordinated plan. Even with proper system design and implementation, the communication network will still have limitations which can significantly degrade performance in convergence. Command delays, however short, may cause a formation of UAVs to perform inefficiently or *churn* (effectively become unstable), which can have disastrous effects such as vehicle collisions [51]. Delayed and dropped messages sent to planning agents, whether centralized or decentralized, can cause inconsistencies in situational awareness, and, as a result, flawed planning [52]. Similarly, message delays in decentralized planning may prevent agents from

reaching consensus on a plan [53]. These limitations motivate the need for robust algorithms which converge as quickly as possible.

1.4 Thesis Contributions

This thesis presents five notable contributions to the multi-UAV cooperative control body of knowledge:

1. A detailed survey of recent studies in network communication control in multi-unmanned vehicle systems is included to provide a broad overview of different techniques developed. The survey categorizes the different types of problems under investigation and provides the relevant network communication technical background needed to get acquainted with this relatively new field of study.
2. A framework is presented to enable multi-UAV teams to cooperatively meet several communication constraints simultaneously including data-rate and messaging bit-error-rate. This further includes methods to dynamically estimate uncertain networking environments and adapt planning strategies to changes in real-time. This portion of the work includes multiple Monte Carlo simulations to characterize the cooperative behavior of the system.
3. A Linear Program (LP) formulated using network optimization theory is presented to compute optimal information routing in a wireless network given realistic channel constraints. This routing algorithm, while unrealistic in its assumptions, is useful to provide relevant upper-bound performance measures for these types of studies without losing generality due to intricacies of individual routing protocols.
4. In an effort to show real world relevance, a significant focus of this thesis involves flight testing the algorithms developed. The distributed dynamic mission planning framework to control network communications is implemented with a team of three UAVs. The results of this unique system operating outdoors are presented.

5. A related research effort was conducted which involved implementing a decentralized heterogeneous multi-UAV system operated under human supervisory control. Four heterogeneous UAVs were controlled at the mission level by a single human operator to execute a complex operationally relevant mission. The lessons learned from this unique outdoor flight test campaign are included in this thesis [54].

1.5 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 includes technical background in wireless communications relevant to this thesis and provides a comprehensive survey of recently developed network communication control techniques for multi-unmanned vehicle systems. Chapter 3 describes principles of decentralized dynamic mission planning as well as algorithms developed in previous work which laid the foundation for the framework presented in this thesis. Chapter 4 then describes the developed framework and algorithms which enable agents to perform task allocation considering realistic networking dynamics and allow them to adaptively plan in uncertain communication environments. This chapter also includes the results of several Monte Carlo simulation experiments used to characterize the system performance. Chapter 5 presents the implementation, execution, and results of several outdoor flight tests in which multi-UAV teams cooperatively plan using the developed algorithms. Finally Chapter 6 describes a complex heterogeneous decentralized multi-UAV system under human supervision which was developed and flight-tested outdoors. The lessons learned from these tests reinforce findings in the previous chapters and are relevant to the overall multi-UAV cooperative control community.

Chapter 2

Network Communication in Multi-Unmanned Vehicle Systems

2.1 Network Communication Background

This section presents fundamental principles in wireless network communications that are relevant to multi-UAV systems and serve as the foundation for communication control elements built upon in Chapter 4.

2.1.1 System and Network Architectures

The design and control of the communication network depends on the system architecture and data transfer requirements. For instance, traditional cellular phone networks form a hub and spoke model between a fixed base station and mobile users in its area. Telephone voice data requires minimal delay to be effective, but does not need high data-rates and has some error tolerance. Conversely, data networks for Internet downloads require high and bursty (sporadic) data-rates and dynamic bandwidth allocation between a gateway, routers, and end users. These networks can tolerate some delays, but should be relatively error free [11].

A multi-UAV system network typically needs to support two types of data. The first type consists of command and control messages including state information

(telemetry), observations of the world (e.g. estimated target location), and control data (e.g. waypoints or task allocation). These messages have low bandwidth requirements, but must be exchanged with minimal delay and error for effective team coordination. Second, information gathered from sensors such as video, still images, atmospheric samples, and other types of remote sensing data may also need to be transmitted to designated processing, exploitation, and dissemination (PED) centers [55]. Again, communication requirements depend on the data type, but live-video, for instance, requires minimal delay, high data-rates, and has some fault tolerance [11]. Both data types are simultaneously considered in the multi-UAV control framework proposed in this thesis (see Chapter 4).

A multi-UAV system architecture built on centralized control, where vehicles remain within transmission range of the command and control node, can operate using a hub and spoke network model, similar to a cellular system (see Fig. 2-1(a)). Some implementations can even extend command and control range by using a low data-rate satellite communication system [8]. However, many of the applications envisioned for multi-UAV teams require greater topology flexibility, with inter-node information exchanges and relaying (see Fig. 2-1(b)). This architecture closely follows concepts of ad-hoc networks which have received significant interest in many disciplines [43], and is the architecture modeled in this thesis.

2.1.2 Wireless Channels

In wireless networks, data is exchanged over communication channels by sending information encoded and modulated from a transmitter, over the air using a signal, to a receiver which demodulates and decodes to process the data. The strength of the received signal power compared to surrounding noise, or Signal to Noise Ratio (SNR), γ , is a critical parameter in assessing the quality of the link. This value is

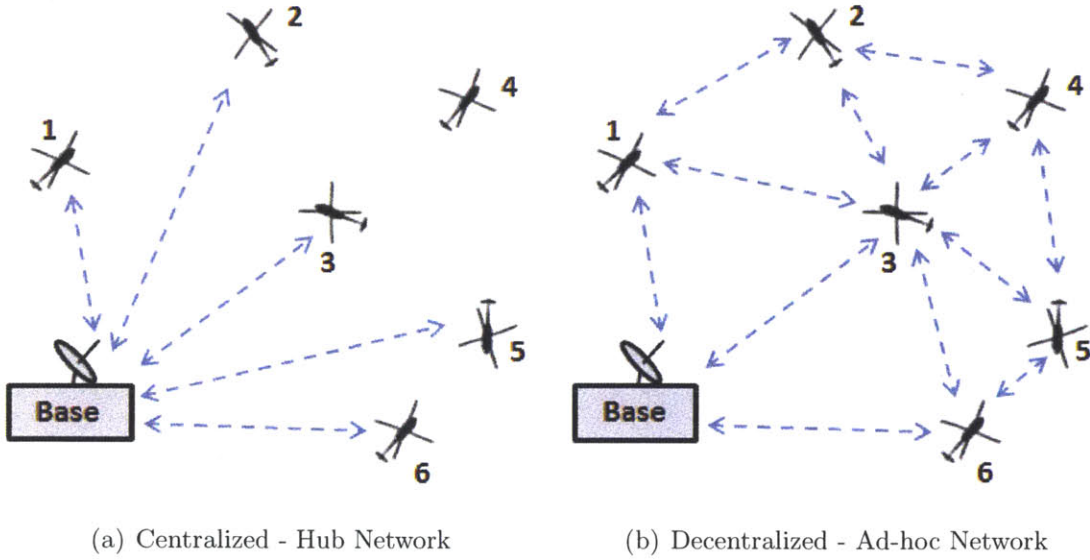


Figure 2-1: Centralized and decentralized network architectures for multi-UV systems

commonly modeled as:

$$\text{SNR: } \gamma = \frac{gP}{N_o W} \quad (2.1)$$

$$g = K \left(\frac{d_o}{d} \right)^\alpha \quad (2.2)$$

where P is the transmission power, $\frac{N_o}{2}$ is the power spectral density of the environment noise (assuming Additive White Gaussian Noise), and W is the bandwidth of the signal. The channel gain g is generally modeled as Eq. (2.2) in its most simple form, in which K is a gain based on equipment characteristics, d is the Euclidean distance between the transmitter and receiver, and α is the path loss exponent which equals 2 in free space, and up to 6 in environments congested with obstacles [10, 43]. This relationship represents the path loss dynamics experienced in wireless channels, in which γ decreases with increasing distance.

The next effect to consider is shadowing of the signal due to obstacles in the line-of-sight. A practical way to model shadowing in uncertain environments is to vary γ according to a lognormal distribution with variance σ^2 which depends on the density of obstacles in the environment [10]. A convenient method to express this is

to convert the channel gain in Eq. (2.2) into decibels (g_{dB}), where shadowing now becomes a Gaussian distributed random variable centered on the path loss gain:

$$g_{dB} = K_{dB} - 10\alpha \log \left(\frac{d}{d_o} \right) - \mathcal{N}(0, \sigma_{dB}^2) \quad (2.3)$$

The distribution has been shown to be spatially correlated to X_c , the typical size of obstacles in the area [10]. The spatial covariance $A(x)$ in the distribution between two points separated by distance x can be expressed as

$$A(x) = \sigma_{dB}^2 e^{-x/X_c} \quad (2.4)$$

This expression can be used to predict the amount of time a moving vehicle may undergo deep shadowing fades, or determine how to reposition the vehicle to exit a shadowed area [28, 44].

Multipath fading, which occurs because of constructive and destructive interference from scatterings and reflections of the signal by the obstacles, can be modeled using a non-stationary random process centered on channel gain g subject to path loss and shadowing in Eq. (2.3). Models using Rayleigh, Rician, or Nakagami distributions with a tunable fading parameter were demonstrated to follow trends in empirical measurements of this dynamic [10, 44]. Multipath fading values can be added to the g_{dB} expression in Eq. (2.3), and then used to predict the overall SNR using Eq. (2.1). This dynamic can vary significantly over small distances and is often modeled without spatial correlation for simplicity [44]. Because shadowing and fading are additive random variables in dB which are both present in channel measurements, this thesis combines both properties into a single random variable without spatial correlation as expressed in Eq. (2.3).

Many field experiments have studied the dynamics of link performance in greater detail. Research in [46] used data collected from aerial links to measure an additional temporal correlation factor in the above dynamics. A study in [49] observed that link qualities vary depending on the relative orientations between the transmitter and receiver, even when using omni-directional radio modules. Multiple antenna

configurations were similarly used in [56, 57] to improve the overall reception for the vehicle through diversity.

The relationships in Eq. ((2.1)–(2.4)) model the dynamics of a single wireless channel. In a multi-user network, multiple agents simultaneously communicating on the same or neighboring channels may interfere with each other. In this case, SNR is often rewritten as the Signal to Interference and Noise Ratio (SINR) represented by γ_{ij} for the channel from node i to j [58],

$$\text{SINR: } \gamma_{ij} = \frac{g_{ij}P_{ij}}{(N_oW)_j + \frac{1}{\rho} \sum_{(m,n) \neq (i,j)} g_{mj}P_{mn}} \quad (2.5)$$

Here, the interference caused by other nodes m transmitting to n adds to the noise at receiver j in the denominator. While some of these effects can be mitigated through signal code processing ($\frac{1}{\rho}$), a common method of deconfliction is to divide channel usage by time (TDMA) where $\rho = 1$ [58].

The SNR in Eq. (2.1) or SINR in Eq. (2.5) are fundamental indicators of wireless link performance and can be used to predict operationally relevant data transmission metrics. First, the theoretical data-rate *capacity* u of the channel in *bits/sec* or *bps* can be evaluated using Shannon’s Equation:

$$u_{ij} = W_{ij} \log_2(1 + \gamma_{ij}) \quad (2.6)$$

While this value is an upper-bound, and actual data transmissions are often set to significantly lower rates due to fading [10], it is a useful indicator in applications where information throughput must be considered. Since remotely sensed data, such as live video streaming require high data-rates (see Sect. 2.1.1), this metric is employed in the framework presented in Chapter 4 to plan network support to route data to the base as described in the motivating scenario.

Another important performance metric is the *bit-error-rate* (BER), or the probability an information bit will be dropped, which can require packet retransmissions and inefficient data flow. BER decreases with increasing γ , and the relationship

depends on the specific modulation scheme, and can also be increased by reducing transmission data rate. Equation (2.7) represents a general relationship for BER where $Q(z)$ is the probability that a zero mean normally distributed variable X with variance 1 exceeds z [10]. Because command and control messages have low data-rate requirements, but require high transmission precision, BER is adopted in this thesis as a metric to plan inter-agent communication requirements.

$$\text{BER}_{ij} = Q(\sqrt{2\gamma}) \quad (2.7)$$

where

$$Q(z) = p(X \geq z) = \int_z^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

Many of the communication network control methods discussed in this chapter have for objective to optimize or meet some threshold values for capacity or BER. The models listed in this section highlight the challenges associated with predicting and controlling wireless channel performance due to dynamic uncertainties in fading and coupling with other agents using the network.

2.1.3 Network Topology and Consensus

Another relevant challenge, especially in ad-hoc networks, involves optimizing the network topology by determining which nodes need to form interconnections to achieve the desired connectivity for information flow. The network topology can be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a set of n nodes \mathcal{V} and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ indicating connectivity between nodes. The graph can be represented using an *adjacency matrix*, A , with $n \times n$ elements a_{ij} , where $a_{ij} = 1$ if i can communicate with j , and $a_{ij} = 0$ otherwise. Each row of the adjacency matrix can be read directly to identify the set of neighbors for every agent. In some formulations $a_{ij} = w$, where $w \in [0, 1]$, depending on the quality of the link and topology properties. The *neighborhood* of agent i , $N_i = \{j \in V : a_{ij} \neq 0\}$, is defined as the set of agents that agent i can communicate with directly [59]. For instance, in the network shown in Fig. 2-1 (b), the neighborhood of UAV4 is $N_4 = \{2, 3, 5\}$. Theoretical formulations have established

bounds on how many agents each node should be connected to for the network to become asymptotically disconnected or connected [47]. For this reason, many multi-vehicle systems control the network so that each agent remains connected to a specific number of neighbors [13, 23, 24, 37, 38, 41].

Analyzing the network topology provides useful insights into how information propagates throughout the network and how the team of agents reaches consensus on quantities of interest such as plans, parameters and situational awareness. The stability and convergence properties of most consensus algorithms can be predicted by analyzing the spectral properties of the network graph *Laplacian*, \mathcal{L} , for a given network structure [59]. The Laplacian is another related matrix representation of the network topology, and is formulated as

$$\mathcal{L} = D - A \tag{2.8}$$

using the adjacency matrix A and the *degree matrix*, $D = \text{diag}(d_1, \dots, d_n)$, with elements $d_i = \sum_{j \in N_i} a_{ij}$, representing the out-going connections of each agent (sum over its neighbors). The maximum degree of the graph, $\Delta = \max_i d_i$, is useful in determining spectral properties of the network Laplacian. In particular, all of the eigenvalues of the Laplacian \mathcal{L} in the complex plane are bounded by a closed disk of radius Δ centered at $\Delta + 0j$. Assuming the graph is undirected, where $a_{ij} = a_{ji}$, the Laplacian \mathcal{L} is symmetric, and therefore its eigenvalues lie on the real line within the set $[0, 2\Delta]$. These can be ordered sequentially as

$$0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2\Delta. \tag{2.9}$$

The first (smallest) eigenvalue of \mathcal{L} is always zero ($\lambda_1 = 0$), since every row sum is $\sum_j l_{ij} = 0$, and is known as the trivial eigenvalue. The second eigenvalue, λ_2 , is known as the *algebraic connectivity* or *Fiedler value* of the graph, and is always positive for *strongly connected* graphs where a single or multi-hop path exists from any agent to every other agent. The algebraic connectivity λ_2 determines the speed of convergence for most consensus algorithms, and, as such, many multi-vehicle applications

attempt to maximize this value by controlling the network topology [13, 17, 32, 36] (see Sect. 2.3.1 and 2.3.2). The last eigenvalue, λ_n , is related to stability guarantees for reaching an equilibrium in time-delayed networks, providing associated bounds on the maximum allowable delays [59].

It must be noted that graph theoretic methods are simplifications of multi-UAV network representations, since connectivity is often assumed to be binary and links are considered active if agents are within a threshold distance of each other [43]. These methods usually do not consider realistic wireless channel dynamics as described in the previous subsection, or some of the complexities in information routing discussed next. Nevertheless, they do provide helpful mathematical interpretations of the potential performance of the network, and are useful in guiding the system design to achieve configurations more suitable for information exchange [17].

2.1.4 Information Routing

Since envisioned multi-UAV team applications require similar flexibility to ad-hoc wireless network architectures, network routing becomes a significant challenge that must be considered in controlling the system. Given the wireless channel properties and network topology, information must be routed along links with sufficient capacities, and arrive at its destination with minimum error (BER) and delay, as dictated by data requirements. Routing is an active area of research in the wireless networking community [60]. If the network involves multiple channels on different frequencies, as is often the case, the problem extends to finding the optimal route and channel allocation [61]. Strategies range from decentralized, highly adaptable methods, to more centralized optimized schemes [10]. Understanding how a particular network system routes information is necessary to adequately control communications in a team of UAVs, as it may affect where agents are positioned to provide a suitable topology.

Several industry standards (802.11s, 802.15, 802.16) drive the design and operation of mesh networking modules commonly used in ad-hoc networks [61, 62]. These standards rely on decentralized routing mechanisms. A commonly used protocol is Ad-hoc On-Demand Distance Vector (AODV) [63]. In AODV, a node which needs

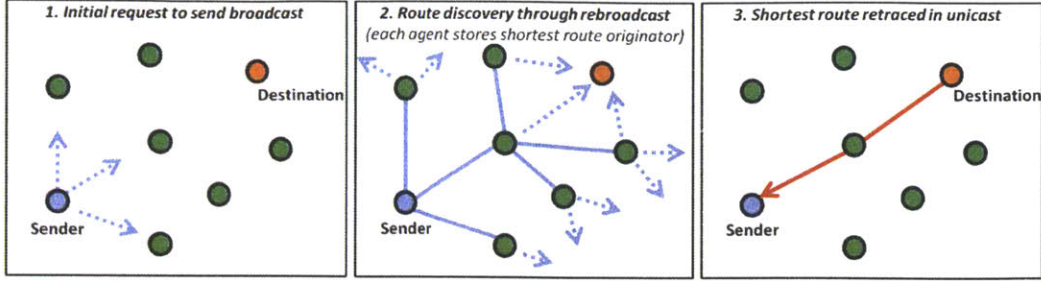


Figure 2-2: Ad-hoc On-Demand Distance Vector (AODV) routing protocol

to transmit broadcasts a “request to send” message which identifies the destination node. Neighbors receive the message, compute a cost metric (discussed later), and rebroadcast the message. The process continues, where cost is computed at every hop, until the destination is reached. Each node stores its lowest cost originator. Once a designated amount of time has elapsed, the destination node sends a unicast response to its lowest cost originator, which similarly relays the response to backtrack the lowest cost route. This process is illustrated in Fig. 2-2.

Other similar routing techniques include Dynamic Source Routing (DSR) [64], Optimized Link State Routing (OLSR) [65], and B.A.T.M.A.N [66]. Different metrics are implemented depending on the objectives, which include lowest-hop count, Expected Transmission Count (ETX in Eq. (2.10) [67, 68]), and Airtime metric (C_{Aij} in Eq. (2.11) [69]). Here, p_{pkt} is the packet loss ratio (based on BER), O is overhead, B_t is the test-frame length, and f is the information flow data-rate (based on capacity).

$$\text{ETX}_{ij} = \frac{1}{1 - p_{pkt}} \quad (2.10)$$

$$C_{Aij} = \left(O + \frac{B_t}{f} \right) \frac{1}{1 - p_{pkt}} \quad (2.11)$$

The protocols listed above are implemented in practice, for instance on Zigbee modules common in robotics research [70], because of their rapid response to changes in wireless channel performance and topology. A significant drawback to them, however, is that information from different nodes is routed mostly independently of other nodes, and tends to use the same high performance routes. This can lead to congestion in high data-rate applications, and unfairness in node usage [67]. Researchers

have proposed different network optimization schemes to address this, which invoke conservation of information flow at each node and link utilization constraints. For instance, authors in [71] optimize routing fairness by considering interference between nodes and using electrical flow analysis. Similarly, convex optimization methods have optimized routing through simultaneous link scheduling and transmitter power allocation [72, 73]. While these schemes produce better routing solutions, they mostly consider static networks, and would require frequent recomputation in a dynamic environment to adapt to realistic changes in the topology [10].

The framework proposed in this thesis explores the use of two protocols. First, AODV is used to model real-world implemented routing dynamics to show feasibility of the proposed framework. Second, an optimized routing protocol (with fixed power transmission) formulated as a Linear Program (LP) is used to provide performance bounds of the general framework. This is to show performance bounds of the system which can be easily adapted to use better protocols than AODV as they are developed (see Sect. 4.3).

2.2 Why Communications Matter in a Team

This section highlights some of the operational problems which can arise in a multi-vehicle system to stress the need for proper network control. For multi-UAV teams to cooperatively execute a set of objectives, the communication network must exchange command and control messages, and when necessary, remotely sensed data. The inability to communicate sensed data, such as live video, to the designated processing centers may render the multi-UAV system ineffective for its mission [74]. Less intuitively, failure to properly exchange command and control messages can lead to dangerous system failures or unintended consequences. This is particularly true in decentralized systems, where these messages enable vehicle control and team decision making. Inadequate team control can lead to formation instability where one or multiple agents increasingly diverge from the desired behavior, and risk colliding or separating from the team. Poor decision making may prevent the team from reaching

consensus on a plan and remain idle without accomplishing anything, or worse, take action on an undesirable plan leading to unwanted consequences.

2.2.1 Communications for Motion Control

The feedback control process uses sensed feedback information with a model of the system dynamics to select a control input. If feedback information is delayed or corrupted due to communication degradations, control decisions may be flawed. For instance, in military MQ-1 Predator procedures, a pilot remotely controls the UAV manually using live forward looking video on the nose of the aircraft as reference. Delays in video feedback and control commands from link latencies have resulted in pilot-induced-oscillations (PIO), an instability from poor feedback control which resulted in numerous aircraft mishaps [75]. The same phenomenon can occur in autonomously controlled cooperative teams of UAVs.

In general, Network Controlled Systems (NCS) require a proper balance of control and feedback update rates: too slow can prevent the system from being controlled sufficiently well, but too fast can excite system dynamics resulting in instabilities [51]. The stability of the system also depends on control implementation and feedback filtering techniques, as described in detail in [76]. In “leader-follower” formations, each vehicle selects control inputs based on the dynamics of the vehicle(s) directly preceding it. Tracking errors can propagate down the formation, and under certain conditions exceed control authority, leading to instability, even with no delay in information sharing [77]. In “string” formations, vehicles are slotted a separate cycle time to communicate to neighbors which propagates an information delay down the chain. If the objective is to base control inputs on the leader’s dynamics, even infinitely small delay propagations can lead to string instabilities [78]. The formation can, however, be stabilized given greater communication delays if the vehicles delay their control response, and all apply inputs in near unison.

Similarly, in [51, 79] the effects of network delays on formation stability of a team of decentralized UAVs is explored in indoor flight-testing. Vehicles can communicate only with their neighbors, and are also slotted to broadcast knowledge of vehicle states

at different times. The authors show that formation stability depends on the structures of the communication graph \mathcal{G} , the formation control graph implementation, communication cycle time, and control gains.

Communication delays can also be problematic in multi-UAV path planning for collision avoidance. In Chapter 6, decentralized vehicles continuously replan the next steps of their receding horizon paths (not in formation). UAVs deconflict by not planning crossing paths to go around each other. Short network delays and rapidly changing dynamics cause the vehicles to sometimes plan using outdated information. On occasion, two UAVs approaching each other head-on may alternate in replanning deconflicting routes left or right multiple times. This phenomenon, known as “churning”, is similar to the common awkward situation where two people mutually block each other in a hallway or sidewalk, and causes delays, hazards, and system performance degradation.

2.2.2 Communications for Decision Making

In order for multiple UAVs to cooperate in a mission, the team as a whole must plan and make decisions collectively. Therefore the objective in cooperative decision making is for the networked team to reach an agreement on the tasks, plans, and actions required to execute the mission. Different strategies exist to achieve *conflict-free* plans, which require information exchange through the network. The dynamics and uncertainties of the communication network have several implications on decision making strategies: (1) bandwidth limitations can restrict the content and frequency of planning messages, (2) even small network delays can cause agents to plan asynchronously making consensus difficult to achieve, and (3) network dropouts may prevent agents from participating in team decision making which can impact plan execution.

Because network bandwidth is a limited resource, cooperative decision making methods need to limit the content of planning messages, efficiently select which pieces of information need to be shared with which members, and overall reach consensus with as few messages as possible to mitigate delays and conflicts. If the system op-

erates under centralized control, every UAV needs to communicate every required bit of situational awareness information to a central leader (e.g. the base station). This strategy minimizes the number of messages required to reach consensus, but may require large amounts of low level planning data to be communicated to the base, and can cause the system to be slow to react in a rapidly changing environment due to network delays. To overcome this, decentralized planning strategies provide individual agents with greater decision making capability, where information can be shared more selectively, and agents can locally decide on courses of action based on their own situational awareness, thus improving reaction time. Now, however, consensus on plans must be explicitly reached by exchanging messages over the network. A common distributed strategy, referred to as *implicit coordination*, consists of agents independently solving the entire centralized planning problem for all agents, and carrying out actions assigned to them. This requires agents to reach consensus on individual situational awareness and planning objectives prior to planning [80–82], which may require large exchanges of data across the network, especially if inconsistencies exist [83]. Another popular planning method involves using *distributed auction algorithms*, where agents individually plan for themselves and consensus is reached through an auction mechanism [84–86].

Another challenging effect in distributed planning algorithms is their reliance on synchronous message passing in order to guarantee convergence. This is difficult to enforce in dynamic environments where the number of tasks or agents in the network may be changing and the network is subjected to the uncertainties and dynamics described in Sect. 2.1.2. When messages are asynchronous, time-delays are not usually constant and messages may be received out of order, which impacts typical consensus algorithms since old information may be processed as new, thus hindering consensus. Robustness to these effects requires new consensus protocols or plan deconfliction rules that can recognize and process out-of-order messages [53].

2.3 Communication Network Control Methods

Two types of problems are commonly studied in the context of multi-unmanned vehicle network communication control. The first consists of controlling a multi-agent system to achieve a defined primary mission objective, and constraining agent motion in order to maintain communication requirements. The second consists of controlling a set of agents designated as communication relays to support an underlying network. Here, their primary objective is to explicitly support the network. Both types of problems are explored in this section, and are then compared to the method proposed in this thesis which uses task assignment to control communications.

2.3.1 Control of Communications Through Motion Planning

A common method to control network communications in a team of multiple UAVs is to plan the motion of the vehicles in such a way that the resulting topology supports the exchange of data required by the system. The general problem consists of controlling a set of N team members to achieve the mission objective, and maintain the required state of connectivity. For mathematical simplification, vehicles are often considered to be point nodes with first order dynamics, $\dot{x}_i(t) = u(t)$, where $u(t)$ is the control input at time t , and $x_i(t)$ is the position of agent i . Some formulations go further and extend control to second order dynamics as in Eq. (2.12) [13] where v is velocity, whereas others include greater detail in modeling actual vehicle kinematics [34]

$$\dot{x}_i(t) = v_i(t) \quad \text{and} \quad \dot{v}_i(t) = u(t). \quad (2.12)$$

Network communication requirements vary from maintaining connectivity with a specific number of neighbors, to establishing routes with sufficient throughput to send data to a base station. One of the earliest studies where motion control was employed to maintain connectivity consisted of a team of distributed robots with the task of meeting at an undefined point (rendezvous problem). The strategy involved each robot independently moving a step amount toward the geometric center of the area outlined by its perceived neighbors [12]. This process was repeated iteratively until

robots converged on a central location. Since then, methods have expanded to include graph theoretic techniques, continuous control through potential fields, reactive control strategies, and adaptive techniques based on learning from the environment. Many of the efforts described here were focused on ground robotic systems, but are relevant to aerial applications as well.

Sect. 2.1.3 introduced the network graph \mathcal{G} and the *algebraic connectivity* metric λ_2 , the second eigenvalue of the graph Laplacian \mathcal{L} . Several studies have formulated \mathcal{L} by assigning weights $w \in [0, 1]$ to each inter-agent link based on relative distance, with bounds $w_{ij} = 1$ if $d_{ij} \leq d_o$ where the link is strong, and $w_{ij} = 0$ if $d_{ij} \geq d_{\max}$ where the signal is considered too weak to connect [13]. Connectivity objectives in multi-agent control problems often consist of meeting some threshold λ_2 or even maximizing its value. Initially, discrete methods were proposed to maximize λ_2 using iterative control inputs for each agent using greedy algorithms in both centralized [14] and decentralized frameworks [15]. These ensure that the team drives to connected configurations throughout the mission. Continuous control methods to maximize λ_2 using properties of \mathcal{L} were then formulated in [16] using potential fields in a centralized framework. Potential fields, which are commonly used in multi-vehicle control applications, are virtual energy fields that lead to attractive or repelling control inputs formulated as:

$$u_i(t) = -k\nabla_i\phi_i - \sum_{j \neq i} \nabla_i\psi_{ij} \quad (2.13)$$

$$\psi_{ij} = \frac{1}{\|x_{ij}\|^2} + \frac{1}{d_{\max}^2 - \|x_{ij}\|^2} \quad (2.14)$$

One benefit of this formulation is that it allows multiple, sometimes competing, control objectives to be considered. Here ψ_{ij} is designed according to the desired connectivity dynamics between i and j , as illustrated by the simple function in Eq. (2.14) which keeps the nodes within communication range and prevents them from colliding [17]. ϕ_i can be any other steering function, used for example for obstacle avoidance or target tracking [18]. A decentralized version of this framework was proposed in [19] using a power iteration algorithm for each agent to compute its local λ_2 using

information from its neighbors, and then estimate the global λ_2 to generate potential fields. Potential fields have also been used to control multi-vehicle systems for other connectivity requirements, using both graph theoretic formulations and considering realistic channel dynamics and measurements. For instance [20] and [21] used bounded forces to minimize distance with connected neighbors for a team to rendezvous or assemble in a formation. One study considered Rician fading in wireless links to control multiple robots to track a target and maintain connectivity with the base [22]. A primal-dual optimization algorithm was used to control agents through a deployment objective while subjected to stochastic fading effects [87]. Finally, in [41] potential fields controlled a team of eight UAVs to form a moving aerial communication network. Forces were created based on the deployment objective of maximizing area coverage over a moving point and multiple connectivity requirements: maintain connection with a designated number of neighbors with prescribed SNR values over fading links.

Instead of controlling motion for connectivity continuously, another strategy is to allow agents to navigate freely according to mission objectives and take action only when connectivity is threatened. One method involves each robot iteratively planning how far it can travel at every step before losing connectivity with a required minimum number of neighbors [23]. Other studies were motivated by the difficulty of predicting wireless link performance with enough accuracy to make control decisions in a realistic environment. In these cases, each robot travels according to deployment objectives, using potential fields or random motion until the measured signal strength with a set number k of neighbors drops below a designated threshold. At this point, the robot is commanded to halt [24] or even backtrack toward its nearest neighbor [18], and can only resume deployment objectives if communications requirements are again met. In these strategies, the required number of neighbors k is part of the design trade. A higher value increases network robustness to unpredicted disconnects, but constrains the configuration of agents to be tighter and therefore covers less surface area [24]. When warranted, the team may benefit from two agents disconnecting to add flexibility to the formation (so long as the graph \mathcal{G} does not disconnect). For

such cases, the study in [88] employs a market based algorithm for agents to come to consensus on which link to remove to meet formation objectives. In this process, robots place bids on the link most at risk with bid value based on their number of neighbors. The bids are propagated through the network, and the highest bid link is allowed to be deleted. The framework only allows one link to be deleted at a time to ensure the graph stays connected, and the approach introduces methods to handle asynchronous planning issues (see Sect. 2.2.2). Finally, certain scenarios involve agents taking on tasks beyond the reach of communications, where the only option is to temporarily disconnect the network. The authors in [25] therefore define an implicit coordination approach which is used to find optimal agent paths to execute the mission, but reach a point of full connectivity at some specific time interval.

Many recent robotic efforts have turned to adaptive and learning techniques to adjust their behavior and overcome unpredicted changes in the environment [3]. Predicting actual wireless channel performance in dynamic environments is difficult. As such, several approaches have used measurements in the field to adapt the motion planning strategy. One basic approach is to take measurements of the environment *a priori* and construct a “radio-map” used to plan constraints on relative node positions to stay connected [18]. However, it may not be practical to perform this type of pre-mission survey as access to the location may be denied, and the environment may change over time due to different obstacles or other emitting devices. For this reason, several studies propose probabilistic methods based on the spatial correlation of channel fading to select locations to transmit. For instance, in [26] a single mobile agent tracks a moving target and must send collected data to a fixed base. Current measures of the SNR are used with link dynamics previously described to adaptively schedule stops where the channel is predicted to be strong to transmit information rather than waste energy transmitting during fades. Bounds on the dynamics are derived to ensure system stability and prevent the agent from falling behind the target. In a similar scenario, a single agent must repeatedly visit multiple areas and send information to the base. Here the planner sequentially solves for the optimal visit path using a Mixed-Integer Linear Program (MILP), and budgets the power needed

to transmit given the environment and path using a nonlinear program [27] for energy conservation. Finally in [28], a team of multiple UAVs learn link dynamics online to plan motion cooperatively and track a target. The measured SNR is used to learn the communication environment and adapt the planning behavior. Estimation of the communication environment inspired by this last method is implemented in this thesis and used to adapt task allocation planning behaviors to changing communication environments.

Motion planning to control network communications is a relatively new research topic in robotics, and as such there have only been a few implementations of these principles using actual vehicles. Most of the studies described previously in this section were conducted in simulation. However, the study in [18] involves an outdoor experiment with four UGVs relaying sensed information back to the base (similar to the motivating scenario in this thesis). The control strategy involves both potential fields and reactive control to prevent disconnects, and the study overall highlights that signal strengths and throughputs vary significantly during the mission due to fading and multi-user interference.

2.3.2 Control of Communications Using Mobile Relays

Due to their rapid and flexible deployment capabilities, persistence, ability to fly above obstacles, and relative low cost, UAVs have generated significant interest to serve as communication relays. Relays can be used to extend the range of a network, add capacity for increased throughput, and increase the robustness of the system to failure. Large scale military UAVs are already delivering wide area communication connectivity to ground troops in mountainous areas [89], and similar smaller tactical concepts have been proposed for disaster area recovery [90], wild land fire fighting [91], and to connect scattered wireless sensors and mobile ad-hoc networks [29]. Many of the communication control methods described in the previous section apply to relay deployment and control, but in this case the underlying network to be supported may not be controllable and can be dynamic. Control objectives typically involve deploying relay agents to optimized locations to support data exchange requirements

at the lowest resource cost possible. Two classes of problems are provided in this section. First, methods are investigated to optimize the control of a chain of relays linking two mobile end nodes. Second, the problem of optimal relay deployment to support a larger, typically fixed network, is considered. In all cases, controlling the communication network involves considering the dynamics of wireless links, the topology, and information routing.

Several applications have called for a chain of relay nodes to extend communications through multiple hops from one end of the chain to the other. In this case, once the chain forms, the relationship between pairs of nodes remains constant (same two neighbors), and information routing is known since it simply follows the nodes in the chain by default. A common application of this concept involves maintaining communications between a moving agent and its base station. This problem has been examined in complex obstructed environments where the objective is to optimally navigate and position N relays to establish multi-hop line-of-sight with the base. Optimal deployment paths can be generated in a centralized framework using a Mixed-Integer Linear Program [30] or a Gauss pseudospectral solver initialized using a Rapidly-Exploring Random Tree (RRT) solution [31]. A suboptimal decentralized receding horizon planner can also be used to provide faster real-time control to repeatedly “redeploy” relays as the mission progresses [31]. In a similar problem, a chain of UGV relays must connect a mobile node to a base and optimize their formation to minimize the end-to-end *bit-error-rate* (BER) (see Eq. (2.7)). Here, wireless links are subjected to path loss, shadowing, and multipath fading [32, 33]. The resulting optimization can be written as:

$$\max \mathcal{J}(x) = \sum_{i=2}^N \ln(1 - \text{BER}_{i-1,i}) \quad (2.15)$$

which is reformulated as a set of decentralized potential fields used for vehicle control as in Eq. (2.14). In [32], the formulation does not assume deterministic link dynamics, since actual channel qualities can vary significantly with predicted models. It instead employs a probabilistic framework with least squares regression to estimate the value

of the path loss exponent α in Eq. (2.2) and shadowing variance and correlation from Eq. (2.4) from several SNR measurements. These estimates are constantly updated to plan motion in order to optimize the chain BER. Since multipath fading is assumed to be spatially uncorrelated and varies quickly over small distances, once the relay reaches its destination it “jitters” around its position to further reduce the quantity $\text{BER}_{i-1,i}$. With a slight modification to the problem formulation, the objective is changed to provide a chain configuration which meets a threshold end-to-end BER but minimizes relay energy costs. This allows motion cost and transmission costs (through variable power transmission) to be traded-off in the optimization, with tunable weights associated with each to favor one over the other [32].

The authors in [34, 35] tackle the problem of forming a chain of relays using actual fixed wing UAVs flying over a mountainous terrain. In this study the objective is to maximize the end-to-end *capacity* u of the chain found from Eq. (2.6), or in other words maximize the minimum link capacity in the chain. Two complicating but realistic factors in this work include: (1) fixed wing aircraft have velocity limits $0 < v_{\min} < v < v_{\max}$ and must loiter in orbits also constrained by ϕ_{\max} bank angle, and (2) the authors chose to control the chain based on SNR signal strength measurements only instead of relying on models considered too inaccurate for a dynamic environment. An extremal seeking algorithm is implemented to hone each UAV on the heading which maximizes γ with its neighbors and a Lyapunov Guidance Vector Field (LGVF) [34] drives the UAVs to an orbital pattern about a center point. Flight test results demonstrate the ability to control an aircraft to climb the SNR gradient and improve capacity using signal strength measurements only. However, RF fluctuations encountered are detrimental to the chaining algorithm and at times cause the measured γ to actually increase while flying away from a neighbor.

In other types of applications, UAVs (or other mobile agents) serving as relays can be deployed to provide general connectivity support to an existing network consisting of many nodes. The problem here is to determine how best to deploy each relay, and which nodes they should form links with to satisfy the data requirements of the network. Objectives include maximizing the performance of the supported network,

or minimizing the number of relays deployed to meet communication thresholds. These problems are often examined in the context of UAVs relaying information from poorly connected ground nodes, or for wireless sensor network (WSN) nodes with connectivity and energy limitations sending data to a base station.

Network graph theoretic processes described in Sect. 2.1.3 have also been explored in these problems. One example consists of maximizing the lifetime of a fixed WSN collecting data which depletes its battery over time as the energy is used to transmit information [36]. Here K relays are available to be deployed and redeployed to support the network. The graph Laplacian \mathcal{L} in Eq. (2.8) is composed of weights w_{ij} in its adjacency matrix A based on the transmitter power needed to maintain a threshold BER across the link. This formulation assumes all starting nodes have equal battery life, and therefore the algebraic connectivity of the graph λ_2 can be used as a metric of remaining network life. A centralized Semi-Definite-Programming (SDP) algorithm optimizes the placement of the K relays to maximize λ_2 over the network. It discretizes the area into cells, finds the best combination of placements for the K relays at the center of these cells, and then rediscrretizes selected cells to refine relay placement over several iterations.

In a similar problem, WSN nodes compute their k -*redundancy* in a distributed framework using neighbor discovery messages [37]. The k_i^{red} is the number of links that need to be removed to disconnect any two neighbors of i , and is a measure of the importance of that node in global connectivity. Here if any k_i^{red} falls below some threshold because of a change in topology, the network requests a relay which is positioned using several different optimization schemes to meet connectivity requirements. Finally, the study in [38] discusses how the deployment strategy changes for one UAV relay to support an ad-hoc network according to different connectivity objectives. Optimization methods are presented to maximize (1) global and (2) worst case connectivity based on properties of the Minimum Spanning Tree (MST) graph [92], or (3) minimize network bisection probability and (4) maximize the Fiedler value using graph Laplacian \mathcal{L} . Each objective results in a different UAV relay deployment plan.

In the previous three examples, the deployed relays essentially became nodes in

the supported ad-hoc network, thus changing the topology and information routing options. However, another strategy is to assume that relays are similar to gateways, and have a separate strong communication network to connect to other relays and the rest of the world. These can then be used as a communication backbone to support a disconnected ad-hoc network (as illustrated with the “UAV Comm. Relay” in Fig. 1-1). The network then takes on a form more consistent with a cellular network architecture, where relays serve as mobile base stations and connect to nodes in their area of responsibility. This is how current military large-scale UAV communication relays operate [89]. Mobile Backbone Network (MBN) optimization seeks to simultaneously position and assign K MBN relays over a network in order to: (1) maximize the minimum throughput in the network [39], (2) maximize the aggregate network throughput [39], (3) maximize the number of nodes meeting a threshold throughput [40], and (4) optimize relay deployment and node positioning (if those nodes can be controlled) for objective (3) [40]. In objectives (3) and (4), the optimization is solved using a Mixed-Integer Linear Program (MILP) for a network optimization formulation [92]. Similar to other previous studies, the optimal solver scales poorly with increasing number of nodes and relays, and a suboptimal greedy solution based on solving the Max-Flow problem [92] is provided in parallel for fast and provably good solutions.

2.3.3 Controlling Communications Through Task Allocation

A similar, yet distinct method to control network communications in a team of UAVs is through task allocation, and is the method proposed in this thesis. In this problem, the system consists of a multi-agent team with a set of objectives and similar to motion planning problems discussed in Sect. 2.3.1, connectivity is an implicitly derived requirement. Past frameworks have considered communication constraints in task allocation to mitigate the risk of network disconnects [18]. As demonstrated in [93] and shown in Chapter 5, this type of strategy remains conservative, and performance can be increased if agents coordinate explicitly to support communication requirements.

Authors in [42] discuss a task allocation based network control mechanism. A

centralized task allocation planner initializes by dividing the team of robots into subset clusters. Each cluster is assigned a common task to accomplish. Then, motion planning potential fields similar to Eq. (2.14) are used to attract the cluster to the goal, to the base, and to maintain connectivity within the cluster while accomplishing the task. If the cluster cannot achieve the goal and stay connected, it is merged with a neighboring cluster to expand the multi-hop capability. Part of this work is demonstrated in a field experiment with a team of UGVs, and results (while successful) again highlight the difficulty in predicting link quality due to rapid variations and uncertainty in the dynamics.

The framework proposed in this thesis is novel compared to these techniques because it dynamically assigns individual agents to serve as communication relays in optimized locations using distributed control. By explicitly coupling the task assignment and relay planning processes, the team is able to better optimize the use of agent resources to address current mission needs. This leads to improved performance and added flexibility in real-time dynamic mission scenarios. Section 3.3 and Chapter 4 address this in greater detail.

2.4 Summary

This thesis chapter presented a broad survey of research studies investigating network communication constraints and control in a team of multiple unmanned vehicles. Technical background was provided in network architecture, wireless channel modeling, topology theory, and information routing mechanisms which form the basis of most of the work in this field. The consequences of degraded communications in a multi-agent system were highlighted through examples covering systems with ineffective return of remotely sensed data, complications in path planning and formation control, and challenges in cooperative mission planning. Several strategies to control mobile agents in order to support communications were investigated in this chapter, including motion planning and optimal deployment of a pre-designated set of mobile communication relays. The framework developed in this thesis is unique

from these studies, as it dynamically optimizes task assignment, agents roles, and network support through relays. The communication models employed in it used are realistic, and founded on the principles listed at the beginning of this chapter.

Chapter 3

Distributed Dynamic Planning Algorithm

3.1 Task Allocation Planning Strategies

The vision to employ teams of robotic agents, such as UAVs or UGVs, autonomously in complex missions [1]. These operations require different agents in the team to have different roles and responsibilities during execution, much like modern day sports teams. Ensuring proper spatial and temporal coordination between them is critical. Assigning tasks to agents involves solving a complex combinatorial decision problem [94], with constraints that may take several forms including [5, 52, 95]:

- Mission tasks may be in different locations
- Tasks can require agents to have specific capabilities (such as a specific sensor) which only certain agents in the heterogeneous team possess
- Tasks may require coordination between several agents

The task allocation problem involves planning the resource allocation in a team of N_a unmanned systems, to accomplish a set of N_t tasks. The objective of the planner is to maximize reward for the mission by assigning the right asset, to the right task, at the right time subject to mission constraints. The task allocation in this work can

be formulated as the following optimization problem:

$$\begin{aligned}
& \underset{\mathbf{x}, \boldsymbol{\tau}}{\operatorname{argmax}} && \sum_{i=1}^{N_a} \left(\sum_{j=1}^{N_t} c_{ij}(\mathbf{p}_i(\mathbf{x}_i, \boldsymbol{\tau}_i)) x_{ij} \right) && (3.1) \\
& \text{s. t.} && \sum_{j=1}^{N_t} x_{ij} \leq L_i, \quad \forall i \in \mathcal{I} \\
& && \sum_{i=1}^{N_a} x_{ij} \leq 1, \quad \forall j \in \mathcal{J} \\
& && x_{ij} \in \{0, 1\}, \tau_{ij} \in \{\mathbb{R}^+ \cup \emptyset\}, \forall (i, j) \in \mathcal{I} \times \mathcal{J}
\end{aligned}$$

where $x_{ij} = 1$ if agent i is assigned to task j , and $\mathbf{x}_i \triangleq \{x_{i1}, \dots, x_{iN_t}\}$ is a vector of assignments for agent i , whose j^{th} element is x_{ij} . The index sets for the agents and tasks are defined as $\mathcal{I} \triangleq \{1, \dots, N_a\}$ and $\mathcal{J} \triangleq \{1, \dots, N_t\}$ respectively. The variable-length vector $\mathbf{p}_i \triangleq \{p_{i1}, \dots, p_{i|\mathbf{p}_i}|\}$ represents the *path* for agent i , an ordered sequence of tasks with elements $p_{in} \in \mathcal{J}$ for $n = \{1, \dots, |\mathbf{p}_i|\}$, representing the order in which the tasks will be executed. The vector of times, $\boldsymbol{\tau}_i \triangleq \{\tau_{i1}, \dots, \tau_{i|\mathbf{p}_i}|\}$, denotes the times at which agent i proposes to execute each task. The length of the path is denoted by $|\mathbf{p}_i|$ and may be no longer than L_i , a maximum allowable task limit for each agent. The objective of the task assignment problem in Eq. (3.1) is to maximize the sum of the reward values for each agent i . In this formulation, the score c_{ij} that agent i obtains by performing task j is defined as a function of the value of the task, the cost of execution (e.g. fuel consumed), and the time τ_{ij} at which the agent executes the task compared to the optimal task execution time [96].

The numerous possible combinations of agent to task assignments and the inherent inter-dependencies in score from executing tasks in different orders makes the task assignment problem very difficult to solve for $L_i > 1$ (NP-hard). A variety of optimization algorithms and frameworks have been developed to solve this problem for a team of unmanned vehicles [97]. Centralized planners solving a Constraint Satisfaction Problem (CSP) or using Dynamic Programming can provide the optimal task assignment solution, but usually do so in an unreasonable amount of time given the dynamic nature of these complex missions [98]. Approximate solvers are avail-

able for this purpose and can be quite fast. For instance, in [99], a process algebra framework is used to model interactions between tasks and UAV agents to optimize task allocation by using an efficient state-space search algorithm. Authors in [100] employ evolutionary optimization and the Genetic Algorithm to rapidly steer a search of possible assignments to a good solution for large teams of UAVs involved in ISR operations. The problem formulation can also be relaxed and then solved using a Mixed-Integer-Linear-Programming approach to find an optimized task allocation. This type of strategy was used in [101] to generate assignments for heterogeneous UAVs to track multiple ground targets collaboratively. While these centralized planners generate good solutions, they require a central node to have full situational awareness of all agents in the team.

Instead, distributed frameworks can be implemented, where each agent makes planning decisions based on its local situational awareness. The distributed task planning architecture has several benefits for a multi-agent system. First, it reduces the need to communicate an excessive amount of state information to a ground control station for centralized control. Second, it mitigates latency effects by enabling agents to leverage their immediate local situational awareness in planning the task assignment. The challenge is finding effective task allocation solutions, and also making sure the distributed team is on the same page with respect to the plan. One method to do this involves an initial centralized space partitioning to then allow agents to make decentralized decisions in their areas of responsibility [102]. Another way is to use *implicit coordination* where agents reach consensus on the global situational awareness of the team, and then each solve the same problem using the centralized techniques listed above [80–82]. Unfortunately, as mentioned in Sect. 2.2.2, this strategy can involve significant communication overhead costs which is not ideal given the problem statement for this work. As such, popular methods to solve the task assignment problem have turned to decentralized auction algorithms which provide efficient but also sub-optimal solutions. This thesis employs one such algorithm named the Consensus-Based Bundle Algorithm (CBBA) which is described in the next section [5, 103]. Other methods that have received significant recent interest are

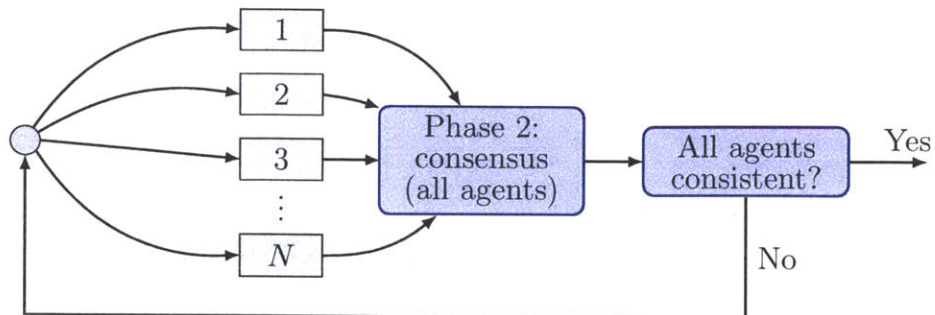


Figure 3-1: Consensus Based Bundle Algorithm

solutions based on Markov Decision Processes (MDP) which describe the environment in terms of states \mathcal{S} , actions \mathcal{A} , probability of state transition \mathcal{P} , reward function \mathcal{R} , and time steps \mathcal{T} [98]. The objective in MDPs is to find the policy that maximizes reward for each state at each time given the possible state-action pairs. MDPs can be reformulated in a decentralized architecture, or Decentralized MDPs (Dec-MDP) to solve task assignment problems in multi-agent systems [98].

3.2 Consensus Based Bundle Algorithm

As described in the previous section, the inter-dependencies in number of possible assignments make Eq. (3.1) combinatorial and very difficult to solve if planning for more than 1 task for each agent. This thesis therefore turns to the Consensus-Based Bundle Algorithm (CBBA), a decentralized market based task allocation protocol that provides provably good solutions for heterogeneous decentralized multi-task assignment functions. Each CBBA iteration is composed of two phases illustrated in Fig. 3-1. First, in the bundle building phase each agent greedily bids on an ordered bundle of tasks based on its capabilities and its location with respect to the task. Second, in the consensus phase, agent bid information is shared, and conflicts in task assignment are identified and resolved. The algorithm iterates until it converges on a conflict-free plan [5].

During the first phase of the algorithm, agents place bids described by $s_{ij} = (i, j, c_{ij})$, where i , j , and c_{ij} respectively represent the agent index, the task index,

and the proposed score. Each agent i adds its top bids to a bundle $\mathbf{b}_i = \{s_{ij_1}, \dots, s_{ij_n}\}$, where the length of \mathbf{b}_i represents the planning horizon established for i . In the bundle construction process, each agent compares its ability to do a task with its local bid space $\mathcal{A}_i = \{s_{i_1j_1}, \dots, s_{i_Nj_N}\}$ which contains i 's understanding of the current winning bids for each task from the previous CBBA iteration. It computes this ability c_{ij} using an internal score function. In this thesis, the score function consists of the task value discounted for every unit of time past its encoded optimal starting time, minus cost incurred for travel to the task. Each task has a time window of validity, and bids for execution outside that window are not allowed. Each agent then compares its ordered scores with \mathcal{A}_i , selects the first score which out-bids a current winner as its next bid, and adds this bid s_{ij} to the end of its bundle [5].

One important requirement for the algorithm to converge is that bids need to satisfy *Diminishing Marginal Gains* (DMG). The DMG condition essentially means that no bid can be made that would increase the score c_{ij} for a task. In other words agents cannot place subsequently higher bids on tasks as a result of their other bids. As such, the bid value of each new bid actually posted in the bundle is limited to the minimum score from previously posted bids [104]. The DMG condition stems from a well studied property in greedy algorithms called submodularity, and if adhered to, ensures CBBA will converge to a *conflict-free* solution where each task is assigned to only one agent.

Each agent then substitutes their new bids in bid-space \mathcal{A}_i . Once agents are done building their bundles, they each share their \mathcal{A}_i with the team synchronously, and pass new information from other agents in a decision table that consolidates new bids and resolves conflicts (see [5] for more details). The algorithm iterates until consensus is reached twice on an unchanged set of proposed assignments, which indicates convergence. At this point the task assignment terminates, and agents carry out tasks in their bundle as planned.

The CBBA algorithm runs in polynomial time, and has demonstrated good scalability with increasing numbers of agents and tasks. It has been demonstrated in numerous applications with teams of heterogeneous unmanned systems, has been ex-

tended to account for asynchronous messaging, and has been modified using hierarchical team structures to control large numbers of UAVs [5, 105–108].

3.3 CBBA with Relays

3.3.1 Motivation for the Algorithm

A complicating factor not considered in the CBBA algorithm is that some operational domains require that the team satisfy system communication requirements during mission execution. As described in Sect. 2.2, a multi-unmanned vehicle system relies on communications to operate. The inability to communicate sensor data to a base station in real time (e.g. live video) may render the multi-agent system ineffective [74]. Failure to properly exchange command and control messages can lead to potentially dangerous system failures. Poor communication may also prevent the team from reaching agreement on a plan, thus remaining idle, or sometimes worse, taking action on an incomplete plan leading to unwanted consequences [52, 109]. UAVs in a team must therefore be able to communicate to the base and to each other.

The motivating scenario for this thesis in Sect. 1.2, illustrates the need to consider communications in the task assignment. Vehicles are tasked to select locations to stream live video back to the base. For simplicity, it is assumed (in this section only) that vehicles have a fixed communication radius, and cannot successfully transmit data if disconnected from the base. The left box in Fig. 3-2 shows the initial environment in an example of this scenario with 5 UAVs, a base station, and 5 possible tasks with associated values. Then Box A in Fig. 3-2 shows the possible outcome of a task assignment using CBBA. Because of the scoring function described in 3.2, agents greedily bid and get assigned high value tasks. In this case, these are farther from the base station, and the assignment results in a disconnected network. Here UAVs 3 and 4 are disconnected, cannot stream data back to the base, and therefore receive no reward for their tasks despite incurring the fuel costs to execute them.

To prevent these unwanted disconnects, communication constraints can be explic-

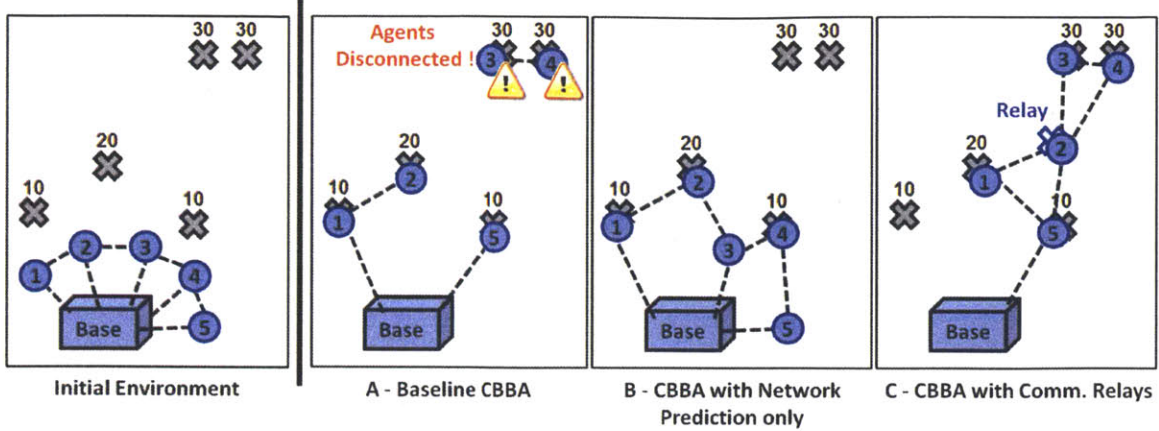


Figure 3-2: Different task allocation strategies for network support

itly considered in the planning process. Task allocation information, such as task locations and planned execution times, can be leveraged by the agents to predict the network topology at execution and identify communication constraint violations. For example, in Box B of Fig. 3-2 agents predict the network topology for the proposed assignment and drop tasks that cause disconnects (UAV 3 drops its task, and UAV 4 out-bids 5 in a nearby task). This approach guarantees network connectivity, but is conservative because agents tend to only accomplish tasks in the local vicinity. In this case UAVs 3 and 5 are unused. An improved solution is to use some agents as communication relays, where data can be transmitted back to the base station through designated neighboring agents. This requires explicit cooperation between agents to determine where and when relay tasks are required, which agents should execute these relay tasks, and which agents can execute the main mission tasks. Figure 3-2 Box C illustrates this cooperative scenario, where agents predict the network, detect the potential disconnects, create relay tasks to fix them, and propose these relay tasks to the rest of the team. Here, UAV 2 changes its assignment to service a relay task proposed by UAV 3. This results in a team capability to accomplish higher value tasks, increasing the overall mission score [96].

Designing a framework to enable this type of task assignment mission planning is non-trivial. As agents travel around the operating area, links are created and destroyed. Predicting the topology over time can be computationally intensive since the network is dynamic. Planning solutions which meet communication constraints are

highly inter-dependent because task assignments, relay task creations, and network connectivity predictions are closely coupled spatially and temporally. Even small perturbations in the agent assignments can alter the network topology and lead to constraint violations. The CBBA with Communication Relays algorithm, first introduced in [93], was built around the CBBA task assignment routing to specifically address communication constraints. This framework differs from previous communication control methods for multi-agent systems described in Sect. 2.3 by simultaneously optimizing relay and task assignments, instead of preallocating agents to specific roles and then solving decoupled task assignment and network connectivity planning problems. By explicitly coupling the task allocation with relay creation, the team is better able to optimize the use of agent resources given the current mission needs, leading to improved performance and added flexibility in real-time dynamic mission scenarios. The framework described in this section was developed in previous work [93, 96], but is introduced here in detail because it serves as the foundation for Chapters 4 and 5.

3.3.2 Algorithm Mechanics

The purpose of the CBBA with Relays algorithm is to efficiently allocate agents to tasks while ensuring that the network remains connected to a predefined *base station* during task execution. The algorithm achieves connectivity by leveraging the task allocation capabilities of CBBA with an outer loop that enforces these connectivity constraints. CBBA solves the allocation problem by creating assignments \mathcal{A} of agent-task pairs over the set of initial tasks \mathcal{J}_i available to agent i for bidding. However, the resulting assignment may lead to a vehicle configuration which causes the network topology to disconnect. For this reason, agents perform a *network prediction*. Using the vector of predicted times each assigned task will be executed obtained by CBBA, along with the known position of each task, agents are able to predict the location of other active agents at the start of each task. As such, agents can predict whether starting a new task will cause a disconnect, and therefore plan how to stay connected. Two strategies are available to prevent network disconnect: (1) simply drop tasks

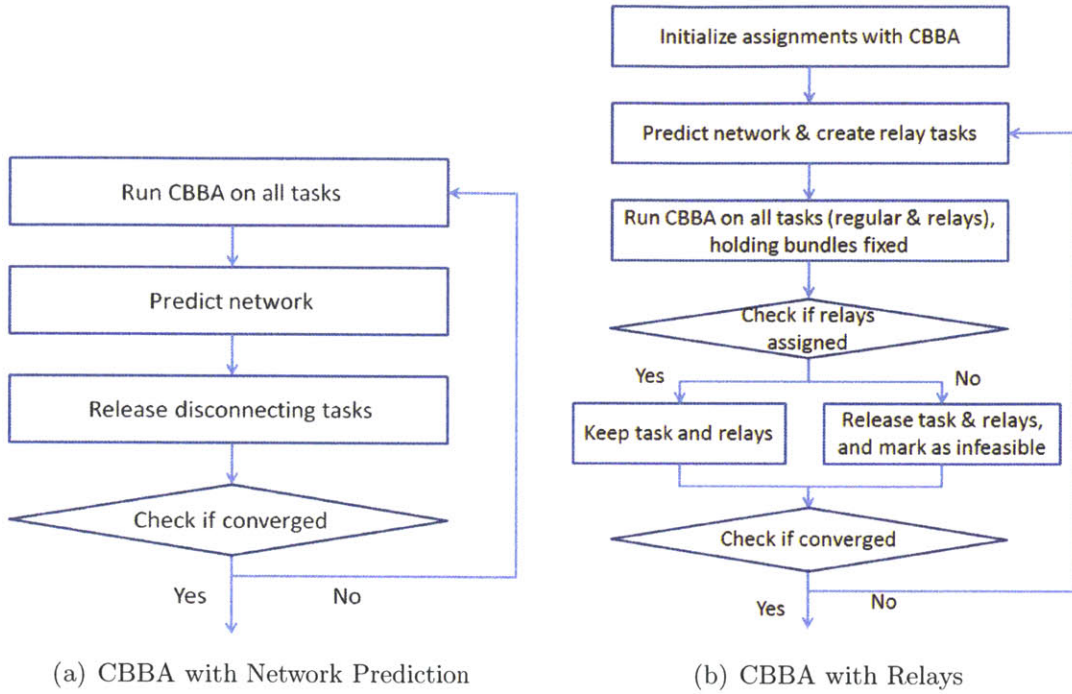


Figure 3-3: CBBA with Communication Relays Algorithm Flow [93]

which cause network disconnects (named *CBBA with Network Prediction* - Box B in Fig. 3-2), or (2) request communication relays to support disconnected tasks (*CBBA with Relays* - Box C in Fig. 3-2). In both cases the disconnected assignment is identified, and with *CBBA with Relays* new relay tasks \mathcal{R} are created. The next CBBA iteration is then initiated with these considerations, and again creates a new task assignment which may also lead to a disconnect. Several mechanisms described later in this section are put in place to ensure that over multiple iterations, this planning algorithm converges to a conflict free plan fully supported by the network. Figure 3-3 (a) and (b) respectively represent the planning loop for the conservative *CBBA with Network Prediction* and cooperative *CBBA with Relays*.

Network predictions are performed for the predicted start times of each task in agent bundles. The agent owning the task for which the network is predicted is responsible for this calculation. If the *CBBA with Relays* planning strategy is used, that agent determines where and when relays must be created to support that network configuration. For each relay task r created, the algorithm tracks which tasks depend

on it, as well as the dependency of this relay on other created relays. If in the following CBBA iteration a relay task is not assigned, the network will likely not be supported for tasks and other relays dependent on it. In this case, the current winning agent i *may* remove the task in violation from its assignment. The decision to drop a task is performed stochastically, which breaks the symmetry in a set of conflicting assignments. This mitigates stalemate scenarios such as when two agents each bid and win disconnecting tasks, propose relays which do not get assigned, then both drop their tasks and immediately re-bid on each other's tasks leading to the same disconnected scenario. If a task is dropped it is then removed from the subset of tasks J_i that agent i can bid on in future CBBA iterations. Dropped relays are simply deleted. This prevents endless cycles from occurring where the same agents bid on the same tasks leading to the same disconnects. Another key feature of this algorithm involves properly incentivizing relay tasks so that agents elect to bid on them in the next CBBA iteration. To accomplish this, relays are assigned a *virtual value* which depends on the predicted reward from accomplishing its dependent tasks. This is discussed more in detail in Sect. 4.2.2 and 4.6.2.

The full CBBA with Relays algorithm is presented in Algorithms 1 and 2 [96]. Components of the algorithm include:

- $\text{Winning-Agent}(j)$ to return the current winning agent for task j .
- $\text{Dependent-Relays}(j)$ to track dependencies for each task j on the listed relay indexes to be connected.
- $\text{Dependent-Tasks}(r)$ returns the indexes of all tasks that rely on relay task r being serviced.
- $\text{Keep-Task}(j, \mathcal{A})$ is the stochastic decision function to drop a task if it causes a disconnect as described above.
- $\text{Place-Relays}(j, \bar{\mathcal{J}}, \mathcal{A})$ creates relay tasks required to connect a disconnected task j to the base station. The function is responsible for specifying appropriate

Algorithm 1 CBBA-RELAYS(\mathcal{I}, \mathcal{J}) [96]

```
1:  $\mathcal{J}_i = \mathcal{J}, \forall i \in \mathcal{I}; \mathcal{R} = \emptyset$ 
2:  $\bar{\mathcal{J}} = \{\mathcal{J}_1, \dots, \mathcal{J}_{N_a}, \mathcal{R}\}$ 
3: while  $\neg$  converged do
4:    $\mathcal{A} \leftarrow$  CBBA( $\bar{\mathcal{J}}$ ) (See [5] for details)
5:   for ( $r \in \mathcal{R}$ ) & ( $r \notin \mathcal{A}$ ) do
6:      $(\bar{\mathcal{J}}', \mathcal{A}') \leftarrow$  PRUNE-TASK-SPACE( $r, \bar{\mathcal{J}}, \mathcal{A}$ )
7:   end for
8:    $\mathcal{J}_{disconnected} \leftarrow$  Predict-Disconnects( $\bar{\mathcal{J}}', \mathcal{A}'$ )
9:   for  $j \in \mathcal{J}_{disconnected}$  do
10:     $\bar{\mathcal{J}}'' \leftarrow$  Place-Relays( $j, \bar{\mathcal{J}}', \mathcal{A}'$ )
11:  end for
12:  if ( $\bar{\mathcal{J}}'' = \bar{\mathcal{J}}$ ) & ( $\mathcal{A}' = \mathcal{A}$ ) then
13:    converged  $\leftarrow$  true
14:  end if
15:   $\bar{\mathcal{J}} \leftarrow \bar{\mathcal{J}}''; \mathcal{A} \leftarrow \mathcal{A}'$ 
16: end while
17: return  $\mathcal{A}$ 
```

locations, values, and time-windows for these relay tasks (further discussion is provided below and in Sect. 4.2.2).

- Predict-Disconnects($\bar{\mathcal{J}}, \mathcal{A}$) is the function used by agents to predict the network at the predicted start time of each of their tasks in assignment \mathcal{A} . It returns the set of disconnected tasks.

The Place-Relays algorithm is responsible for creating relay tasks with locations and time-windows that ensure connectivity for the main task they are designed to connect. This process depends on the communication environment and on networking models used to describe information flow (see Sect 2.1.2). The original development of this algorithm, described here, considers a simple *disc* communication model. Communication links are supported as long as inter-node distances are within a known radius R_{COMM} . The global network can therefore be described by its set of *subnetworks* with interconnected agents. The Place-Relays algorithm therefore creates the necessary number of relays to bridge the gap between the closest agent in a disconnected subnetwork s_m to one in the base subnetwork s_0 . If the number of needed relays exceeds the number of available agents, the disconnected task is labeled as infeasible and subsequently dropped. Chapter 4 revisits this algorithm in greater

Algorithm 2 PRUNE-TASK-SPACE($r, \bar{\mathcal{J}}, \mathcal{A}$) [96]

```
1:  $\bar{\mathcal{J}} \leftarrow \bar{\mathcal{J}} \setminus \{r\}$ 
2: for  $j \in \text{Dependent-Tasks}(r)$  do
3:   for  $r' \in \text{Dependent-Relays}(j)$  do
4:      $\text{Dependent-Tasks}(r') \leftarrow \text{Dependent-Tasks}(r') \setminus \{j\}$ 
5:     if  $\text{Dependent-Tasks}(r') = \emptyset$  then
6:        $\bar{\mathcal{J}} \leftarrow \bar{\mathcal{J}} \setminus \{r'\}$ 
7:        $\mathcal{A} \leftarrow \mathcal{A} \setminus \{r'\}$  if  $r' \in \mathcal{A}$ 
8:     end if
9:   end for
10:   $\text{Dependent-Relays}(j) \leftarrow \emptyset$ 
11:   $keep \leftarrow \text{Keep-Task}(j, \mathcal{A})$ 
12:  if  $\neg keep$  then
13:     $\mathcal{A} \leftarrow \mathcal{A} \setminus \{j\}$ 
14:     $\mathcal{J}_{\text{Winning-Agent}}(j) \leftarrow \mathcal{J}_{\text{Winning-Agent}}(j) \setminus \{j\}$ 
15:  end if
16: end for
17: return  $(\bar{\mathcal{J}}', \mathcal{A}')$ 
```

detail to consider more realistic communication dynamics which involve path loss, uncertainty in fading, data-rate and bit-error-rate constraints, as well as information routing.

3.3.3 Algorithm Performance

The CBBA with Relays algorithm presented in this section is guaranteed to converge, runs in real-time, and ensures a strongly connected network while tasks are being executed. Initial experiments were conducted in prior work to compare the performance of the three different planning strategies discussed in this section: (1) baseline CBBA with no network consideration, (2) the conservative CBBA with Network Prediction, and (3) the cooperative CBBA with Relays [93]. These were conducted both using Monte Carlo simulations, as well as hardware experiment involving six UGVs at the MIT Aerospace Control Lab indoor autonomous vehicle test environment [110]. Results show that the baseline CBBA causes significant network disconnects leading to poor performance. CBBA with Network Prediction improves the mission performance by preventing network disconnects, but is very conservative in the tasks it schedules and achieves only marginally higher performance. CBBA with Relays outperforms

the other algorithms by allowing cooperative task execution, achieving a higher score throughout the mission and ensuring connectivity during task execution. This comes at cost of increased runtime per planning cycle than the other two planning algorithms. Nevertheless, experiments described in Chapter 5 show the algorithm retains real-time performance in outdoor flight testing with a team of multiple UAVs.

3.4 Summary

This chapter introduced methods of task allocation used to coordinate efforts of a team of UAVs in a complex mission. The problem of assigning tasks to agents is combinatorial and difficult to solve for real-time applications. The Consensus Based Bundle Algorithm (CBBA) was introduced as a sequential greedy method to effectively assign tasks in a decentralized architecture. While CBBA is guaranteed to converge on a conflict-free plan, where each task is assigned to at most one agent, it does not consider network communication constraints inherent to multi-vehicle systems. Therefore the CBBA with Communication Relays algorithm was developed to enable agents to predict the network topology at task execution using information encoded in the CBBA task assignment. The algorithm uses this prediction to identify network disconnects and plan for underutilized agents to support the network as communication relays. This cooperative strategy allows the team to achieve higher mission performance and guaranteed connectivity with a base station. This work laid the foundation for the next two chapters of this thesis, but considered simplified communication models which will break down in real world applications. The next chapter in this thesis, therefore, builds onto this framework to enable teams of UAVs to cooperatively execute complex missions and maintain network connectivity given realistic wireless communication dynamics, uncertainties, and information routing protocols. Chapter 5 then demonstrates the implementation of these algorithms in outdoor flight test experiments with a team of three UAVs.

Chapter 4

Improving Network Support

4.1 Motivation for Improvements

The CBBA with Relays algorithm introduced in Sect. 3.3 laid the foundation for network communication control using task allocation in a multi-agent team. It was shown that through cooperative planning, agents can be dynamically assigned to take on mission tasks or support network requirements to improve global mission performance. This is fundamentally different from other multi-vehicle network control methods described in Sect. 2.3.2 where roles are typically pre-designated. CBBA with Relays dynamically optimizes the use of agents which adds flexibility in complex mission executions. In this chapter, the framework is improved to include realistic networking dynamics beyond the simplified and deterministic connectivity objective considered thus far.

Multi-UAV systems may have different communication requirements based on design and application. For instance, UAVs are commonly used in surveillance missions, and deployed to collect and stream live imagery back to an analysis center. Live video data requires high data-rate, but is tolerant to some faults in transmission [11]. This imposes an end-to-end wireless channel capacity requirement from the UAV collecting data, through other routing agents, to the base, at the time of task execution. If multiple UAVs are simultaneously sending data, channel capacities must account for the total data flow routed through each link. Figure 4-1 illustrates four UAVs

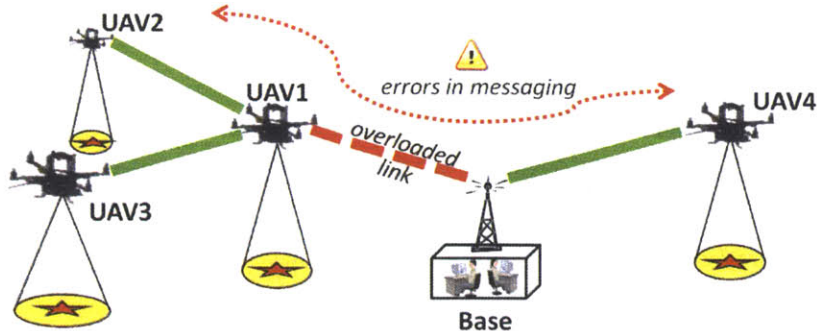


Figure 4-1: Motivation to improve CBBA with Relays

executing tasks which are connected to the base through multiple hops, a satisfactory condition for the work in Sect. 3.3. While each link can support a single UAV streaming video, UAV 1 must support additional data routing for UAVs 2 and 3, and becomes overloaded which creates a data bottleneck. This prevents data from getting to the base. In this situation, network support is not just based on connectivity, but also on link capacities and data routing.

The communication network also supports the exchange of command and control messages. In a centralized control system, each agent communicates back and forth only with the planning agent (typically the base). However in a distributed system, as the one in this thesis, messages are exchanged between all agents in the network. Messages can include bidding information to run CBBA, motion planning, or task data needed by other agents for cooperative execution. These exchanges require lower data rates, but are intolerant to errors [11]. This imposes an end-to-end bit-error-rate (BER) threshold requirement, or the probability of transmission error (see Sect. 2.1.2), between all agents in the network. Errors in transmission, if recognized, require retransmission which creates delays and degrades performance. Since each link has an associated BER, end-to-end BER propagates from hop to hop. In Fig. 4-1, all UAVs are appropriately connected to the base, but the end-to-end BER between UAV 4 and UAV 2 is insufficient to properly coordinate plans. This can significantly degrade the team’s ability to execute the mission [52].

Finally, wireless communications are subjected to dynamics which can vary the performance of each link. Path loss attenuates wireless links with increasing distance.

Furthermore stochastic shadowing and multipath fading can significantly impact the quality of the received signal. As such, the system faces continuously changing network performance. Finally, the rate of attenuation and amount of fading from the environment may be unknown to the system, and can change over time. These dynamics affect the capacity and BER in wireless channels, and must be accounted for during the planning stages to support the network requirements.

This chapter introduces modifications to the planning framework described in Sect. 3.3 to address the following communication performance considerations: (1) data-rate, (2) data routing, (3) inter-agent BER, and (4) uncertainty.

4.2 System Overview

4.2.1 Wireless Channel Model

Section 2.1.2 provided technical background to network communications. In this work, the network topology is controlled purely through positioning agents, as is often the case in multi-agent communication studies (see Sect. 2.3). A fundamental factor in wireless channel performance is the Signal to Noise Ratio (SNR), which generally decreases as the distance increases, and varies stochastically due to shadowing and multipath fading. Equations (2.1) and (2.3) are combined to provide the model for SNR (γ) in dB used in this thesis

$$\gamma_{dB} = 10 \log \left(\frac{P}{N_o W} \right) + K_{dB} - 10\alpha \log \left(\frac{d}{d_o} \right) - \mathcal{N}(0, \sigma_{dB}^2) \quad (4.1)$$

This work assumes that the radio modules transmit at a data-rate approaching *Shannon Capacity*, described by Eq. (2.6) as a function of γ . Even though this value is an upper-bound, and actual data transmissions are often set to lower rates due to fading [10], it serves as a good general indicator of data-rate performance in *bps*. Similarly, this study assumes $BER = Q(\sqrt{2\gamma})$ where Q is the *Q-function* defined in Eq. (2.7). While BER varies using different modulation schemes and data-rates, the relation used here is general and accurately represents performance trends.

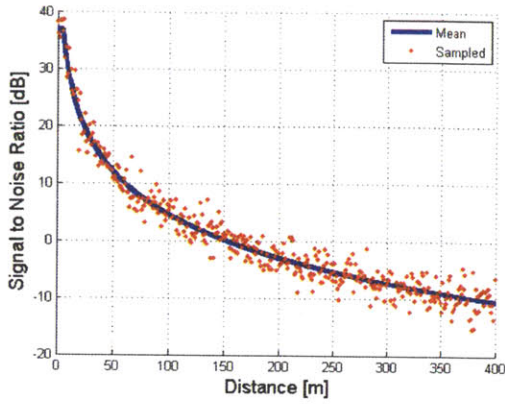
Table 4.1: Communication Model Parameters

Parameter	Symbol	Value
Transmission Power	P	500 <i>mW</i>
Channel Bandwidth	W	1 <i>MHz</i>
Path Loss Exponent	α	2.5
Reference Distance	d_0	5 <i>m</i>
Noise Power Density	N_0	10^{-10} <i>W/Hz</i>
Fading Variance	σ_{dB}	2 <i>dB</i>
Equipment Gain	K_{dB}	0 <i>dB</i>

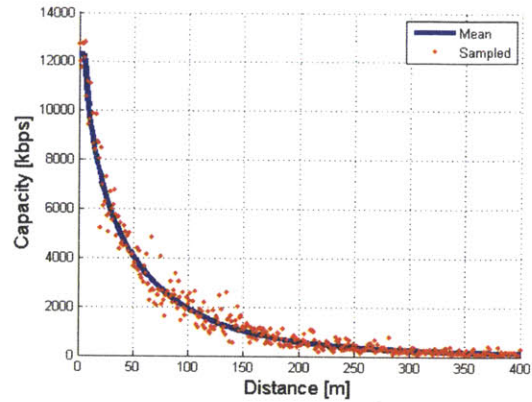
Parameters listed in Table 4.1, which are representative (but not specific) of small video transmission modules used in UAV applications, are substituted into the above relationships for γ , channel capacity u , and BER to form the model. These parameters can be changed freely without impacting the approach of this work. The resulting wireless performance trends for SNR, capacity, and BER are plotted in Figure 4-2. In these plots, the blue line represents the deterministic trend without stochastic effects from channel fading ($\sigma_{dB} = 0$), whereas the red dots show values sampled over the normal distribution according to $\sigma_{dB} = 2$. These performance trends relate back to cooperative mission planning with network support as follows:

- Both types of communication requirements, data-rate and BER, depend on the channel SNR which is a measurable value using modern radio equipment.
- In order to increase the capacity of a channel, the SNR must increase, and therefore agents must be positioned closer to one another.
- Similarly to decrease BER, SNR must increase and inter-agent distance must decrease.

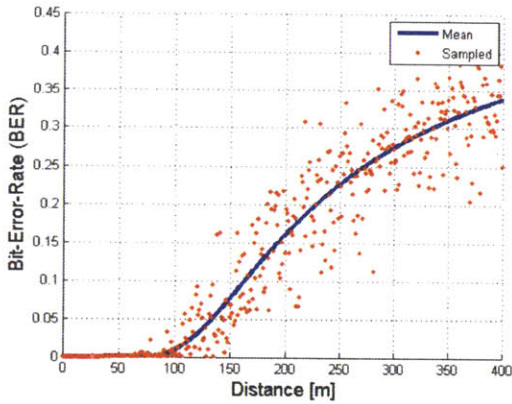
The models listed above are then used along with known agent positions to compute the network topology of inter-agent channel capacities and BER. Since radio modules require a threshold minimum SNR to recognize a link, wireless channels between agents with weak links (below γ_{thresh}) are excluded from the communication graph \mathcal{G} . The topology is subsequently used to predict how information is routed



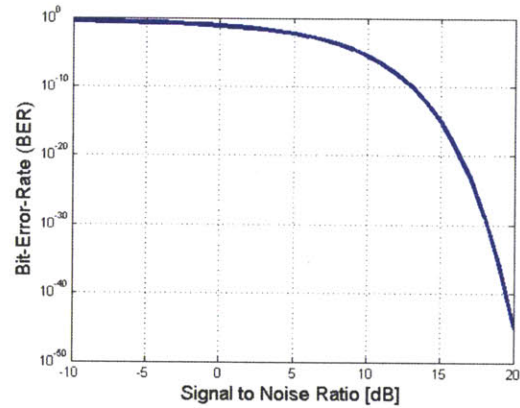
(a) SNR to distance trend



(b) Capacity to distance trend



(c) BER to distance trend



(d) BER to SNR trend

Figure 4-2: Wireless channel performance model

through the network, as will be discussed in Sections 4.3 and 4.4. This model for wireless communications assumes the following:

- There are two different communication systems on each agent, one to handle sensor data (video link) and one for command and control (C2 link). This implementation is common on modern UAV systems [8], and allows separate optimized routing for both types of messages. For simplicity in this work, both communication systems share the same parameters and numerical values but are routed differently.
- Communication devices can transmit and receive data at the same time. This is commonly achieved by incorporating multiple interfaces to the physical layer [61].

- Wireless channel frequencies are efficiently assigned. In reality, channel assignment is a difficult real-time decentralized optimization problem in itself and can have significant implications on the performance of the network. This topic is receiving significant attention in the wireless ad-hoc network research community [61], and is considered beyond the scope of this thesis.
- There are no delays in transmissions, and the multiple nodes do not interfere with each other. It is recognized that both phenomena can impact network performance, but like channel assignment, their mitigation is beyond the scope of this thesis.

4.2.2 General Approach

Several modifications to the original CBBA with Relays implementation (see Section 3.3) were required to consider realistic network communication requirements. This section describes the improved general planning approach proposed given any communication requirement considered in this thesis. This approach is described in Algorithm 3 and illustrated in Fig. 4-3. Recall that the overall objective of the framework is to maximize mission score by enabling agents to generate high reward at low cost while meeting communication requirements.

A description of each block in the figure follows:

- Block 1: The algorithm begins with the CBBA task assignment over all tasks and relays created in the previous iteration.
- Block 2: Relays which are unassigned or now *irrelevant* are removed and their dependencies dropped stochastically as described in Section 3.3.2. Each relay tracks the set of assignments used in the topology prediction at the time of its creation. If any of those assignments have changed, say if another agent out-bids one of the tasks to execute it early, the relay becomes *irrelevant* and is deleted (lines 8–13).
- Block 3: For each assigned task j , agent i assigned to it predicts the network

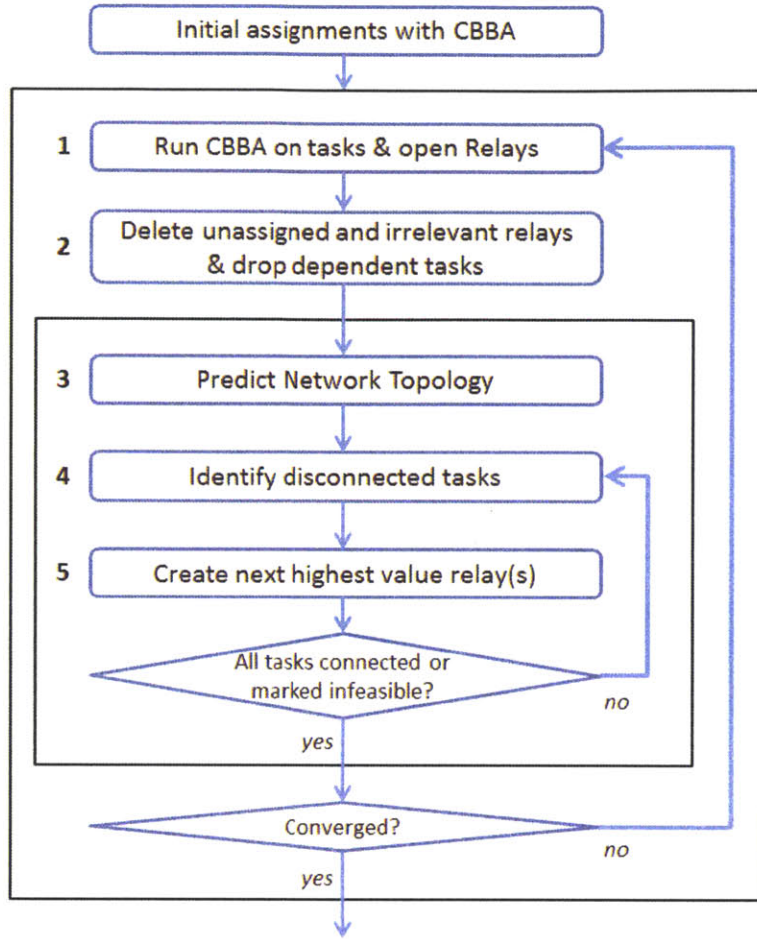


Figure 4-3: Improved general approach to CBBA with Relays

topology \mathcal{G} using: (1) CBBA information containing the set of committed tasks and their start times, and (2) that agent's current belief of the communication environment parameters \mathcal{C}_i (line 18). A method to estimate these parameters in uncertain environments is provided in Section 4.5.

- Block 4: The set of tasks which create network requirement violations, or *disconnects*, are identified in $\mathcal{J}_{disconnected}$ (line 19). All other tasks in this network prediction \mathcal{G} are therefore connected and allowed to be executed without relays assigned to them. This step varies depending on which type of communication requirement is used for planning. If connectivity is the only planning requirement, as was the case in the original implementation, $\mathcal{J}_{disconnected}$ includes all

Algorithm 3 IMPROVED CBBA-RELAYS($\mathcal{I}, \mathcal{J}, \mathcal{J}_{active}$)

```

1:  $\mathcal{J}_i = \mathcal{J}, \forall i \in \mathcal{I}; \mathcal{R} = \emptyset; \bar{\mathcal{J}} = \{\mathcal{J}_1, \dots, \mathcal{J}_{N_a}, \mathcal{R}\}$ 
2: while  $\neg$  converged do
3:    $\mathcal{A} \leftarrow \text{CBBA}(\bar{\mathcal{J}})$  (See [5] for details)
4:   for ( $r \in \mathcal{R}$ ) do
5:     if ( $r \notin \mathcal{A}$ ) then
6:        $(\bar{\mathcal{J}}', \mathcal{A}') \leftarrow \text{PRUNE-TASK-SPACE}(r, \bar{\mathcal{J}}, \mathcal{A})$ 
7:     else
8:       for ( $a \in \mathcal{A}_r'$ ) do
9:         if ( $a \notin \mathcal{A}$ ) then
10:           $\mathcal{J}_{\text{Winning-Agent}(a)} \leftarrow \mathcal{J}_{\text{Winning-Agent}(a)} \setminus \{j\}$ 
11:           $(\bar{\mathcal{J}}', \mathcal{A}') \leftarrow \text{PRUNE-TASK-SPACE}(r, \bar{\mathcal{J}}, \mathcal{A})$ 
12:        end if
13:      end for
14:    end if
15:  end for
16:  for ( $j \in \mathcal{A}'$ ) do
17:     $\mathcal{C} \leftarrow \mathcal{C}_{\text{Assigned-Agent}(j)}$ 
18:     $\mathcal{G} \leftarrow \text{Predict-Topology}(\bar{\mathcal{J}}', \mathcal{A}', \mathcal{C})$ 
19:     $\mathcal{J}_{\text{disconnected}} \leftarrow \text{FIND-DISCONNECTS}(\mathcal{J}_{active}, \mathcal{G})$ 
20:     $(\bar{\mathcal{J}}'', \mathcal{J}_{\text{infeasible}}) \leftarrow \text{PLACE-RELAYS}(\bar{\mathcal{J}}', \mathcal{J}_{\text{disconnected}}, \mathcal{G}, \mathcal{C})$ 
21:    for ( $j_{inf} \in \mathcal{J}_{\text{infeasible}}$ ) do
22:       $keep \leftarrow \text{Keep-Task}(j_{inf}, \mathcal{A})$ 
23:      if  $\neg$ keep then
24:         $\mathcal{A} \leftarrow \mathcal{A} \setminus \{j\}$ 
25:         $\mathcal{J}_{\text{Winning-Agent}(j_{inf})} \leftarrow \mathcal{J}_{\text{Winning-Agent}(j_{inf})} \setminus \{j_{inf}\}$ 
26:      end if
27:    end for
28:  end for
29:  if ( $\bar{\mathcal{J}}'' = \bar{\mathcal{J}}$ ) & ( $\mathcal{A}' = \mathcal{A}$ ) then
30:    converged  $\leftarrow$  true
31:  end if
32:   $\bar{\mathcal{J}} \leftarrow \bar{\mathcal{J}}''$ 
33:   $\mathcal{A} \leftarrow \mathcal{A}'$ 
34: end while
35: return  $\mathcal{A}$ 

```

tasks not in the base subnetwork. The processes adopted for data-rate and BER are discussed in Sections 4.3 and 4.4.

- Block 5: Given the network \mathcal{G} and its disconnected tasks $\mathcal{J}_{\text{disconnected}}$, relays are sequentially created in a greedy value optimization scheme as described in Algorithm 4 below.

Algorithm 3 converges when the same set of assignments, supported by the network, are proposed as in the previous iteration, and no new relay tasks are created.

The objective in Algorithm 4 is to create relays conducive to high reward achievable assignments in the next CBBA iteration. This does not necessarily mean re-connecting all tasks using the minimum number of relays. This is a key distinction from many other relay deployment optimization studies (see Section 2.3.2). Reward is maximized by performing high value tasks on time while incurring minimal cost. Relay tasks alone do not generate any reward since their agents are not generating mission data; however, they do incur costs. As such, it is only beneficial to execute relay tasks if the sum of the reward they enable through reconnected tasks is greater than the sum of (1) the cost they incur, (2) the reward opportunity passed up by their agents, and (3) the reward opportunity available to reconnected agents by executing connected tasks instead.

In some cases, it is more beneficial to allow a task to be disconnected than try to reconnect it, as illustrated in Fig. 4-4. Box A shows the same result from the baseline CBBA with Relays algorithm originally shown in Fig. 3-2, which leads to overloaded links. Box B shows how the network can be fixed using the minimum number of relays. To execute these tasks, all relays must be assigned, which exceeds how many agents are available. However, if any one relay does not get assigned, the network becomes either disconnected or overloaded, and reward is lost.

A potentially better solution, shown in Fig. 4-4 Box C, is to determine up front that one of the two tasks at the top (with value 30) should be dropped to better allocate the relay resource. However, to make this determination in a general way is a complex combinatorial problem with several challenges. It involves finding the optimal combination of up to N_a tasks and relays to assign, where N_a is the number of available agents and N'_t is number of tasks at that assignment time. This first requires solving for the optimal relay placements for each combination of task assignment, or $\binom{N'_t}{N'_t} \times \binom{N'_t}{N'_t-1} \times \dots \times \binom{N'_t}{1}$ combinations. Then, the best assignment must be selected by considering the location of available agents with respect to each combination to determine the best value. Finally, the task assignments are inter-dependent, and

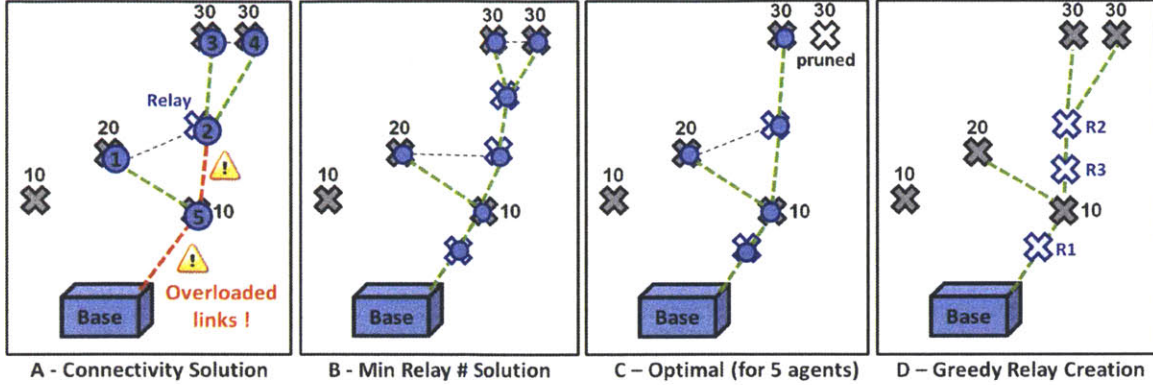


Figure 4-4: Relay placement to satisfy data-rate requirements for a team of 5 agents

the agent positions as well as the mission environment are dynamic, and therefore the solution may require frequent recomputation (see Section 3.1). The computation becomes quickly intractable and is impractical for dynamic replanning.

Instead, Algorithm 4 provides an efficient solution by sequentially choosing the next best task j in $\mathcal{J}_{disconnected}$ to reconnect using relays. For each j , it computes optimized relay locations to satisfy communication requirements (line 10). For instance, if connectivity is the only requirement, it finds the closest node between the subnetwork of the disconnected task and the base subnetwork, and solves for the number of relays N_r and their equally spaced positions. Here, $N_r = \lceil \frac{d_{min}}{R_{COMM}} \rceil$ and d_{min} is the distance between subnetworks. Relays are then temporarily added to the network graph \mathcal{G} and the FIND-DISCONNECTS function from Algorithm 3 then determines which additional tasks from $\mathcal{J}_{disconnected}$ are now reconnected. The value of the created relays is then determined using

$$v_r = K_v \sum_j^{\mathcal{J}_{depend}(r)} \frac{c_j}{N_r'(j)} \quad (4.2)$$

where $\mathcal{J}_{depend}(r)$ indicates the set of tasks j dependent on relay r , c_j is the predicted reward for j from CBBA, $N_r'(j)$ is the number of other relays j depends on, and K_v is a constant multiplicative scaling factor.

For all disconnected tasks, the algorithm finds the highest value task to reconnect and permanently adds that task with other reconnected tasks $\mathcal{J}_{connected}$, and its relays

\mathcal{R}' , to the network graph \mathcal{G} . This process is repeated until all tasks are reconnected or designated infeasible. The algorithm is illustrated in Fig. 4-4 Box D, where the index on the relays represent the order in which they were created. If a task requires more relays than the maximum possible number of available agents, or $N - 1$, it is marked as infeasible. If in Fig. 4-4 Box D only 4 agents were available, R3 would not be created and its dependent task with value 30 would be labeled infeasible. This eliminates task assignment options which are bound to fail but may cause additional CBBA iterations. Relays created in this process are then awarded a *virtual value* determined by Eq. (4.2) recomputed after all relays have been created. This incentivizes agents to bid on relay tasks during the CBBA task assignment. The choice of K_v has an impact on the cooperative behavior of the team, as will be shown in Section 4.6.2. Back to Fig. 4-4 Box D, all relays get created, but R3 has a significantly lower incentive value than R1 and R2 since it only connects one task which depends on all 3 relays.

The improved CBBA with Relays algorithm presented in this section also guarantees convergence and offers real-time performance as demonstrated in Chapter 5. This framework can be easily adapted to support several different communication requirements and information routing protocols simultaneously, a capability shared only by a few of the studies listed in Sect. 2.3.

4.2.3 System Framework

A full system framework was developed with this algorithm to conduct multi-UAV missions both in simulation and flight test. The system consists of multiple modules operating in the MATLAB environment, many modified from previous efforts [93]. The components and information flow are illustrated in Fig. 4-5. The following is a description of each component:

- **Simulator Mission Manager:** This module runs the mission, primarily by creating and managing tasks. These are created at uniformly distributed random locations, times, and windows of validity within the constraints of the environment. The value of each task is scaled linearly as a function of its distance from

Algorithm 4 PLACE-RELAYS($\bar{\mathcal{J}}', \mathcal{J}_{disconnected}, \mathcal{G}, \mathcal{C}$)

```
1:  $\mathcal{R} = \emptyset$ 
2:  $\mathcal{J}_{infeasible} = \emptyset$ 
3:  $\mathcal{J}_{active} = (j \in \mathcal{G}) \ \& \ (j \notin \mathcal{J}_{disconnected})$ 
4: while  $\neg done$  do
5:    $v_{max} = 0$ 
6:    $newRelay = false$ 
7:   for  $(j \in \mathcal{J}_{disconnected})$  do
8:      $\mathcal{J}_{disc}' \leftarrow \mathcal{J}_{disconnected} \setminus j$ 
9:      $\mathcal{J}_{connect} \leftarrow j$ 
10:     $(\mathcal{J}_{infeasible}, \mathcal{R}', \mathcal{G}') \leftarrow \text{FIX-DISCONNECT}(\mathcal{G}, j, \mathcal{J}_{infeasible}, \mathcal{J}_{active}, \mathcal{C})$ 
11:     $\mathcal{J}_{active}' \leftarrow \mathcal{J}_{active} + \mathcal{R}' + j$ 
12:     $\mathcal{J}_{disc}'' \leftarrow \text{FIND-DISCONNECTS}(\bar{\mathcal{J}}', \mathcal{J}_{active}', \mathcal{G}', \mathcal{C})$ 
13:     $\mathcal{J}_{connect} \leftarrow \mathcal{J}_{connect} + \{(j' \in \mathcal{J}_{disc}') \ \& \ (j' \notin \mathcal{J}_{disc}'')\}$ 
14:     $(\mathcal{R}', \mathcal{J}_{connect}) \leftarrow \text{Find-Dependencies}(\mathcal{G}')$ 
15:     $v \leftarrow \text{Compute-Value}(\mathcal{R}', \mathcal{J}_{connect})$ 
16:    if  $(v > v_{max})$  then
17:       $v_{max} = v$ 
18:       $\mathcal{R}_{max}' \leftarrow \mathcal{R}'$ 
19:       $\mathcal{G}_{max} \leftarrow \mathcal{G}'$ 
20:       $\mathcal{J}_{connect-max} \leftarrow \mathcal{J}_{connect}$ 
21:       $newRelay = true$ 
22:    end if
23:  end for
24:  if  $newRelay$  then
25:     $\bar{\mathcal{J}}' \leftarrow \bar{\mathcal{J}}' + \{\mathcal{R}_{max}', \mathcal{J}_{connect-max}\}$ 
26:     $\mathcal{J}_{active} \leftarrow \mathcal{J}_{active} + \{\mathcal{R}_{max}', \mathcal{J}_{connect-max}\}$ 
27:     $\mathcal{J}_{disconnected} \leftarrow \mathcal{J}_{disconnected} \setminus \{\mathcal{J}_{connect-max}\}$ 
28:     $\mathcal{J}_{infeasible} = \emptyset$ 
29:     $\mathcal{G} \leftarrow \mathcal{G}'$ 
30:  else
31:     $done = true$ 
32:  end if
33: end while
34: return  $(\bar{\mathcal{J}}', \mathcal{J}_{disconnected})$ 
```

the base. Specifically, a task located exactly at the base has zero value, whereas a task at the furthest Euclidean distance from the base in the environment has a value of 100. The purpose for this setting is to incentivize agents to execute difficult tasks from the perspective of network connectivity. It also relates well to real-world remote sensing objectives which often consist in surveying out-of-reach areas. Tasks further away offer more reward, but require agents to cooperate to execute them successfully. The Mission Manager also tracks

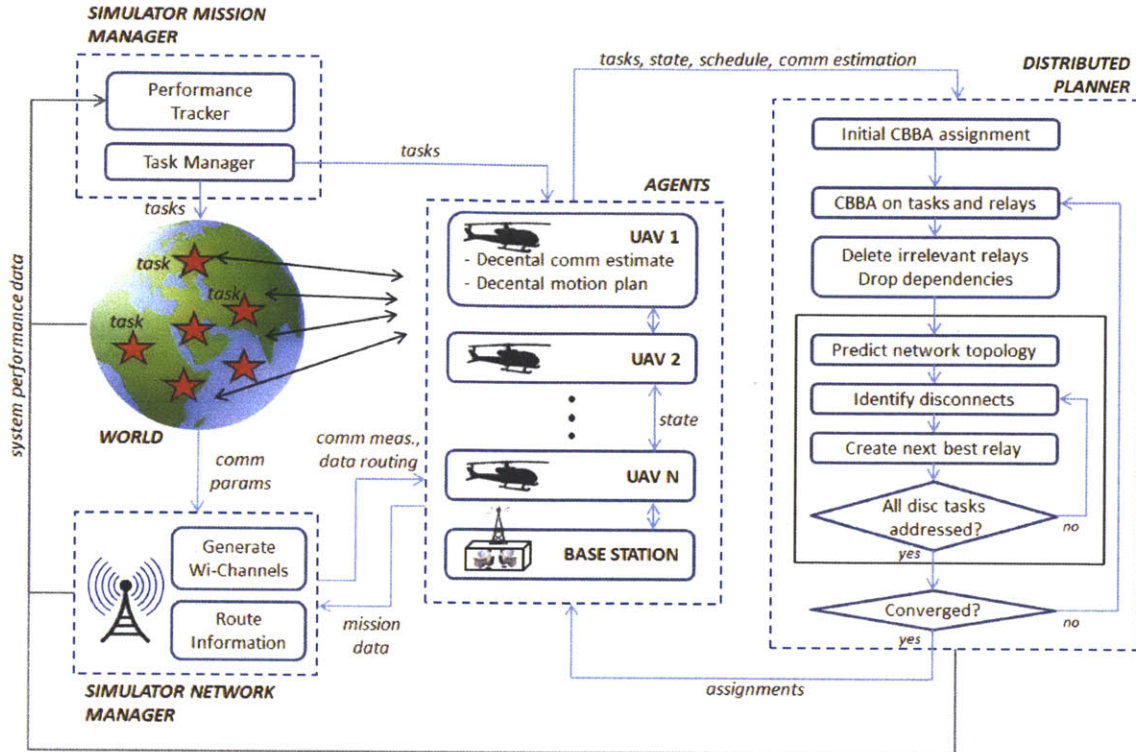


Figure 4-5: System framework block diagram

system performance metrics used for evaluation in experiments presented in Section 4.6 and Chapter 5.

- **Agents:** This module is separated into distributed submodules for each agent. Each submodule simulates its agent state, estimates the communication environment parameters using measurements from the network (see Sect. 4.5), and computes motion control inputs for the agent to execute its tasks or maintain connectivity with other agents (see Sect. 4.4). It also determines when the agent is ready to start its task.
- **Distributed Planner:** This module runs the task assignment routine using the distributed CBBA with Relays algorithm described in this section.
- **Simulator Network Manager:** This module generates wireless communication channels using the models described in Section 4.2.1 and manages routing of information according to protocols presented in the next two sections. The

separation of the planning module and the network manager module is key. The planner uses the model of the communication environment estimated by the planning agent to formulate a task assignment plan. It does not however actively control network routing during actual task execution, and the network could therefore behave differently than planned. This is important for two reasons: (1) wireless channels may vary significantly over short time-scales so the ability to adapt to these unplanned changes through routing is important [10], and more importantly (2) the multi-vehicle system may not have dynamic control of information routing and is therefore at the mercy of the protocol behavior.

4.3 Supporting Data-Rate Requirements

In order to transmit live sensor data back to the base, the multi-UAV network must ensure information routes have sufficient end-to-end channel capacity at the time of execution. The algorithmic approach introduced in Section 4.2.2 is used as a foundation for the planning strategy and is tailored based on the data-rate requirement and information routing protocol. Many protocols exist, along with many variations, each with strengths and weaknesses [61–68]. As such, this section includes two different representative routing strategies: (1) the model of a protocol representative of shortest path routing schemes commonly used in decentralized wireless networks, and (2) a protocol providing the optimal routing solution based on the network topology, formulated assuming centralized global real-time knowledge of the network. The purpose is to show the commonality in the planning strategies and provide performance bounds of the system as future, better protocols become available.

4.3.1 Realistic Routing

A model for the Ad-hoc On-Demand Distance Vector (AODV) [63] protocol (see Sect 2.1.4) is implemented to represent real-world networking dynamics. AODV employs local information at every node to dynamically compute the shortest transmission route according to an established metric. This protocol can rapidly adapt to

changes in topology, but also tends to converge to commonly used shortest routes in multi-user scenarios leading to data bottlenecks [67]. The metric employed in this thesis is a representation of the *airtime* metric popular for these applications and described by Eq. (2.11) [69]. To keep the model simple, general, and representative, its parameters are set to: $O = 0$, $B_t = 1$, $p_{pkt} = 0$, and $f = u$ the wireless channel data-rate capacity. As such, the airtime cost metric C_A for transmitting between any two nodes i and j is simply $C_{A_{ij}} = \frac{1}{u_{ij}}$. To transmit information from node i to k , the protocol finds the path which minimizes $\sum C_{A_{ij}}$ over any number of hops. This identifies routes with high data-rates, but also simultaneously minimizes the number of hops to reach its destination, which is a desirable behavior especially in high-bandwidth transfers such as live video [10].

The protocol is used in planning to predict how information will be routed from each active agent to the base. Route discovery is performed in simulation using the well known Dijkstra’s shortest path algorithm [92]. The predicted network graph \mathcal{G} at the time of task execution (see Algorithm 3, line 18) is used as input with active agents as vertices, and existing wireless links as edges with respective cost $\frac{1}{u_{ij}}$. Dijkstra’s algorithm is fast, and therefore ideal to compute the shortest *airtime* path for all agents to the base. Then, because each task executed requires data to flow to the base at a certain rate f_{task} , the flows can be summed over each generated path to determine the channel capacity requirement between each node as:

$$u_{req_{ij}} = \sum_k f_{ki} + f_{task_i} \quad (4.3)$$

Predicted data-rate violations are identified when $u_{req_{ij}} > u_{ij}$. This information is then used by the planner to decide: (1) which tasks are allowed to be executed given the present configuration (Box 4 in Fig. 4-3), and (2) how best to create relays to support disconnected tasks (Box 5 in Fig. 4-3).

To choose which tasks are allowed or disconnected given an unsupported graph is more complicated than when considering connectivity alone. Nodes (task locations) may be connected in the base *subnetwork*, but contribute to data-rate violations as in

Fig. 4-1. Since the mission objective is to maximize team reward, a selection process is required to find the combination of proposed tasks which provides the highest reward without violating data-rate requirements. To avoid an exhaustive search over the set of all tasks and to reduce the number of network recomputations when discarding tasks, Algorithm 5 leverages the *Minimum Spanning Tree* (MST) network graph structure produced by AODV routing. The tree is rooted at the base and each node sends its data to a single other node as shown in Fig. 4-4. The algorithm therefore greedily prunes lowest value leaves (or end node tasks) that are up-link of edges with data-rate violations (Algorithm 5, line 9). If all end-nodes are connected tasks being executed (in \mathcal{J}_{active}) which were carried over from a previous planning cycle, the algorithm prunes the furthest (number of hops) and lowest value task up-link of a violation not in \mathcal{J}_{active} . In this case, routing must be recomputed with the remaining graph. This process repeats until all remaining tasks are supported by the network.

Once the set of disconnected tasks is identified, the planning framework uses Algorithm 4 to reconnect them with relays. Line 10 in the algorithm uses the known required link capacity $u_{req_{ij}}$ to satisfy flow requirements on the lowest cost route, along with Equations (2.6) and (4.1) to solve for the maximum inter-node distance d_{max} to meet the requirement. The number of required relays and equally spaced positions is computed ($N_r = \lceil \frac{d_{ij}}{d_{max}} \rceil$). Once relays are created, information routing, which may now have changed for other nodes, is recomputed to identify possible new link violations. This process repeats until all tasks in the configuration are connected or designated infeasible.

Because one of the objectives of AODV is to minimize the number of hops, relays placed too close to one another to support a high data-rate requirement may actually skip over each other on a lower cost route. Since information is now transmitted over greater distances than intended, data-rate requirements may once again be unsupported. This occurs specifically when the total cost of 2 hops each over a shorter

Algorithm 5 FIND-DISCONNECTS-DATARATE-AODV($\mathcal{J}_{active}, \mathcal{G}$)

```

1:  $\mathcal{J}_{disconnected} \leftarrow j \notin \mathcal{G}_{sub-base}$ 
2:  $\mathcal{G}' \leftarrow \mathcal{G} \setminus \mathcal{J}_{disconnected}$ 
3:  $(U_{req}, \mathcal{P}) = \text{ROUTE-DATA}(\mathcal{G}', \text{AODV})$ 
4:  $\mathcal{J}_m \leftarrow \arg(u_{req_{m,n}} > u_{m,n})$ 
5: while  $\neg(\mathcal{J}_m = \emptyset)$  do
6:    $m = \arg \max(\mathcal{P}_{j,base}, j \forall \mathcal{J}_m)$ 
7:   while  $u_{req_{m,n}} > u_{m,n}$  do
8:      $\mathcal{J}_{candidate} \leftarrow \text{Find-Children}(m, \mathcal{P}, \mathcal{J}_{active})$ 
9:      $j_{del} = \text{Select-Node-to-Delete}(\mathcal{J}_{candidate})$ 
10:     $\mathcal{J}_{disconnected} \leftarrow \mathcal{J}_{disconnected} + j_{del}$ 
11:     $\mathcal{G}' \leftarrow \mathcal{G}' \setminus \{j_{del}\}$ 
12:     $(U_{req}, \mathcal{P}) = \text{ROUTE-DATA}(\mathcal{G}', \text{AODV})$ 
13:  end while
14:   $\mathcal{J}_m \leftarrow \arg(u_{req_{m,n}} > u_{m,n})$ 
15: end while
16: return  $(\mathcal{J}_{disconnected})$ 

```

distance is greater than a single hop over longer distance:

$$\frac{1}{u_{ij}} + \frac{1}{u_{jk}} \geq \frac{1}{u_{ik}}$$

or

$$2W \log_2 \left(1 + \frac{KP \left(\frac{d_o}{2d_{ij}} \right)^\alpha}{N_o W} \right) \geq W \log_2 \left(1 + \frac{KP \left(\frac{d_o}{d_{ij}} \right)^\alpha}{N_o W} \right) \quad (4.4)$$

Solving for $d_{Rmin} = d_{ij}$ above provides the minimum distance between relays below which they skip each other. Given this model, relays are never placed closer than d_{Rmin} to one another. If this does not satisfy data-rate requirements, then the algorithm attempts to reroute information by connecting the disconnected task directly to the base using relays. If that fails, the task is designated infeasible in this network configuration.

4.3.2 Optimized Routing

In the previous subsection, a popular routing protocol was embedded in a planner to identify network violations and create relays to meet data-rate requirements. However, AODV and similar protocols have certain limitations [67], and new protocols

providing better performance are the subject of constant investigation [71–73]. In order to provide a performance bound of the planning framework independent of the intricacies of specific routing protocols, a mechanism simulating optimal data routing is proposed. This simulated protocol assumes perfect real-time knowledge of the global network topology, and precise flow control node to node. These centralized assumptions may be difficult to implement in the real-world, but are useful to represent the upper-bound performance of routing protocols given this communication model. There are two main distinctions between the optimal routing protocol and the AODV model previously discussed:

- In the optimal protocol a transmitting node can split its data flow to go out to multiple other nodes, rather than sending all its data to a single other node as in AODV. This is subject to a total transmission constraint described later in this section.
- The optimal protocol simultaneously optimizes transmission requirements of all active nodes to distribute information flow through available links. In AODV, each node routes its information on its best path independently from other nodes, leading to possible bottlenecks.

The routing mechanism is formulated as a Linear Program (LP) solving a network flow optimization problem inspired by methods in [92]. The LP in Eq. (4.5) solves the minimum cost flow f from all task nodes to the base node along available edges (links), subject to communication constraints. A slack variable ϵ is added to the cost function to allow the network to *overload* some of its link capacities as needed to reach a solution to the problem. In the formulation, $\sum_j \epsilon_{ij}$ is the percent overload of an agent, and can be used to identify data-rate bottlenecks in the network. The main cost function c can vary based on the application. In this study this cost function is simply set to $c_{ij} = 1$ to again incentivize reducing the number of hops from a node to the base [71]. The cost function for ϵ , $\frac{M}{u_{ij}}$ is designed to deter links from overloading unnecessarily by setting M to a large value, but incentivize fewer stronger links to be overloaded if needed.

The first LP constraint (Eq. (4.6)) describes the conservation of flow principle true for all closed networks: the sum of flows into a node i , or $\sum_j f_{ji} + f_{task_i}$, equals the sum of the flows out $\sum_j f_{ij}$ [92]. Relay tasks do not generate flow of their own and as such $f_{task_{relay}} = 0$. The base station is treated as a sink for all incoming flows $\sum_i f_{task_i}$. The next constraint, (Eq. (4.7)), is the fundamental reason why the simpler Max Flow problem formulation [92] common in network optimizations cannot be used for this type of wireless network. Every node has a total transmission constraint. If it transmits at full available data-rate to a single other node, it is fully utilizing its resources to do so. In order to split its flow out, it must reduce its transmission rate to the first node to award channel utilization resources to others. This is described through the total link utilization ratio constraint $\sum_j \frac{f_{ij}}{u_{ij}} \leq 1$ [71] and incorporates the slack variable ϵ to allow that agent to overload if needed. Finally Eq. (4.8) and (4.9) constrain flows and overloads to be positive and bounded by capacity, with C as a constant to tune the maximum percent overload allowed on each link.

$$\arg \min \quad \sum_i \sum_j \left(c_{ij} f_{ij} + M \frac{1}{u_{ij}} \epsilon_{ij} \right) \quad (4.5)$$

$$\text{s.t.} \quad \sum_j f_{ij} - \sum_j f_{ji} = \begin{cases} f_{task_i} & \text{if } i \text{ is an active task} \\ 0 & \text{if } i \text{ is a relay} \\ -\sum_i f_{task_i} & \text{if } i \text{ is the base} \end{cases} \quad (4.6)$$

$$\sum_j \frac{f_{ij}}{u_{ij}} - \epsilon_{ij} \leq 1 \quad (4.7)$$

$$0 \leq f_{ij} \leq C u_{ij} \quad (4.8)$$

$$0 \leq \epsilon_{ij} \leq C \quad (4.9)$$

The LP formulated above is executed using a MATLAB solver. While the problem size allows a solution to be computed in real-time, the runtime is significantly slower than route computation using Dijkstra's algorithm. As such, the intent to reduce the number of routing computations during planning also holds when using this protocol. To select tasks that are disconnected, the planner runs Algorithm 6 which

Algorithm 6 FIND-DISCONNECTS-OPTIMALROUTING($\mathcal{J}_{active}, \mathcal{G}$)

```
1:  $\mathcal{J}_{disconnected} \leftarrow j \notin \mathcal{G}_{sub-base}$ 
2:  $\mathcal{G}' \leftarrow \mathcal{G} \setminus \mathcal{J}_{disconnected}$ 
3:  $(U_{req}, \mathcal{P}) = \text{ROUTE-DATA}(\mathcal{G}', \text{LP})$ 
4:  $\mathcal{J}_m \leftarrow \arg(u_{req_{m,n}} > u_{m,n})$ 
5: while  $\neg(\mathcal{J}_m = \emptyset)$  do
6:    $\mathcal{J}_{candidate} \leftarrow \text{Find-Children}(\mathcal{J}_m, \mathcal{P}, \mathcal{J}_{active})$ 
7:    $j_{del} = \text{Select-Node-to-Delete}(\mathcal{J}_{candidate})$ 
8:    $\mathcal{J}_{disconnected} \leftarrow \mathcal{J}_{disconnected} + j_{del}$ 
9:    $\mathcal{G}' \leftarrow \mathcal{G}' \setminus \{j_{del}\}$ 
10:   $(U_{req}, \mathcal{P}) = \text{ROUTE-DATA}(\mathcal{G}', \text{LP})$ 
11:   $\mathcal{J}_m \leftarrow \arg(u_{req_{m,n}} > u_{m,n})$ 
12: end while
13: return  $(\mathcal{J}_{disconnected})$ 
```

is a slight modification to the version presented in the previous subsection. First, for planning purposes only, a spanning tree is created using the largest flow out of each node. This helps identify nodes which contribute most to overloaded links by summing slack values ϵ along their path to the base. A greedy process again is used to sequentially prune nodes until the network is supported, starting with end nodes. Unlike in Algorithm 5, the network routing must be recomputed after each node is removed because flow is distributed across the network. To minimize the number of recomputations, the next node to delete is not selected simply based on lowest value, but also by considering a weighting factor of its total slack contribution, or overload. The remainder of the process is identical to the AODV routing protocol.

Once disconnected tasks are identified, relays are created using the process as presented in the previous subsection with some notable differences. First, since network routing here does not consist of a spanning tree, the search space for optimal relay placement is reduced by connecting disconnected tasks directly to the base. This generally requires more relays at each placement, but also reconnects more tasks per step since data is efficiently routed in a convex task space. Second, relay and task dependencies cannot be determined explicitly based on routing paths since all links are inter-dependent. As such, relays and tasks added at every step in the algorithm are dependent on relays created in previous steps.

4.3.3 Implementation into Framework

The two routing protocols described in this section were both implemented in the simulation network manager described in Section 4.2.3. At every simulation time step, the module uses agent locations to compute the network topology, the resulting network routing, and to determine whether tasks being executed are supported and thus generate reward. Even though the same models are used in planning as in mission execution, actual routing performance may vary during execution for several reasons:

- Stochastic fading during execution cannot be deterministically predicted.
- Communication model parameters may be unknown to the agents, vary over time, and differ from agents' estimates during planning.
- The network prediction in planning only accounts for agents committed to doing tasks. Agents traveling or temporarily idle can cause unplanned deviations in network routing during execution. If using a protocol like AODV, this can temporarily overload paths even in a deterministic environment. However, as will be shown in the simulation results, the LP formulation guarantees network support since it eliminates this effect by balancing network load.

Despite these limited guarantees, simulation results at the end of this chapter, as well as flight test results in the next chapter, show good performance of these planning strategies.

4.4 Supporting Error-Rate Requirements

In addition to satisfying data-rate requirements, the multi-UAV network must also ensure that agents can adequately exchange command and control messages. The Bit-Error-Rate (BER) is used as a performance metric for this requirement, and is modeled as in Section 4.2.1 for every link in the network graph. As data travels over multiple hops to its destination, each hop has a probability of error. Since

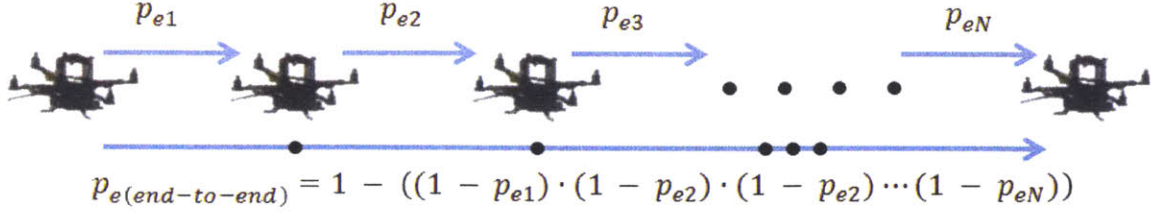


Figure 4-6: Probability of messaging error hop to hop

agents at each hop act as regenerators, and fully decode, interpret (to use the data themselves), recode, and resend messages to the next hop, the end-to-end probability of error increases at every hop as modeled in Fig. 4-6 [11]. This section describes how the task allocation network control strategy can be tailored to ensure inter-agent BER requirements are met for agents executing tasks. In addition, a reactive motion control strategy is provided to maintain agents not active during these task executions below the BER threshold for degraded communications with all other agents.

4.4.1 Inter-Agent Routing

The planning strategy to ensure BER requirements are met during task execution follows the same general approach described in Section 4.2.2. However, there are two important differences when planning for the BER compared to the data-rate requirement. First, the BER constraint is active between all agents and not just between every agent and the base station. Second, command and control messages are assumed to have negligible bandwidth requirements, so there is no constraint on how many agents can share a same routing path for these messages. Therefore, nodes can route messages independently of one another without risk of bottleneck.

Routing for these messages is also assumed to be independent of routing sensor data back to the base. However, the same AODV routing protocol can be used with a different metric to provide the set of optimal minimum BER routes \mathcal{P}_{ber} between agents. Since error probabilities are multiplicative over multiple hops, the additive cost of each hop is set to $C_{ber} = -\ln(1 - \text{BER}_{ij})$ [32, 33]. Dijkstra's Algorithm is used again to generate the minimum cost routes between all agents. BER violations occur if any pair of agents i and j have $\text{BER}_{ij} > \text{BER}_{thresh}$ which is the threshold maximum

Algorithm 7 FIND-DISCONNECTS-BER($\mathcal{J}_{active}, \mathcal{G}$)

```
1:  $\mathcal{G}' \leftarrow \mathcal{G}$ 
2:  $(BER, \mathcal{P}_{ber}) = \text{ROUTE-DATA}(\mathcal{G}', \text{BER})$ 
3:  $\mathcal{J}_{disconnected} \leftarrow \arg(BER_{j,base} > BER_{thresh}), \forall \{(j \in \mathcal{G}') \& (j \notin \mathcal{J}_{active})\}$ 
4:  $\mathcal{G}' \leftarrow \mathcal{G}' \setminus \mathcal{J}_{disconnected}$ 
5:  $(\mathcal{P}) = \text{ROUTE-DATA}(\mathcal{G}', \text{DATA-RATE PROTOCOL})$ 
6:  $\mathcal{J}_{end-node} \leftarrow \text{Find-Tree-End-Nodes}(\mathcal{G}')$ 
7:  $\mathcal{J}_{ber-violate} \leftarrow \arg(BER_{m,n} \notin \mathcal{J}_{active}, (m, n) \forall j \in \mathcal{G}')$ 
8:  $\mathcal{J}_{candidate} \leftarrow (\mathcal{J}_{end-node} \cap \mathcal{J}_{ber-violate})$ 
9: while ( $\mathcal{J}_{candidate} \neq \emptyset$ ) do
10:    $j_{del} = \text{Select-Node-to-Delete}(\mathcal{J}_{candidate})$ 
11:    $\mathcal{J}_{disconnected} \leftarrow \mathcal{J}_{disconnected} + j_{del}$ 
12:    $\mathcal{G}' \leftarrow \mathcal{G}' \setminus \{j_{del}\}$ 
13:    $\mathcal{J}_{end-node} \leftarrow \mathcal{J}_{end-node} \setminus \{j_{del}\}$ 
14:    $(BER, \mathcal{P}_{ber}) = \text{ROUTE-DATA}(\mathcal{G}', \text{BER})$ 
15:    $\mathcal{J}_{ber-violate} \leftarrow \arg(BER_{m,n} \notin \mathcal{J}_{active}, (m, n) \forall j \in \mathcal{G}')$ 
16:    $\mathcal{J}_{candidate} \leftarrow (\mathcal{J}_{end-node} \cap \mathcal{J}_{BER-Viol})$ 
17: end while
18:  $\mathcal{J}_{candidate} \leftarrow \mathcal{J}_{ber-violate}$ 
19: while ( $\mathcal{J}_{candidate} \neq \emptyset$ ) do
20:    $j_{del} = \text{Select-Node-to-Delete}(\mathcal{J}_{candidate})$ 
21:    $\mathcal{J}_{disconnected} \leftarrow \mathcal{J}_{disconnected} + j_{del}$ 
22:    $\mathcal{G}' \leftarrow \mathcal{G}' \setminus \{j_{del}\}$ 
23:    $(BER, \mathcal{P}_{ber}) = \text{ROUTE-DATA}(\mathcal{G}', \text{BER})$ 
24:    $\mathcal{J}_{candidate} \leftarrow \arg(BER_{m,n} \notin \mathcal{J}_{active}, (m, n) \forall j \in \mathcal{G}')$ 
25: end while
26: return ( $\mathcal{J}_{disconnected}$ )
```

allowable error rate for the system to be effective. Once BER violations are identified, the framework runs Algorithm 7 to greedily prune disconnected tasks. First, all tasks which violate BER with the base are pruned (line 3). Next, the algorithm considers the data-rate routing Minimum Spanning Tree, and prunes its low value end-nodes with BER violations in an effort to avoid changes in satisfactory data-routing (lines 5-17). Finally, if violations are still present, the algorithm greedily selects low value nodes to remove (lines 18-25).

Disconnected tasks are then then greedily repaired using the general approach of Algorithm 4. In line 10, the process sequentially adds relays along the computed shortest BER path between the two neighboring nodes with weakest link until the threshold is met. The algorithm terminates when all node pairs meet threshold or are labeled as infeasible.

Although this algorithm can operate alone to ensure active agents satisfy BER requirements, the intent in this chapter is to run it in addition with the data-rate planning protocol from the previous section. The same general approach is used to jointly satisfy both requirements. The algorithm first iterates between identifying nodes violating data-rate requirements, and then BER requirements, until it converges on a graph supporting both. Next, the algorithm greedily creates relays positioned based on the type of violation (starting by repairing data-rate) using the same sequential process.

4.4.2 Reactive Motion Control

The planning framework described in the previous subsection guarantees that agents that are busy executing tasks or relays meet interconnectivity BER requirements. This ensures agents performing cooperative tasks, such as target tracking or perimeter monitoring, can effectively communicate with each other and the base during execution. However, this process does not ensure other agents not busy executing tasks at the time of network prediction are connected. It is necessary, however, to have the entire fleet properly interconnected during the task allocation planning process. Poor inter-communications during the CBBA algorithm can significantly degrade the outcome of the plan [52, 53].

In order to drive these other agents, which are either idle or traveling to their next task, to an interconnected state, a decentralized reactive motion control policy is proposed. Its objectives are to minimize additional fuel costs incurred from extra travel, but also ensure that agents can satisfy their proposed plans. This second point is key and is discussed later. Each agent i uses its current knowledge of other agent locations and communication environment parameters to locally estimate the network topology and BER values with other agents. The communication environment may include uncertainty which is then treated as described in Sect. 4.5. Next, the agent determines if it is at risk of violating BER with any other agent m if $BER_{im} > BER_{plan}$, where BER_{plan} is a tunable planning threshold less than BER_{thresh} . As such, the goal is to maintain $BER_{im} \leq BER_{plan} < BER_{thresh}$ with all other agents and prevent unaccept-

able messaging error-rates. It then plans to reconnect with the agent most at risk of being disconnected with, above this threshold, through the estimated minimum BER route. It computes its minimum distance $d_{min_{ij}}$ to the next agent j in the route needed for $BER_{im} = BER_{plan}$ and starts traveling to meet that requirement. If that distance $d_{min_{ij}}$ is less than a safety threshold for collision avoidance, it skips on toward the next agent k in the BER route. This can occur if j is currently busy executing a task and cannot move itself to maintain connectivity with other traveling agents. As agents move about the environment, $d_{min_{ij}}$ is dynamically updated to steer i .

Each agent executes this policy until it needs to break off to travel to the next task to ensure it starts on-time as planned. This is critical because each task is part of network predictions used to ensure communications are supported. This prediction is invalidated if the task is not executed as proposed. Because of this constraint, and because agents do not account for locations relative to each other in continuous time during planning, it is not possible to guarantee all agents will be interconnected at all times using this framework. However, results in Section 4.6.3 and 5.4.2 show good performance of this strategy both in simulation and outdoor flight tests. This type of decentralized reactive motion control policy could be implemented with periodic connectivity constraints as done in [25] to ensure agents are connected at known times of task allocation replanning.

4.5 Planning with Uncertainty

Shadowing and multipath fading lead to uncertainty in wireless channel performance as illustrated by the samples in Fig. 4-2. In addition, the environment in which the communication system operates can be dynamic in obstacles, noise, and interference from other sources, and further vary network performance. This uncertainty must be considered in planning to ensure mission execution is robust to degradations in the environment.

The fundamental parameter in wireless channel capacity and BER performance is the Signal to Noise Ratio (SNR). SNR_{dB} is commonly modeled with normally

distributed shadowing and fading distortions (see Section 2.1.2). In this thesis, these values are generalized into value $\gamma_{f_{dB}}$ sampled from $\mathcal{N}(0, \sigma_{dB}^2)$. If the fading distribution variance is known, conservatism can be added to the planning process to drive the system performance to a desirable confidence level. This type of *risk adjustment* is often employed to add robustness in planning for multi-agent systems [98]. The SNR used to predict the network, identify disconnected tasks, create relays, and plan connectivity motion can be degraded by a certain amount to ensure a confidence level of its minimum performance. In Eq. (4.10) K_m is used as a multiplier of σ_{dB} to tune the confidence level. $K_m = 0$ is the center of the normal distribution, and as such offers 50% confidence that $\gamma_{ij} > \gamma_{plan}$, whereas $K_m = 1$ offers 84% confidence.

$$\gamma_{plan_{dB}} = 10 \log \left(\frac{P}{N_o W} \right) + K_{dB} - 10\alpha \log \left(\frac{d}{d_o} \right) - (K_m \sigma_{dB}) \quad (4.10)$$

The impact of varying K_m on system performance is explored in a Monte Carlo simulation in the next section, and used in outdoor flight testing with a team of UAVs to plan in uncertain communication environments in the next chapter.

A common problem encountered in field operations involves predicting the performance of wireless communications [18, 35, 42]. Often the attenuation, fading, and interference levels are initially unknown, and can vary spatially and temporally during mission execution [46]. Previous efforts have attempted to sample the communication environment a priori to construct a *radio-map* indicating areas of high network performance. However, these were then undercut by changes in performance during actual mission execution [18]. Therefore, this thesis turns to a dynamic real-time communication model estimation method for the multi-UAV system to initially learn the communication environment, sense changes during execution, and adapt its planning strategy accordingly.

The Least-Squares sample estimation method described below was obtained from [32] and adapted for this application. It assumes that over the course of mission execution, agents collect and share SNR samples with each other. These measurements are then used to estimate the global environment communication parameters

for planning purposes. The SNR sample γ_s can be rewritten as Eq. (4.11), where Φ is defined in Eq. (4.12), and $\gamma_{f_s dB}$ is the stochastic contribution to SNR due to fading in that sample.

$$\gamma_{s dB} = \Phi - 10\alpha \log\left(\frac{d_s}{d_o}\right) - \gamma_{f_s dB} \quad (4.11)$$

$$\Phi = 10 \log\left(\frac{P}{N_o W}\right) + K_{dB} \quad (4.12)$$

It is reasonably assumed that each agent knows the transmission power P , bandwidth W , and equipment gain K_{dB} used in the sample. However, environment parameters such as the path loss exponent α , fading variance σ_{dB}^2 , and noise power density N_o are unknown, may vary, and need to be estimated. A set of k samples γ_s accessed by an agent can be stacked into Γ , as in Eq. (4.13) where G_q , θ , and Γ_f are defined in Eq. (4.14).

$$\Gamma = G_q \theta + \Gamma_f \quad (4.13)$$

$$G_q = \begin{bmatrix} 1 & -10 \log\left(\frac{d_1}{d_o}\right) \\ 1 & -10 \log\left(\frac{d_2}{d_o}\right) \\ \vdots & \vdots \\ 1 & -10 \log\left(\frac{d_k}{d_o}\right) \end{bmatrix}, \quad \theta = \begin{bmatrix} \Phi \\ \alpha \end{bmatrix}, \quad \Gamma_f = \begin{bmatrix} \gamma_{f_1 dB} \\ \gamma_{f_2 dB} \\ \vdots \\ \gamma_{f_k dB} \end{bmatrix} \quad (4.14)$$

Estimation of the communication parameters α , σ_{dB} , and N_o is then performed using the LS regression procedure in Eq. (4.15) [32]:

$$\begin{aligned} \hat{\theta} &= (G_q^T G_q)^{-1} G_q^T \Gamma \\ \Gamma_{G_q} &= \left(I_k - (G_q^T G_q)^{-1} G_q^T \right) \Gamma \\ \hat{\sigma}_{dB} &= \sqrt{\frac{1}{k} \Gamma_{G_q}^T \Gamma_{G_q}} \end{aligned} \quad (4.15)$$

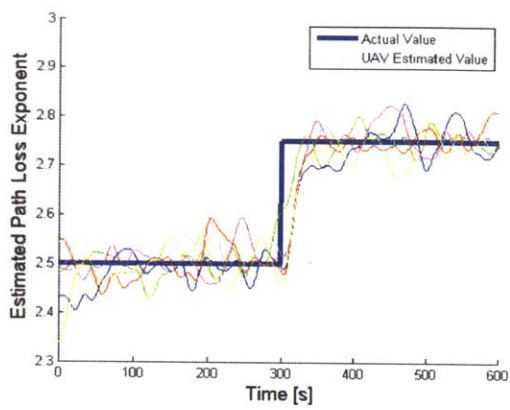
A weighted moving average filter is then applied to the estimated values to smooth out measurement disturbances and allow estimated values to vary over time with changes in the environment. As implemented, each agent performs this estimation over a

different set of measurements in a fully decentralized setting, and may have different parameter estimates from other agents. These estimates are then used in the following stages of planning: network topology prediction (Fig. 4-3 Block 3), identification of disconnected tasks (Block 4), relay placement (Block 5), and reactive motion control planning to maintain inter-agent BER. Fig. 4-7 shows the estimation profile of the three initially unknown communication parameters over the course of a 600s mission with six UAVs. Parameters are initially set to those listed in Table 4.1, and then each increased by 10% halfway through the scenario. Estimates for α (Fig. 4-7(a)) and N_0 (Fig. 4-7(b)) are used to estimate expected SNR performance. Figure 4-7(c) is a profile of the estimated versus actual γ_{dB} at a distance of 100m. Agents are able to detect 90% of the change in expected γ_{dB} performance in an average $t_{90\%} = 37.8s$ (on a 1s estimation cycle) and track the new value with $\bar{e} = 7.4\%$ mean error. For σ_{dB} , which is used to plan conservatively, $t_{90\%} = 15.2s$ and $\bar{e} = 1.5\%$. The ability to detect and adapt to these changes improves system performance as will be shown in Section 4.6.5.

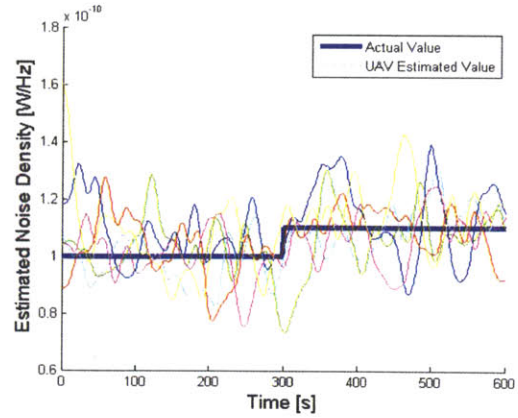
4.6 Performance Evaluation

4.6.1 Experiment Overview

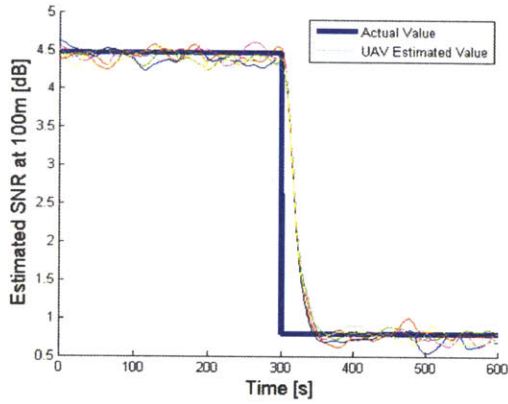
Several Monte Carlo simulation experiments were conducted to evaluate system performance and observe trends in cooperative behaviors using the proposed framework. Experiments consisted of 30 scenario trials per data point using the mission parameters listed in Table 4.2 and the communication model parameters from Table 4.1 (with variations as listed). The same 30 scenarios were used across each data point for direct comparison. In all experiments, reward was only generated when data-rate requirements to the base were supported at each time step. In other words, if a specific task was only supported 50% of the time during its execution, it received 50% of its reward. The following four experiments were conducted:



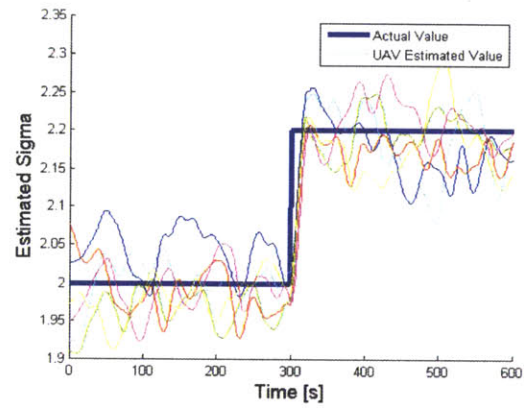
(a) Path loss α



(b) Noise floor N_0



(c) Signal to Noise Ratio (γ_{dB})



(d) Fading variance σ_{dB}

Figure 4-7: Estimation of simulated communication environment with $N = 6$ agents

- Varying the level of incentive for relay tasks in the CBBA task assignment algorithm
- Comparing different communication planning strategies: (1) considering connectivity only, (2) planning for data-rate using the AODV routing protocol, (3) planning for data-rate using optimal routing, and (4) planning simultaneously for BER and data-rate
- Varying the level of planning conservatism given uncertain communication environments

Table 4.2: General Simulation Mission Parameters

Parameter	Value
Trials per Scenario	30
Mission length	600s
Replan Rate	60s
Task Creation Cycle	60s
New Tasks per Cycle	20
Environment Size	$350 \times 400 \times 50m$
Communication Radius R_{COMM}	30% Environment Max Distance
Task Data-Rate f_{task}	mean capacity at R_{COMM}
Threshold BER (BER_{thresh})	mean BER at R_{COMM}
UAV Cruise Velocity	8m/s
UAV Fuel Cost	0.1/m

- Evaluating the benefits of estimation for adaptive planning in variable communication environments

4.6.2 Varying Relays Incentive

This Monte Carlo simulation evaluates the impact of varying the level of incentive K_v for relays. This value is a multiplier on the *virtual task value* listed for each relay used in the agents' reward function during the CBBA bidding process (see Eq. (4.2)). In this experiment, K_v is varied from 0 for no incentive, up to 4. This experiment assumes a deterministic communication environment with no fading ($\sigma_{dB} = 0$) and each trial is executed with a team of six UAVs planning for data-rate only, using the realistic AODV routing protocol.

Figure 4-8 (a) shows the total mission score distribution for each setting (+'s represent 25% and 75% percentile values) and show that $K_v = 1$ offers the best planning performance. The reason can be seen in Fig. 4-8 (b), where the solid blue line represents the number of tasks executed, and the dashed red line shows the average task value. If no incentive is provided for the relay tasks, agents resort to executing lower value tasks near the base, similar to the conservative planning strategy discussed in Sect. 3.3, but executing more of them since they are not occupied by relaying. As

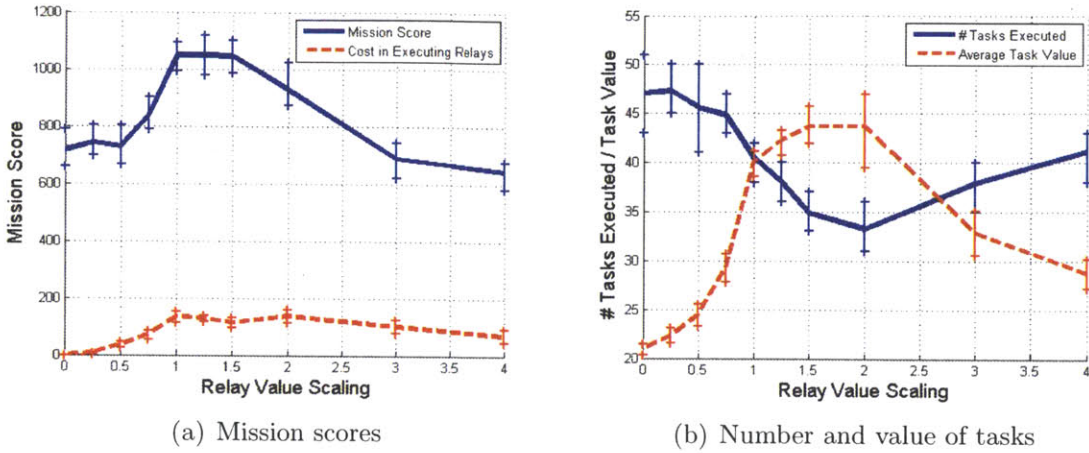


Figure 4-8: Varying the level of incentive for relay tasks

incentive increases, agents can perform higher value tasks using relays, and despite performing fewer tasks, generate higher total reward. This trend peaks and plateaus at $K_v = 1$. With even more incentive, agents continue to perform even higher value tasks, but now perform less of them which eventually results in lower scores. The peak in task value occurs at $K_v = 2$ after-which an interesting phenomenon in cooperative behavior occurs.

When relays become highly over-incentivized ($K_v > 2$), agents become overly interested in these tasks and always favor them in the bidding process over regular tasks. Agents then drop what ever task they had planned whenever a relay becomes available, which often invalidates earlier network predictions that over multiple iterations eventually created those relays. These relays then become *irrelevant* and are deleted, freeing up the agent. Because mechanisms are built into the algorithm to ensure these types of cycles eventually converge to a solution (see Section 3.3.2 and 4.2.2) agents tend to rule out plans with relays. As such, the task assignment trend reverses, and agents tend to perform more tasks which are less valuable as relay incentive is hyper-inflated. The performance however is significantly lower than with $K_v = 1$. This interesting behavior indicates that a proper balance must be struck for distributed agents to cooperate, since forcing the cooperation by over-incentivizing can have negative consequences.

4.6.3 Comparing Planning Strategies

These experiments compare the different planning strategies discussed in this chapter. Agents plan the mission considering (1) connectivity requirements only as was performed in the original CBBA with Relays framework, (2) data-rate requirements using real-world AODV routing protocol, (3) data-rate using optimal data routing, and (4) both BER and data-rate requirements while using the motion planning strategy to stay interconnected. To compare the planning effectiveness in supporting the network, these experiments assume a deterministic communication environment with no fading ($\sigma_{dB} = 0$). This assumption is removed in the subsequent experiments discussed in Sect. 4.6.4 and sec:mccomest. The number of UAVs in the team in these experiments is varied from 2 to 10.

Figure 4-9 shows the average total mission score using the different strategies. All strategies exhibit similar performance for small counts of agents. However as the team size grows, teams considering connectivity only, and not more accurate networking dynamics, quickly overload the network with their plans. As can be seen in Fig. 4-10 (a), agents planning for connectivity only without stricter requirements attempt more and higher value tasks. However Fig. 4-10 (b) shows these plans are too ambitious and often not supported by the network (as low as 38.6% average supportability for teams of 10 UAVs) which results in significant mission performance degradations compared to other strategies considering data-rate.

The same three figures also show the impact of information routing on performance. When information is routed optimally (in black) the team can make more effective task assignments attempting slightly higher value and greater numbers of tasks using the same overhead investment in relays. Real-world networking protocols such as AODV lead to data-rate bottlenecks that limit the total throughput of information, and in this case result in decreased mission performance. Most importantly, Fig. 4-10 (b) reveals a fundamental limitation of this overall communication control framework. Since network prediction is performed at discrete task execution times, as opposed to continuous time, the network is vulnerable to changes in rout-

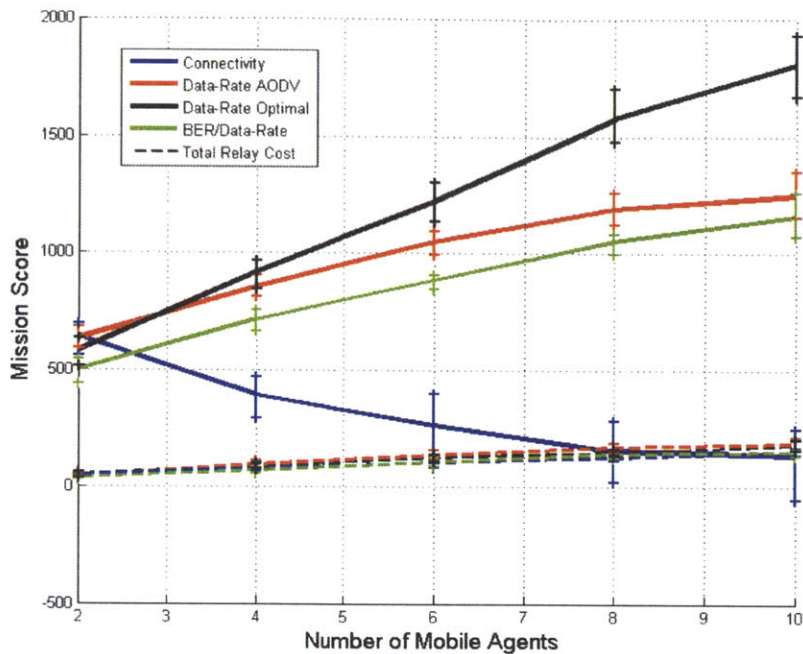


Figure 4-9: Mission performance for different connectivity planning strategies

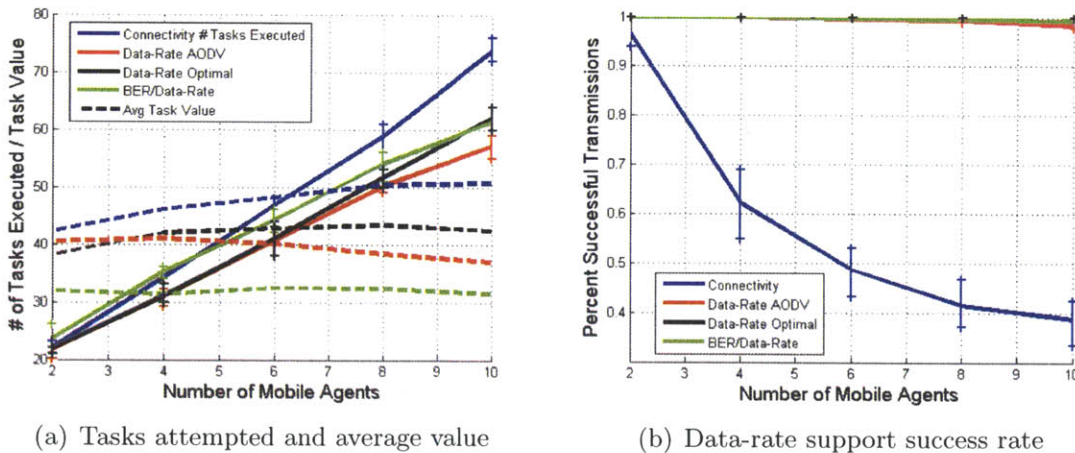


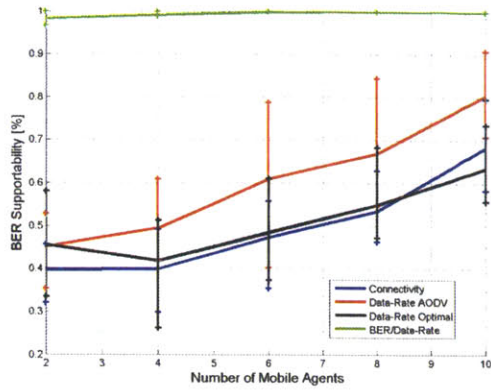
Figure 4-10: Task assignment supportability for different planning strategies

ing caused by agents not active during this prediction (such as agents traveling to tasks). When information is routed optimally, using centralized assumptions, plans generated are guaranteed to be supported by the network because routing can be balanced to account for these additional nodes (100% for optimal routing). However, when using real-world routing protocols, the additional inactive agents unaccounted for in the prediction may significantly change data routing, and create unanticipated

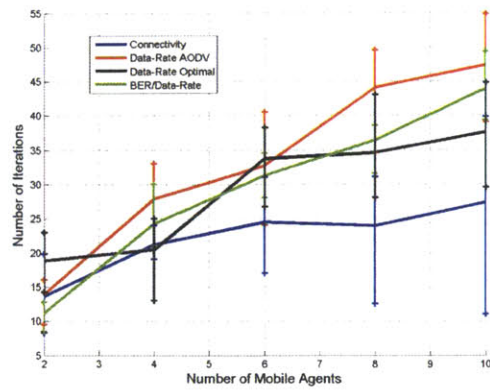
bottlenecks even in a deterministic world. While network supportability and plans are still good, with 98.4% average network support in the **worst** case, perfect performance guarantees are not possible with this framework and real-world protocols such as AODV.

Next, the experiment shows the performance of the system maintaining inter-agent connectivity. Figure 4-11 (a) shows that teams not considering inter-agent BER requirements spend significant amounts of time disconnected from one another during the mission. However as the team grows, and the environment becomes more densely populated with agents, this problem tends to decrease. On the other hand, if BER is considered as a requirement in addition to connectivity or data-rate alone, and inactive or traveling agents plan their motion properly, the team can maintain good inter-connectivity performance (98.2% in the worst case for small teams, and 99.95% with larger teams). Again, the framework cannot provide performance guarantees since agents formulate task assignments without considering continuous time motion based on other agents. A periodic connectivity strategy similar to that proposed in [25] could be implemented to ensure the team is fully connected at the time of re-planning. However, the framework does guarantee inter-connectivity of active agents performing cooperative tasks. Figures 4-9 and 4-10 (a) also show that the added inter-connectivity requirement further constrains the team to perform generally less valuable tasks which results in lower mission scores.

Finally, Fig. 4-11 (b) shows the number of CBBA with Relays planning iterations across the different planning strategies. Although planning iteration counts vary greatly depending on the replan, the trends suggest that the algorithm converges faster when information routing is not considered. Furthermore, as the number of agents increases, the number of required iterations also increases. When using the AODV protocol for data-rate consideration only, it takes an average 47 iterations of the process described by Fig. 4-3 to converge on a solution given 10 agents and > 20 tasks.



(a) Inter-connectivity success rate



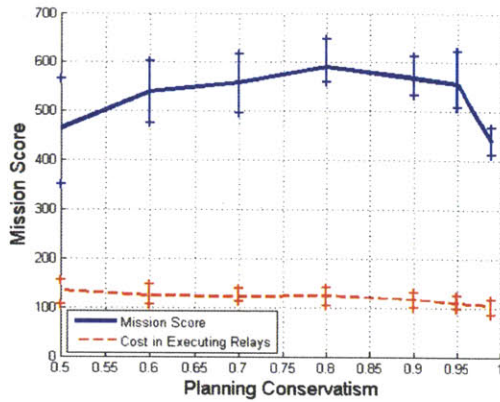
(b) Number of planning iterations

Figure 4-11: Inter-connectivity and algorithmic convergence performance comparison

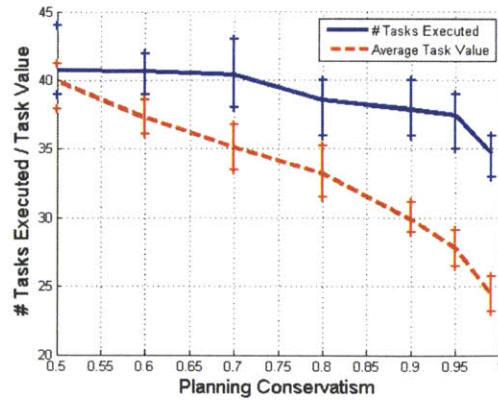
4.6.4 Varying Level of Planning Conservatism

In this next experiment, the level of planning conservatism is varied in a stochastically fading communication environment ($\sigma_{dB} = 2$). K_m from Eq. (4.10) varies across each trial to provide between 50% and 99% confidence that predicted communication links will support data-rate requirements. Each trial is executed with a team of six UAVs, planning for data-rate only, using the realistic AODV routing protocol. Here, agents are aware of communication parameters and therefore do not need to perform estimation.

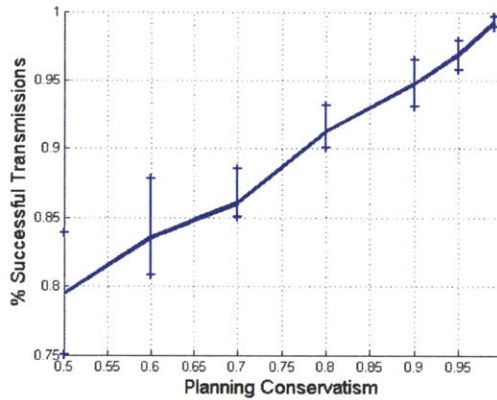
Results in Fig. 4-12 (c) show that even in a stochastic environment, the system can achieve high network reliability by increasing its level of conservatism, or reducing risk. This comes at a cost of performing fewer and lower value tasks closer to the base which results in decreased mission score. As the level of conservatism is relaxed, agents attempt more and higher value tasks and achieve higher scores despite more frequent data-rate violations. This trend peaks at 80% conservatism, below which agents become too careless in planning, and over strain the network with lofty goals resulting in lower data-rate supportability and lower mission scores. It is further observed when comparing these results with the previous experiment that adding uncertainty in the communication environment significantly decreases system performance (here a minimum 43.6% decrease in mission score for a team of six UAVs).



(a) Mission scores



(b) Tasks attempted and average value



(c) Data-rate support success rate

Figure 4-12: System performance with varying level of planning conservatism

4.6.5 Estimating the Communication Environment

The last Monte Carlo simulation evaluates the ability of agents to estimate unknown and possibly changing communication parameters and adapt the planning strategy accordingly (see Sect. 4.5). Two sets of trials are presented: (1) the communication environment improves half way through the mission (α , σ_{dB} , N_0 decrease 10%), and (2) the communication environment instead degrades (α , σ_{dB} , N_0 increase 10%). The missions are initiated using the baseline communication parameters listed in Table 4.1 with fading variance $\sigma_{dB} = 2$. Each trial is executed with a team of six UAVs unaware of the communication parameters, planning for data-rate only using the realistic AODV routing protocol.

Figure 4-13 shows an averaged mission execution profile with three settings: (1) agents have perfect up-to-date knowledge of the communication parameters (black line), (2) agents estimate parameters (blue line), and (3) agents have perfect knowledge of initial parameters but are unaware of the change (red line). The wavy shape of the reward profile is a result of the 60s replan cycle, where after each replan agents tend to travel to a new set of tasks leading to a temporary decrease in score. Results clearly show the ability of team to estimate changes in the environment and adapt the planning strategy to improve performance. In Fig. 4-13(a) the communication environment improves at 300s allowing agents to achieve more tasks of higher reward. Teams which do not adapt to this setting (red) maintain the original strategy and marginally improve the rate of reward gain because of more successful communication rates. On the other hand, teams equipped with perfect knowledge (black) adapt their planning to be more ambitious which generates increased reward. Teams estimating these parameters (blue) closely follow the trends obtained with perfect knowledge with a short lag due to planning cycle, and clearly outperform teams which do not adapt. Similarly in Fig. 4-13 (b) the communication environment degrades at 300s. Teams which do not adapt degrade in performance because of increased network violations, whereas teams with perfect knowledge propose more conservative plans and are able to keep generating reward. Once again, the teams estimating these parameters are able to closely follow the perfect knowledge case with a short lag, and outperform teams which do not adapt.

4.7 Summary

This chapter presented methods to improve the CBBA with Relays algorithm by considering realistic wireless communication dynamics and requirements. The original framework introduced in the previous chapter only accounted for connectivity between agents assuming a deterministic fixed communication radius. Here realistic models for channel data-rate capacity and probability of messaging error (BER) were introduced into the framework as real-world communication requirements. The

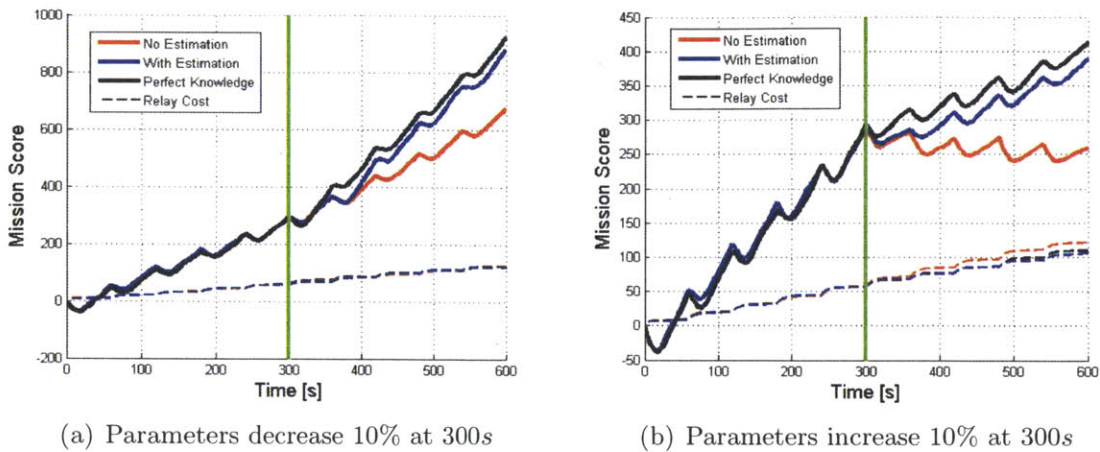


Figure 4-13: Adaptive planning to changes in the communication environment

data-rate requirement and protocol used to route information add an additional layer of coupling between tasks in this already combinatorial task assignment problem. A method to efficiently determine which tasks to prune from the network, and how to greedily create relays for support was presented and tailored to different routing algorithms. In addition, a method to jointly satisfy data-rate requirements as well as agent BER inter-connectivity requirements was enabled with a small change to the general planning strategy. Finally to overcome uncertainty in communication performance, which is a common problem in multi-robot operations, an estimation process of the wireless environment was incorporated. This allows teams to adapt to changes and plan according to a desired level of conservatism. Four Monte Carlo simulation experiments were conducted using this updated framework, and showed (1) improved performance over the baseline framework from Chapter 3, (2) the ability to satisfy realistic communication requirements, and (3) effective estimation and adaptive behavior in a changing communication environment.

Chapter 5

Flight Test Experiments

5.1 Overview

This chapter presents a set of outdoor flight test experiments showing dynamic mission planning for communication control of a multi-UAV team using CBBA with Relays. The objective is to show operational proof of concept of the work developed. Previous work has included an experiment using the CBBA with Relays algorithm conducted for a team of unmanned ground vehicles in the controlled setting of the MIT Real-time indoor Autonomous Vehicle test ENvironment (RAVEN) [93]. There are several challenges and ultimately benefits to implementing a distributed multi-agent system in an outdoor uncontrolled environment:

- Real-time operation: multi-agent distributed task allocation is a complex combinatorial problem. In outdoor flight testing, UAVs rely on real-time algorithmic execution to operate effectively and safely, and cannot tolerate to simply wait for the planning framework to reach a solution as it would in simulation.
- Imperfect environment: In simulation, vehicles behave according to modeled dynamics which usually match those predicted by the planner. Outdoors, vehicle dynamics may be different, are subjected to the environmental effects (wind, gusts, etc...), and exhibit non-linear behaviors unaccounted for in the algorithm. In addition, state estimates reported back to the planner include inaccuracies,

delays, or may arrive less frequently. These imperfections can have surprising effects on the system as discussed in the results of this chapter.

- System degradations: Systems operating outdoors are subject to uncertainties and degradations. Communication degradations between UAV's and the base station can significantly hinder the system from behaving as intended.

These factors must be considered in the design of an algorithm, and just as importantly must then be tested in an operationally relevant environment.

A number of outdoor multiple unmanned vehicle flight tests have been previously executed, which examined target tracking and air-ground robot coordination [111, 112], movement of a heavy load [113], formation flight [114], forest fire monitoring [115], and coordinated surveillance of a road [116] or a large ground surface area [117]. Past experiments have even shown dynamic distributed task assignment frameworks using market based algorithms similar in scope to the CBBA algorithm discussed in Sect. 3.2 [118]. On the other hand, very few of the multi-agent network communication control studies presented in Sect. 2.3 have performed field implementations of their work. In fact, authors in [34, 35] discuss the only known experiment where multiple UAVs are autonomously controlled outdoors specifically to meet cooperative communication objectives. The work presented in this chapter is therefore the only known implementation of a team of UAVs controlling network communication through distributed task assignment.

Two sets of flight experiments are presented in this chapter. First, the original CBBA with Relays framework was implemented and evaluated against the baseline CBBA algorithm which does not consider connectivity, and the conservative CBBA with Network Prediction strategy discussed in 3.3 (Experiment 1). Second, the improved framework presented in Chapter 4 was flight tested to compare planning strategies with more complex communication dynamics (Experiment 2.1) and evaluate the ability to estimate the communication environment and adapt to changes (Experiment 2.2). This chapter focuses on the implementation, test execution, and lessons learned.

5.2 Experimental Setup

This section describes the field test implementation of the multi-UAV architecture. Flight testing was conducted at a military base to take advantage of the site's restricted airspace. Operations took place in a large open remote area secure from the public.

5.2.1 System Architecture

The flight test architecture in this experiment interfaces three key components as seen in Fig. 5-1. The CBBA with Relays MATLAB framework shown in Fig. 4-3 serves as the center of the multi-vehicle system and is adapted to interface with other components. This module operates on its own computer and manages the mission scenario, executes the multi-UAV distributed planning algorithm, provides high level UAV motion requests, and in Experiment 2 simulates the communication environment against which the algorithm is evaluated. Simulated vehicles are replaced with a submodule to read-in vehicle states and send command requests to assigned tasks or for inter-connectivity motion control. A visualization interface leveraged from the previous indoor test [93] is incorporated into the framework and modified to provide operator feedback on the performance of the system (Fig. 5-2).

The MATLAB module is interfaced via a network hub with two other components of the system operating on a separate computer: (1) the UAV high level autopilot module, and (2) another operator station known as the *Safety Monitoring Interface* (SMI). Both are discussed in the following subsections. Messages are exchanged between modules using Lightweight Communications and Marshalling (LCM), a recently developed protocol based on UDP multicast [119]. LCM replaces traditional point-to-point UDP messaging by instead offering network communication channels. Software modules on the multicompiler network can subscribe to these channels, and post messages of specified format to send information, or listen on the channel and capture messages as they come in. This protocol as implemented mitigates the need to rebroadcast failed message exchanges, and allows multiple modules to simultane-

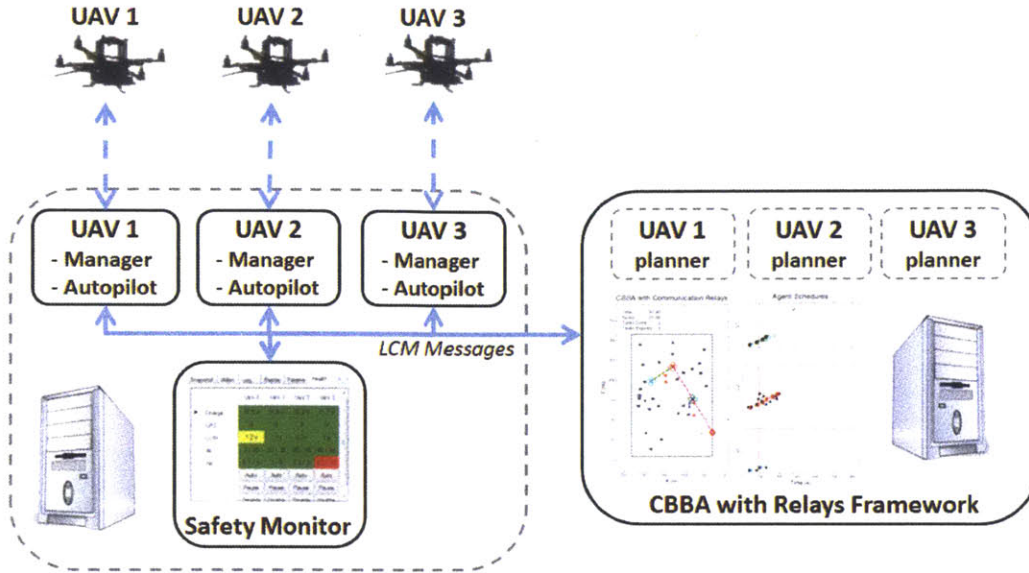


Figure 5-1: CBBA with Relays flight test architecture

ously access posted information. In this case, two sets of channels are created to pass UAV state information and send out waypoint command requests.

5.2.2 Quadrotor Unmanned Aircraft

The UAVs used in this experiment consist of three Ascending Technologies Pelican Quadrotors shown in Fig. 5-3(a). Each vehicle weighs 2.5 lbs and electrically powers four rotors using a 5000 mAh LiPo battery for flight endurances of up to 18 minutes. These aircraft are ideal for this type of research because of their ability to hover, to accurately track waypoints even in windy situations, their simple mechanical design, low footprint, and their relative ease of operation. Each quadrotor is capable of GPS waypoint navigation while communicating with the Ground Control Station (GCS) using a Digi-Mesh XBee 2.4 GHz radio module. These aircraft nominally travel at speeds of $5 - 8m/s$ and are set to achieve $2.5m$ waypoint accuracy. A C++ module developed for this experiment runs on the second computer of the architecture to interface with the vehicle in flight. This high level autopilot module requests state information from the vehicle using manufacturer libraries, converts and posts the information on LCM, and formats command requests from the planner into waypoints sent to the vehicle. A separate instance of this module runs for each quadrotor in

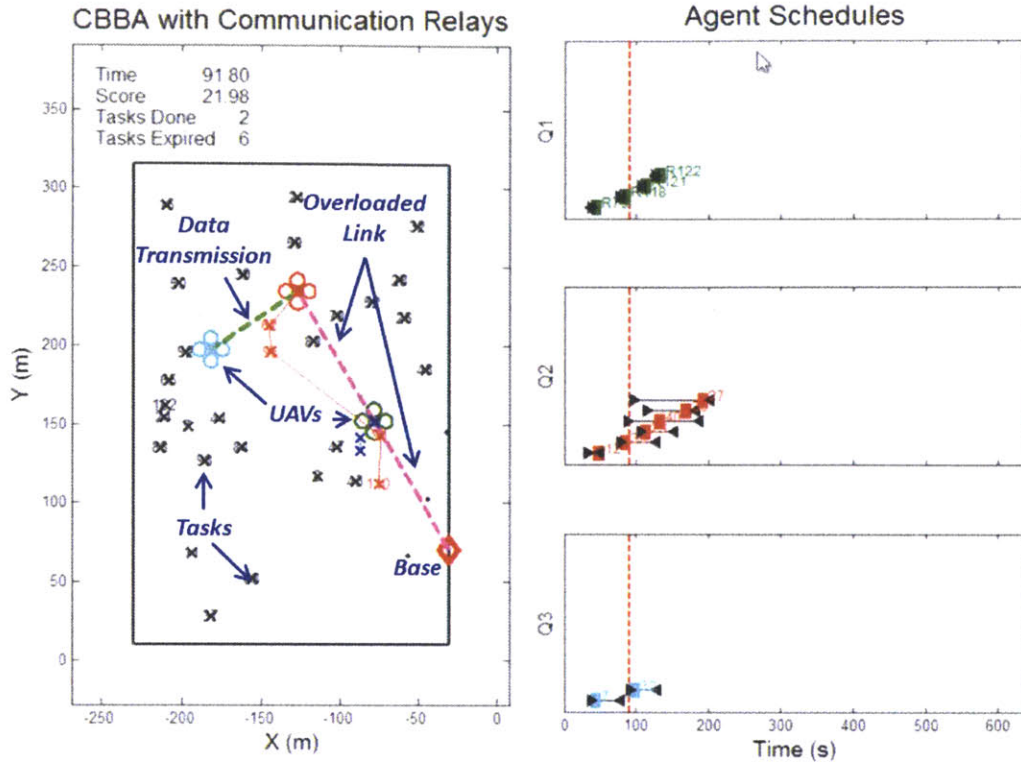


Figure 5-2: CBBA with Relays visual interface

operation. Despite their ability to operate autonomously, each vehicle is actively monitored by a safety pilot responsible for takeoff, transition to autonomous flight, and landing. Once the mission is initiated however, the UAVs are in full autonomous control, and safety pilots only intervene to avoid hazardous situations.

5.2.3 Safety

The Safety Monitoring Interface (SMI) was developed by Aurora Flight Sciences specifically for another flight test campaign described in Chapter 6. The purpose of the SMI is to allow an additional safety operator to monitor the health and status of the UAVs during the experiment, as shown in Fig. 5-3(b). The interface displays a moving map of the vehicles and information such as battery voltage, speed, altitude, GPS, and communication link strength which is color coded to quickly highlight hazardous states. The SMI operator can radio information to the safety pilots for situational awareness or manual takeover. In addition, the SMI has the ability to

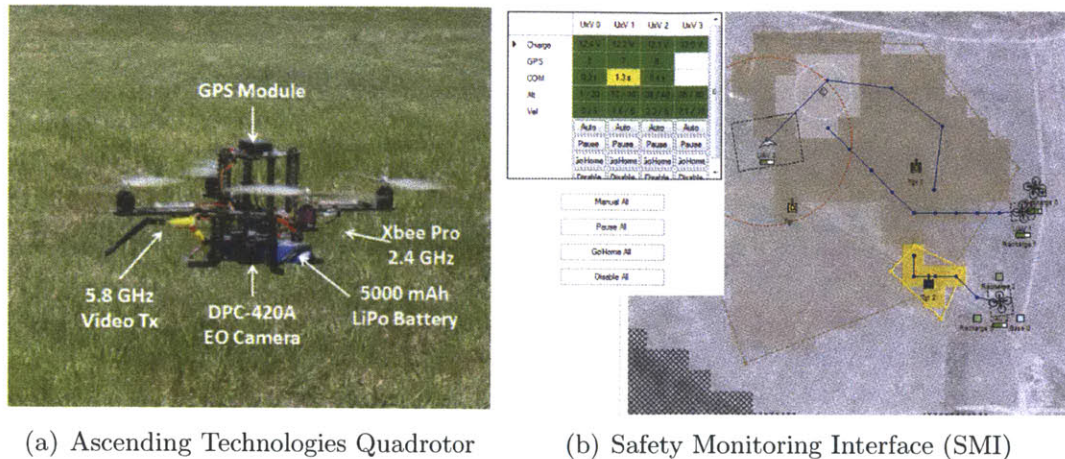


Figure 5-3: Flight test systems

disconnect a vehicle from the mission to prevent a vehicle from receiving unwanted waypoints. This functionality is discussed further in Chapter 6.

5.3 Experiment 1 - Original CBBA with Relays

The first series of outdoor experiments implemented the original CBBA with Relays framework in flight test. More information about the algorithms in this section is provided in Section 3.3.

5.3.1 Test Scenario

In the first test, the mission scenario consisted of gathering information in the environment using a team of three Pelican quadrotors. The communication radius R_{COMM} of the UAVs was set to 30% of the flight arena size, which was $300 \times 200 \times 50m$. To account for potential positional inaccuracies of the quadrotors (nominally $> 2.5m$) a $5m$ buffer was imposed on R_{COMM} for planning only. This generated slightly more conservative plans, but added robustness in the outdoor environment. In the scenario, an area of greater information interest was designated at distances greater than R_{COMM} from the base station, and occupied half of the flight arena (designated as *High Priority Area* in Fig. 5-4). Information gathering tasks were then randomly

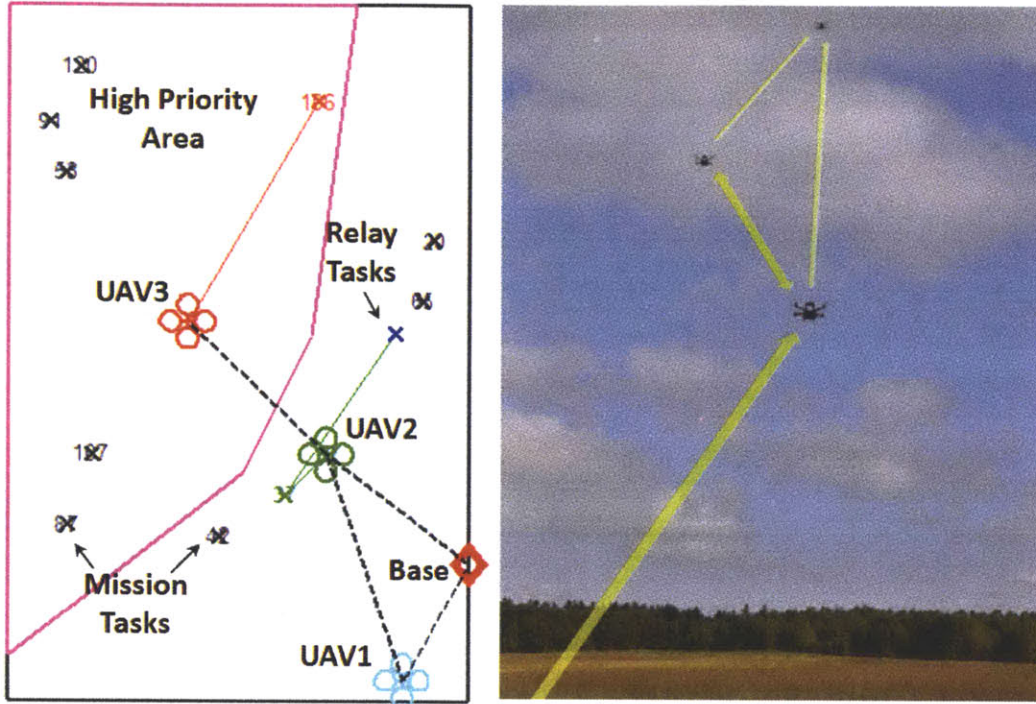


Figure 5-4: Flight testing the CBBA with Relays framework

created with uniform distribution across the flight arena. Each task consisted of a location, altitude, and time window of validity for a UAV to fly to and take an observation. Tasks inside the area of greater interest were set to yield between 7.5 and 10 times more reward than tasks outside that area, based on the distance from the base. This incentivized execution of these tasks even though they required a relay to stay connected.

Three 10 minute missions were executed each using different planning strategies: (1) baseline CBBA with no consideration for network connectivity, (2) the conservative CBBA with Network Prediction which drops tasks leading to disconnects, and (3) CBBA with Relays which cooperatively plans to maintain connectivity using relays. In each set of flights safety pilots performed a manual takeoff and climb to altitude. Then the UAVs were handed off to the autonomous distributed planning framework and sent on task executions. Finally the UAVs were recovered by the safety pilots after the mission for landing.

5.3.2 Results Flight Experiment 1

Each mission was first performed in software simulation, and then executed in outdoor flight test. Figure 5-5 shows the results of the experiment. The mission score plot is a profile of the cumulative reward as the mission progresses. The reward varies as tasks are executed and fuel is consumed traveling to the task locations. Stronger connectivity is achieved using the two strategies which account for this communication constraint. CBBA with Relays has the highest performance when comparing mission scores since agents cooperate to accomplish valuable tasks in the search area and relay the information to the GCS. The baseline CBBA strategy has the lowest performance since agents consume energy to travel to tasks for which they receive no reward.

The plots also indicate that the simulation results follow the same trends as the hardware results. This demonstrates that the algorithm is capable of operating during the mission execution and can overcome elements of communication drop outs and erroneous state estimates from the vehicles flying outdoors. Two effects were observed in the field which had not been captured in simulation. First, each replan in this implementation took significant time (up to 20 seconds in some instances). In simulation however, time stops during replans and therefore the vehicle positions remain the same as the plan converges. During flight tests, vehicles continued to navigate during the planning phase, and could therefore reach a configuration introducing flaws into the plan by the time it converged. Second, the actual velocities of the vehicles differed from what was predicted in planning which would result in arrival times different than predicted in planning. In one such instance, a UAV actually reached a task ahead of the predicted time and therefore started it early. Once finished, it moved on early to the next task, allowing another task dependent on it at the predicted time to disconnect with the CBBA with Relays algorithm. The lesson learned from this event resulted in a fix in the algorithm which forces agents to wait until the predicted time to start their tasks in order to maintain the planning schedule. Overall six successful sorties involving three quadrotors were executed to successfully integrate and collect data in this experiment.

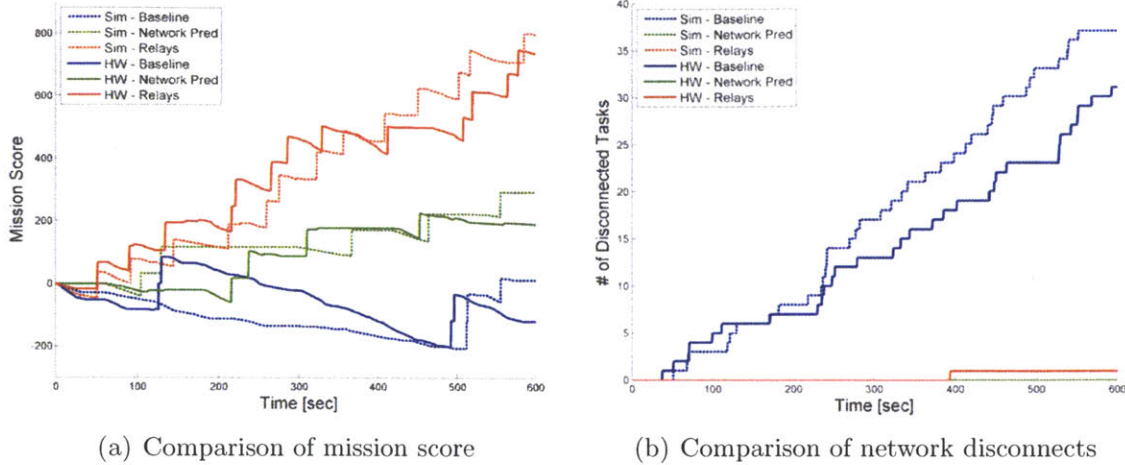


Figure 5-5: Flight test results of CBBA with Relays algorithm

5.4 Experiment 2 - Updated CBBA with Relays

The second series of outdoor experiments implemented the updated CBBA with Relays framework in flight test. More information about the algorithms in this section is provided in Chapter 4.

5.4.1 Test Scenario

The second test was executed in an arena of the same size, and also consisted of an information gathering mission with three quadrotors. This test did not designate an area of interest, and instead tasks linearly increased in value based on distance, from 0 at the base station, up to 100 at the furthest distance. The same R_{COMM} was used to define the information data-rate required to support tasks f_{tasks} as well as threshold bit-error-rate $\text{BER}_{\text{thresh}}$, similar to the Monte-Carlo simulations setup in Sect. 4.6. Again, a $5m$ buffer on R_{COMM} was accounted for in link capacity, BER, and routing predictions in planning for robustness to inaccuracies in the outdoor environment. Following lessons learned from the first flight test, the fix was implemented to force agents to wait until predicted time to start a task, and the MATLAB code was optimized to generate plans in 1 to 4s.

Four 10 minute missions where executed. The first three, in Experiment 2.1, com-

pared different planning strategies considering different communication requirements: (1) connectivity only as done in the original CBBA with Relays algorithm, (2) data-rate with the modeled AODV information routing protocol, and (3) both inter-agent BER and data-rate with AODV simultaneously. During these three missions, the communication environment was deterministic and did not include stochastic channel fading. The Linear Program optimal routing protocol proposed in Sect. 4.3.2 to evaluate performance upper-bounds was slightly too slow to safely operate in real-time flight test. For the fourth flight test, in Experiment 2.2, stochastic fading was introduced. Agents were unaware of communication parameters and had to perform real time estimation during the mission to plan effectively. For this test alone, R_{COMM} (which does not include fading) was set to 50% of the flight arena, and agents planned at an 84% level of conservatism.

5.4.2 Results Flight Experiment 2.1

Results in Figs. 5-6 and 5-7 compare the planning strategies for the first three flights. All three strategies are also compared to their respective performance predicted in simulation. Results first show that the two strategies planning to meet data-rate requirements outperform the original CBBA with Relays algorithm which results in frequent link capacity overloads and missed reward. In addition, the green line in Fig. 5-7(b) indicates that the framework successfully maintains interconnectivity between nodes through relay placement and using motion control, as opposed to the two other strategies.

Results not plotted here also indicate that all three planning strategies achieved 100% connectivity throughout mission execution as a result of lessons learned from the first flight test. Interestingly though, results in Fig. 5-7(a) reveal three instances of overloaded capacity using the two strategies which specifically plan for this requirement (each step up in the graph represents a separate event). These were not predicted in simulation, and were the result of the opposite phenomenon from the first flight test. In all three instances, an agent traveled slower than predicted due to head wind and arrived at its task late. In each case the task was still in its window of

validity and agents started late anyways. Shortly after, other agents start other tasks which had not considered the late task still being active during network prediction in planning. This led to the three unpredicted overloads which were directly responsible for reducing the mission score in data-rate and joint BER/data-rate planning strategies compared to their simulated runs. Different solutions are proposed to fix this issue, but need further evaluation: (1) the planner could use more conservative velocities in planning, (2) network predictions could include time window buffers to account for possible task delays, and (3) agents could be forced to stop executing or not execute a task and stay put in the event of a late arrival. Overall though, the results show the algorithm generally works well and as intended, and improves the performance of the multi-UAV team subject to more realistic communication requirements.

Finally, the results in Fig. 5-6 show that the mission considering connectivity only (blue) actually outperformed in mission score the value predicted in simulation, despite running the same scenario. The difference stems again from model inaccuracies in the simulator compared to real-world dynamics. Because quadrotors were actually in different locations in flight than in simulation at any given time, agents proposed a slightly different set of bids in the task allocation, which sent them on different paths, and then changed the outcome of the mission (here for the better). This behavior calls for repeated trials for such experiments, which unfortunately was not possible here due to limited resources.

5.4.3 Results Flight Experiment 2.2

Results in Figs. 5-8 and 5-9 show the system performance during the mission and compares it to predictions in simulation. In Fig. 5-8(a) there is a noticeable mission score performance difference between simulation and flight test. The reason for the difference can be attributed again to differences in predicted versus actual arrival times at the tasks. In this experiment agents experienced significant delays due to wind, and arrived late to six of the tasks during the mission. On two occasions, the delays were enough to make the task invalid and therefore force assigned agents to

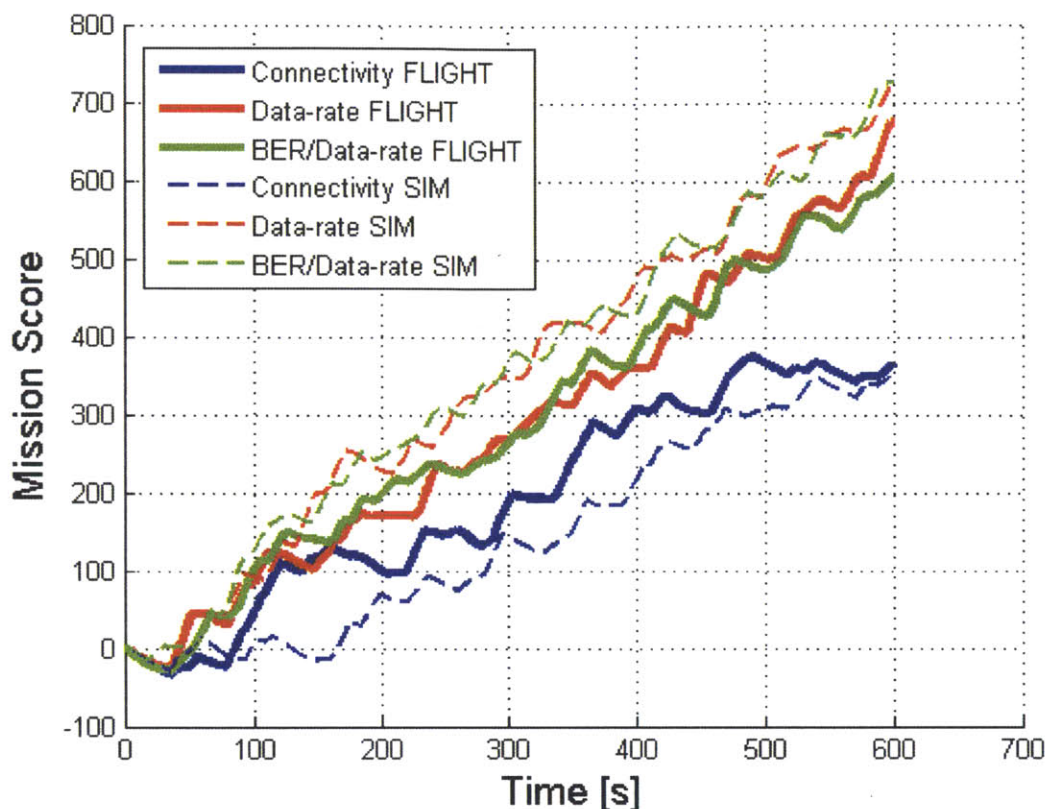
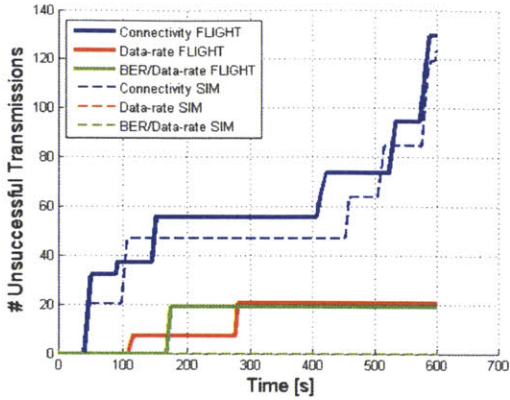


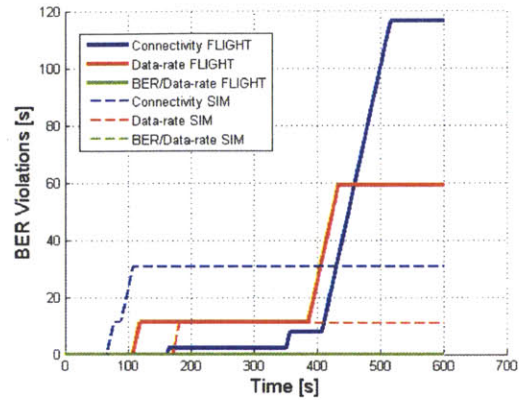
Figure 5-6: Outdoor flight test mission performance comparison using three different planning strategies

skip on to the next item on their list. As such, UAVs incurred cost for a plan they could not realize in time. In fact in one of these instances, a second UAV served as relay for the delayed and skipped task which incurred travel costs for two UAVs without the expected high reward. Clearly this indicates the planning velocity of $4m/s$ was too high for these conditions and should be reduced. It also reveals small changes in weather conditions, even over the course of one day, can have a significant performance impact if the system is operating near its capability limits.

On the other hand Figs. 5-8(b) and Fig. 5-9 indicate that the UAVs were successfully able to estimate the communication parameters throughout the mission and adapt plans accordingly. Here the number of network overloads follows the same trends as in simulation, and the estimates adjust rapidly to the sudden 10% parameter degradation in the environment at 320s in the mission. The time to detect 90% of the change ($t_{90\%}$) and mean tracking error (\bar{e}) of new values σ_{dB} and γ_{dB} (per-



(a) Comparison of data-rate violations



(b) Comparison of BER violations

Figure 5-7: Outdoor flight network control performance comparison using three different planning strategies

Table 5.1: Communication Estimation Performance in Flight Test

Parameter	γ_{dB} $t_{90\%}$	γ_{dB} \bar{e}	σ_{dB} $t_{90\%}$	σ_{dB} \bar{e}
Simulation	61s	4.8%	11.8s	0.8%
Flight Test ($t < 550s$)	52.3s	6.5%	19.1s	1.0%
Flight Test ($t \geq 550s$)	N/A	49.0%	N/A	8.1%

formance predicted for 100m estimated on a 1s cycle) are listed in Table 5.1. A significant erroneous deviation in estimates can be seen in Fig. 5-9 during flight test at 550s. Here 2 UAVs became positioned very close to one another due to randomly generated and assigned tasks. This small distance created a singularity in the SNR samples generated by the small team which led to the degraded estimate. In future implementations these values can be identified and filtered out prior to the estimation. Despite these challenges, the results from Experiment 2.2 show the team was successfully able to estimate and plan adaptively in real-time flight test, and therefore shows initial proof of concept for this method. Experiment 2 overall consisted of six total missions with three UAVs, two for implementation, and four for execution.

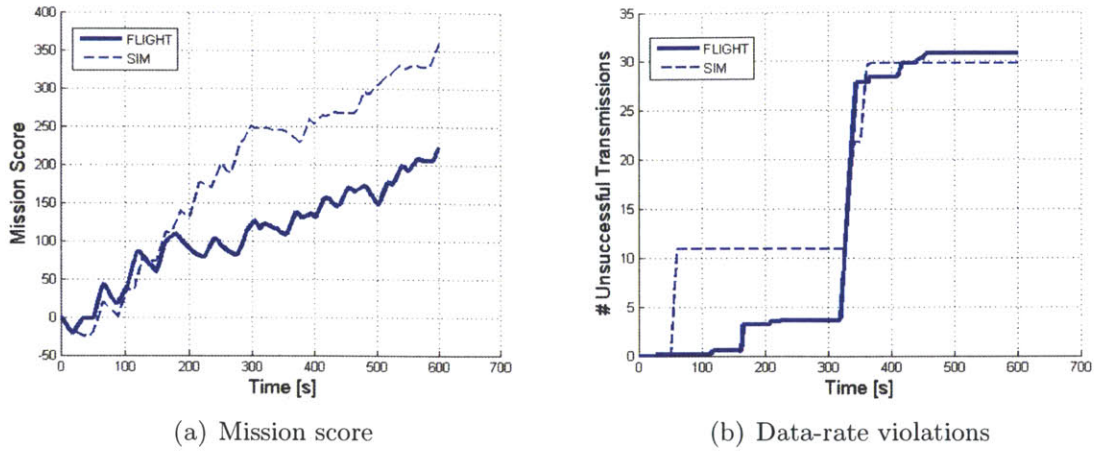
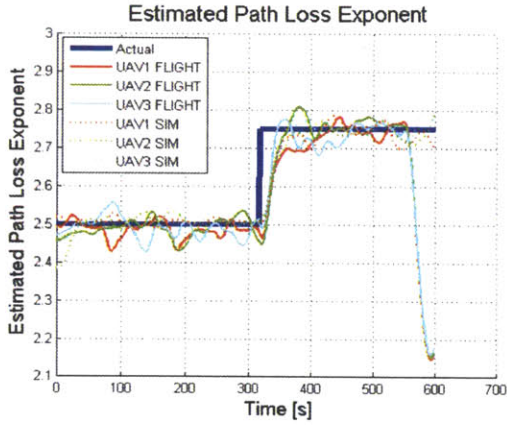


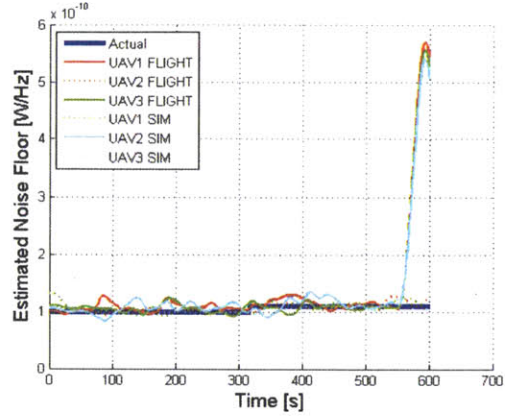
Figure 5-8: Uncertainty and adaptive planning in flight test

5.5 Summary

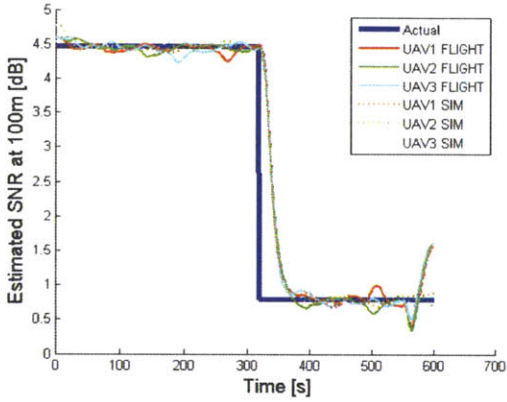
This chapter presented outdoor flight tests to perform real-time dynamic task allocation while planning to control network communications for a team of UAVs. The objectives of the flight test were to show concept feasibility through (1) real-time operation and (2) robustness to uncertainties and imperfections of an uncontrolled outdoor environment. The CBBA with Relays framework was adapted to interface with three Pelican quadrotors as well as a flight test safety monitoring interface. In a first set of experiments the original CBBA with Relays algorithm from Sect. 3.3 was flight tested and showed similar performance trends to those predicted in simulation. It also validated in flight that cooperative planning with relays outperformed other non-cooperative strategies. A network disconnect was observed however using the algorithm, and was due to agents traveling faster in flight than predicted. This lesson learned was incorporated as a fix in the improved CBBA with Relays framework and tested in a second set of flight experiments. With it, agents outperformed the original algorithm by planning considering realistic network dynamics to meet data-rate and BER interconnectivity requirements. The ability to perform real-time estimation of a simulated communication environment and use that information for adaptive task allocation planning was also successfully validated in flight.



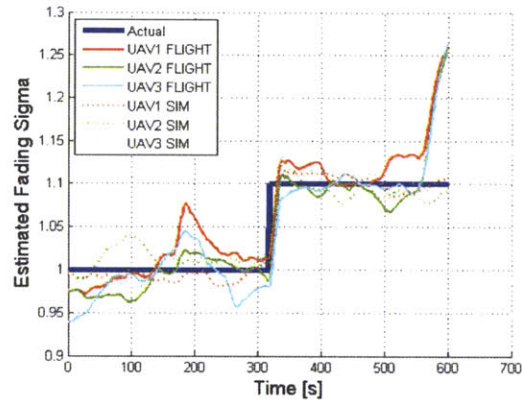
(a) Path loss α



(b) Noise floor N_0



(c) Signal to Noise Ratio (γ_{dB})



(d) Fading variance σ

Figure 5-9: Estimation of simulated communication environment in flight test

During the second set of tests the opposite phenomenon from the first set was observed and agents this time traveled slower than predicted. This led to three un-predicted instances of network capacity overloads, two tasks being canceled due to delays, and resulted in lower mission performance than predicted in simulation. Overall results from these flight tests and the discrepancies identified, motivate the need for further field testing to validate planning algorithms. In addition, they highlight the importance of including robust planning strategies [107] to understand and mitigate the risk of unachievable plans. The next validation step required for this work is to incorporate actual networking modules onto the vehicles and execute planning based on their measurements.

Chapter 6

Decentralized Multi-UAV Control Under Human Supervision

6.1 Motivation and Overview

This thesis chapter includes work which was presented as a co-authored research paper [54, 120, 121]. Andrew Clare, Olivier Toupet, Jonathan P. How, and Mary L. Cummings are acknowledged for their contributions to this paper as well as the technical work described in this chapter.

Here focus shifts to provide the results of a multi-UAV flight test campaign with a control framework and mission scenario relevant to the work in previous chapters. Even though effective network communications control is paramount for a multi-vehicle system to operate properly, a number of other challenges must also be addressed in order to field such a system. These include dynamic resource allocation of a possibly heterogeneous UAV team as discussed the Chapter 3. Next, path planning is needed not only to route vehicles to their assigned tasks, but also maintain safe separation from ground obstacles and other moving vehicles. Effective human supervision is critical to make high-level execution decisions to ensure the mission is carried out as intended. Finally, this concert of elements must be robust to operational uncertainties which include hardware failures, degraded communications, and

imperfect sensor measurements, in order to ensure objectives are properly and safely met.

This chapter presents the outdoor flight test implementation and results of a multi-UAV system supervised by a human operator and designed to address the above considerations. The system was tested in an operationally relevant scenario based on recent military interest in implementing Cargo UAVs in theater [122]. The mission consisted of the team surveying an area, identifying, tracking, and neutralizing hostile ground targets, then safely routing a Cargo UAV to and from its designated resupply point. Three quadrotors and one fixed-wing UAV were flown during the test. Real-time task allocation was performed in a decentralized framework using the Consensus Based Bundle Algorithm (CBBA) (see Sect. 3.2). Decentralized path planning was then conducted through receding horizon planning and a hierarchical approach for collision avoidance. A multi-vehicle control human interface was designed to allow an operator to control the mission execution by designating tasks and approving plans generated by the autonomous planner. Tools were incorporated in the overall framework to enable the system to overcome failures and operational degradations, and ensure the mission could still be carried out with degraded capability. A total of 14 outdoor multi-UAV missions were conducted using this system, and demonstrated that the framework overall performs well in its intended implementation. The results and lessons learned from adapting this multi-year laboratory research effort in an operational field experiment are relevant to the overall cooperative control research community and further support the need for effect network communication control in multi-UAV systems.

6.2 Background Information

6.2.1 Related Research

To enable a human operator to control multiple UAVs, a significant level of team autonomy is required to alleviate the workload of flying and navigating an aircraft.

Numerous methods have been explored to allow autonomous multi-unmanned system collaboration. Section 3.1 presented different techniques developed to allocate different mission tasks to a team of agents in either centralized or decentralized control. Many of these studies also included methods of path planning, which is often coupled with the task assignment. While these frameworks and many others in literature provide various methods for UAVs to cooperate, most of these approaches rely on an autonomous scheduler with little to no human input for the task assignment.

The user community however recognizes the importance of maintaining human control in operating unmanned systems [123, 124]. As such, the study of human supervisory control has also received significant attention in recent years. For instance, Ref. [125] explores how changing the level of system autonomy affects the performance of a human operator controlling multiple UAVs. Furthermore, while it is recognized that these operations require the computational capability of optimization algorithms to coordinate unmanned system teams, a number of studies have shown that humans collaborating with algorithms can achieve higher performance than the algorithms alone [126, 127]. Lessons learned from previous work was therefore used to design the human-computer interface to most effectively balance the roles of the human operator and the automation [128].

Several outdoor flight tests have been previously executed with fully autonomous algorithms to coordinate UAVs. These are listed in Sect. 5.1 and include the experiments in the previous chapter. Others have designed systems where the human operator can guide the team of heterogeneous unmanned vehicles with the assistance of a centralized planning system [129, 130] and a distributed planner [118]. This work presents novel contributions in three areas. First, the system demonstrates the ability to conduct an outdoor cargo resupply mission, including surveillance of the landing site and autonomously selecting a clear path to the target considering potentially hostile targets that were detected on the ground. Second, the system in this work utilizes a decentralized control architecture that can conduct dynamic re-tasking in real-time in order to plan for a variety of tasks including surveillance, target tracking, simulated target engagement, and cargo resupply. Third, the system enables

the human operator supervising the team of vehicles to guide the system at a high level, without requiring manual vehicle control but while maintaining the operator's situational awareness of the mission. The operator collaborates with the planning algorithm to generate effective schedules for the vehicles, monitors the health of the vehicles, creates and prioritizes tasks, and identifies objects in imagery captured by the vehicles.

6.2.2 Cargo UAV Application Scenario

The Department of Defense has recently been involved in rapid fielding efforts to implement Cargo UAV technology in theater. The intent is to provide "time-sensitive" logistical support to troops in forward locations and mitigate the need for ground convoys which are subject to the hazards of ambushes and improvised explosive devices [122]. The DoD announced the first operational use of a Cargo UAV to resupply a forward combat outpost in Afghanistan in January 2012 [131]. In order to provide operational context to the research, the scenario created for this experiment consisted of supporting a Cargo UAV resupply mission using the multi-UAV framework.

The objective in the scenario is to provide aerial reconnaissance, enemy suppression, and safe routing support for the Cargo UAV (CUAV). The multi-UAV team consists of four heterogeneous agents. First, one Weaponized UAV (WUAV) serves as a surrogate for a Medium Altitude Long Endurance UAV with ISR and strike capabilities such as the MQ-9 Reaper. Second, two Sensor UAVs (SUAVs) are surrogate small tactical UAVs which provide low altitude high resolution imagery for target classification and precision targeting. Finally the fourth agent is the CUAV which has no defensive capability and is only used to resupply designated areas. To accomplish the mission, the WUAV must first scout the area around a resupply point to locate possible ground targets (scout task). Search tasks are then created to send SUAVs to image located targets for classification. Hostile targets must be suppressed and require the WUAV to collaborate with a SUAV to obtain coincident precision targeting data during a simulated strike (destroy task). A battle damage assessment task is then carried out by a SUAV to confirm the kill. Targets with unknown intent

need to be tracked by SUAVs, and friendly targets can be dismissed. Once the area is safe, the CUAV needs to be routed on the shortest safe path away from potentially hostile targets to complete its resupply.

This task based scenario is very similar to the one originally introduced in Sect. 1.2. The difference however are: (1) the tasks here are more specific in type and are only compatible with certain UAVs, (2) the tasks are not encoded as time sensitive and remain valid unless the human operator cancels them, and (3) the scenario assumes perfect communications and therefore does not create relay tasks.

6.3 Flight Test Architecture

6.3.1 High Level View

A key objective of the framework is to adequately balance the roles of the human operator and the automation controlling the multi-UAV fleet in order to achieve the best mission performance. In general, computer automated planners are capable of rapidly handling complex computations to optimize task allocation far better than humans. However, the algorithms they rely on are unable to process variables which were not identified in the design process unlike a human [132]. Furthermore, certain functions of the system, such as launching a weapon, may only be authorized by a human operator due to socio-technical reasons.

The multi-UAV human-supervised system consists of multiple integrated modules, each responsible for a certain aspect of the system function as shown in Fig. 6-1. The human operator supervises the system from the Ground Control Station (GCS) through the Human Interface (HI). Here the operator monitors the progress of the UAV assets, defines and prioritizes missions tasks (search, track, etc...), and classifies targets based on UAV sensor observations. A centralized task planner is embedded in the HI to propose different optimized plans to the human for review. The operator can either modify inputs and request a new plan, or approve the proposed one, which distributes the approved tasks to the UAVs. Agents then perform a decentralized

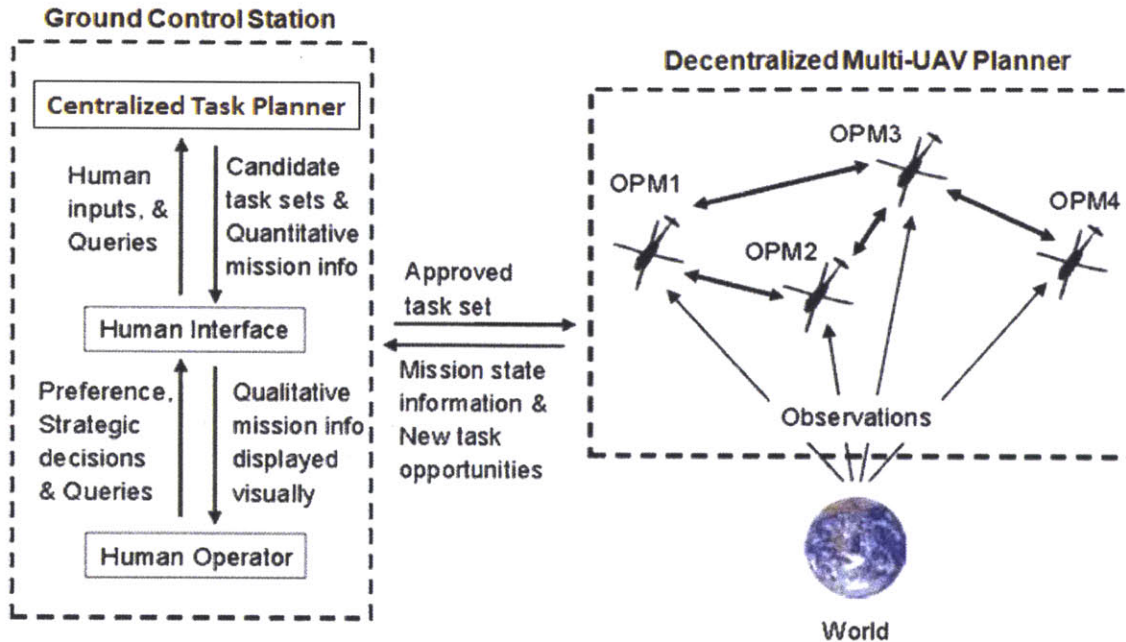


Figure 6-1: High-level architecture of decentralized multi-UAV system with human supervision

task allocation routine to optimize real-time execution of the approved plan in the dynamically changing environment. The decentralized task planning architecture retains the benefits for a multi-agent system discussed in Sect. 3.1. Once they reach consensus on a plan, each UAV plans its trajectory to accomplish the tasks and avoid obstacles and other UAVs. The trajectory is formulated as a set of waypoints, which are communicated to the vehicle autopilot for low-level control. Observations and data generated by the decentralized UAVs are sent to the GCS and displayed in the HI. This framework was developed jointly between the MIT Aerospace Controls Lab, the MIT Human and Automation Lab, and Aurora Flight Sciences.

6.3.2 Planning

The Onboard Planning Module (OPM) for each UAV (see Fig. 6-1) is responsible for both task allocation and path planning. Decentralized task allocation planning is performed using the Consensus Based Bundle Algorithm (CBBA) previously discussed in Sect. 3.2. The scores for the tasks used in the algorithm are based on human op-

erator inputs of task prioritization (“high”, “medium”, or “low”). In addition, since the routine is executed in a fleet of heterogeneous UAVs with different capabilities, the different types of tasks in the mission scenario (see Sect. 6.2.2) are appropriately encoded so that only compatible UAVs may place bids on them. Once the algorithm converges on a set of assignments, vehicles turn to their decentralized path planners to compute their flight paths.

Two different algorithms enable the UAVs to autonomously plan their routes in the environment. One algorithm is used to find the path that maximizes the probability of discovering a target when searching a given area (used for search and scout tasks). This algorithm is based on a decomposition of the environment into cells and a breadth-first tree search technique [92] over a limited, receding planning horizon, as illustrated in Fig. 6-2. The cells are used to represent, in finite dimension, the probability distribution of the target’s estimated location and the obstacles in the environment (both static, e.g. trees, and dynamic, e.g. other UAVs). The probability distribution and location of the moving obstacles are updated periodically as new information is collected during the mission, prompting a dynamic replan of the vehicle trajectories. In order for the vehicles to coordinate their paths and ensure collision avoidance in a tractable manner, a simple hierarchy scheme was implemented: each UAV plans its path around the trajectories of the UAVs of higher priority.

The other algorithm is used to find the shortest path from the current location of the vehicle to its desired destination and is used to bring the UAVs to their recharge stations, resupply location, targets to be tracked, etc, in minimal time. This algorithm relies on the same decomposition of the environment and the well-known Dijkstra’s algorithm [92]. The same hierarchical approach is used to ensure collision avoidance among the UAVs.

6.3.3 Human Supervisory Control

Operator overload is mitigated by controlling the multi-UAV team at a goal-based level instead of at the level of individual vehicles. While the autonomous planners provide rapid solutions through complex computations, human management of the

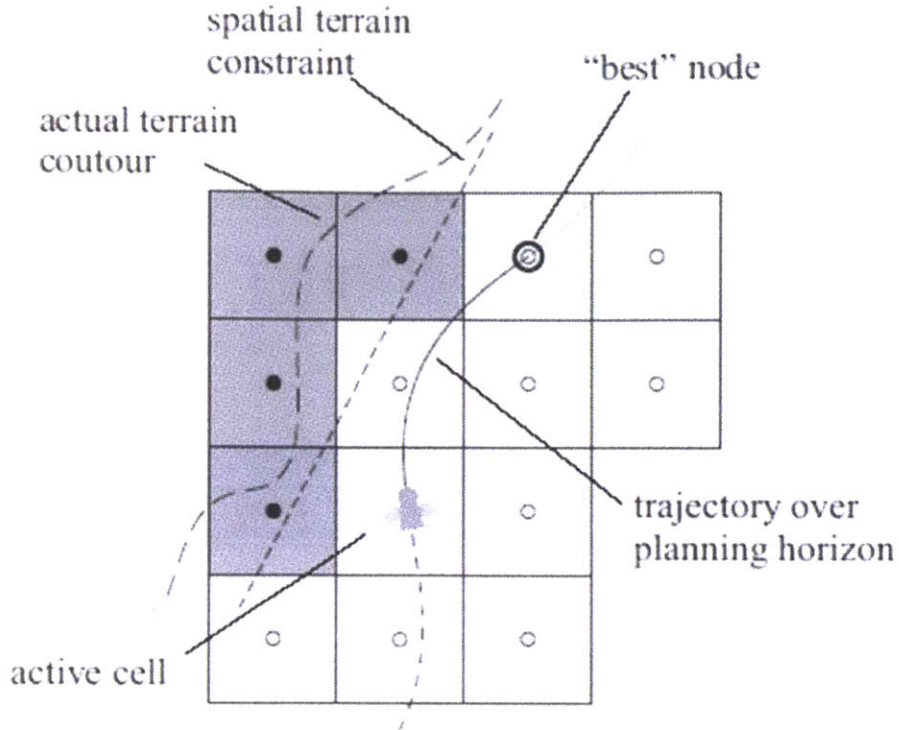


Figure 6-2: Receding-horizon path-planning based on a grid decomposition of the environment.

UAVs is paramount as autonomous algorithms do not always generate accurate solutions in the presence of unknown variables [132]. This is particularly important for a field implementation, as will be discussed in the lessons learned section, since the UAVs are subject to increased levels of uncertainties and system failures. Furthermore it is typically infeasible to obtain an “optimal” plan in a rapidly changing environment through a hard-coded objective function, in which case a human can help by guiding the automation to improve performance [128].

The HI in this framework enables the operator to supervise the progress of the mission, maintain situational awareness, and interact with the planning algorithm to improve overall mission performance [133]. The interface features a moving map showing the location of the vehicles, detected targets, tasks, and the environment (Fig. 6-3 (a)). The operator guides the planning process by creating and prioritizing tasks directly on the map. During execution, pop-ups prompt operator input for target classification, approval to launch a weapon, and battle damage assessment

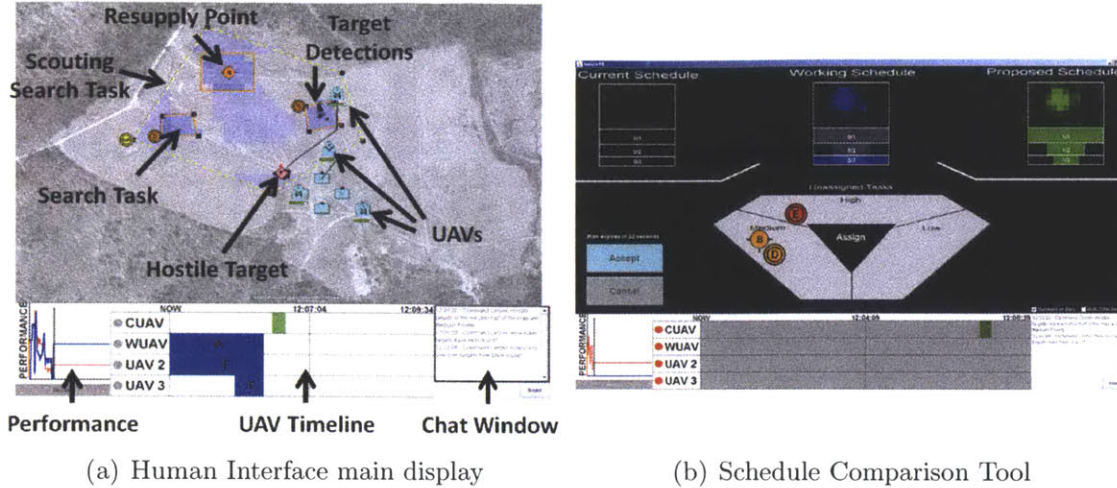


Figure 6-3: Human Interface

confirmation, which further guides the autonomous planner. The Schedule Comparison Tool (SCT) (Fig. 6-3 (b)) embedded in the HI runs a centralized version of the CBBA algorithm to allow the operator to rapidly assess the performance of different plans based on different task priorities. The plan approved in the SCT does not allocate associated tasks to the UAVs, but instead distributes those tasks to the UAV team to execute decentralized planning. In other words the centralized planner used by the operator predicts what the task assignment will be, but the actual decentralized assignment for those tasks may be different. This element of the design is key since the operator is focused on supervising the mission at a high level, and is less concerned with lower-level actions such as task hand-offs, refueling, etc. The decentralized planner enables the UAVs to rapidly adapt to the dynamic environment using local situational awareness to optimize execution of the operator’s intent.

6.4 Flight Test Implementation

This section describes the field test implementation of the multi-UAV architecture. Flights were conducted at the same site as tests described in Chapter 5 (see Fig. 6-4). The area of operation was limited to $400m \times 400m \times 50m$.



Figure 6-4: Team of four UAVs

6.4.1 Systems

Figure 6-5 illustrates the architecture implemented for flight testing. The heart of operations was the Ground Control Station (GCS) located inside the MIT command center van. The GCS housed every component of the system except for the UAVs and targets used in the scenario. Small research grade UAVs were used for this test. The twin tilt-rotor, fixed wing, Aurora Flight Sciences Skate UAV was used as an unarmed surrogate platform for the Weaponized UAV in the mission scenario (top left of Fig. 6-5). The Skate takes off vertically, weighs 2.5 lbs, is electrically powered, and has a 60 minute endurance. It is equipped with a Digi-Mesh XBee Pro digital wireless module for command and control. The same Pelican quadrotors introduced in Sect. 5.2 were used as platforms for the Cargo UAV and Sensor UAVs in the mission scenario. Each Pelican was slightly modified to be equipped with a downward looking electro-optical 640x480 NTSC video camera and a 5.8 GHz analog transmitter for live full-motion-video (see Fig. 5-3 (a)). Due to the small size of the UAV's, "onboard" computing for task allocation and trajectory planning was actually performed on the ground. Each UAV was assigned a dedicated computer located in the GCS to perform onboard functions. These separate computers maintained the intended distributed framework

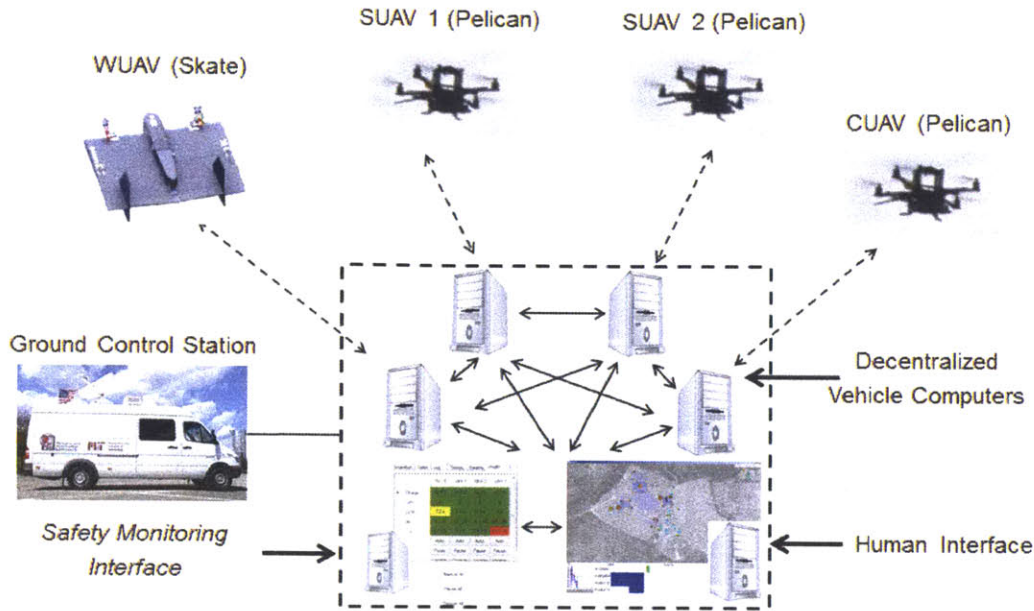


Figure 6-5: Flight test system architecture

and are a notable difference in implementation than the more simple single computer framework described in Sect. 5.2. Waypoints computed as part of the trajectory were then communicated to the vehicles autopilots while in flight.

The Safety Monitoring Interface (SMI), also discussed in Sect. 5.2, was implemented into the framework. Messages were passed between the HI, SCT, SMI, and UAV planning modules using the same LCM protocol.

The targets during the test consisted of Pioneer 3 Unmanned Ground Vehicles (UGV), a manned car, and fixed position objects. Targets were equipped with a large colored platform for the operator to perform visual recognition and classification using the UAV video sensor (Fig. 6-6). The system assumes targets are autonomously detected by the UAVs if they enter the sensor field-of-view (FOV). Since that capability was beyond the scope of this research, it was simulated by equipping the targets with GPS and XBee modules to communicate their position to the GCS (locations were not shown on the HI). A sensor module computed the sensor footprints by projecting the FOV using aircraft attitude obtained from telemetry. It then identified instances where sensor footprints overlapped a target location to trigger a detection.



(a) SUAV tracking target

(b) Target classification in HI

Figure 6-6: Targets used for during field test

6.4.2 Health Monitoring and Management

As discussed in the previous chapter, outdoor flight testing adds many elements of uncertainty and complexity in operation which may not be recognized in a controlled laboratory environment. Outdoors, UAVs travel faster and farther, and are influenced by wind, gusts, and degraded wireless communications which the autonomy was not programmed to consider. State estimations provided by GPS and IMU measurements are less reliable and can be delayed or reported less frequently, which affects the task and trajectory planners. The system is overall more prone to failure. This uncertainty and its impact on the planning system is, of course, one of the main reasons for testing outdoors. Thus the design philosophy adopted was to develop tools to enable the system to maintain mission effectiveness despite degraded capability.

A high level health monitoring system was embedded in the HI to alert the operator of UAV health issues, such as degraded communications or GPS tracking. Additional feedback was provided by observing the vehicle behaviors on the map. On occasion, vehicles failed to track waypoints or certain conditions caused the planning algorithms to rapidly alternate (or “churn”) between plans, as will be discussed later.

If a UAV was found to be degraded or its behavior was affecting mission performance, the HI operator could issue a “pause” or “disable” command. The pause command holds the UAV at its current position and temporarily removes it from the task allocation routine so that other agents can accomplish its tasks. This could then be reversed if the degraded health improved. The disable command performs a similar function, but instead permanently removes a clearly inoperable UAV so that it can be replaced by a spare during the mission.

The Safety Monitoring Interface (Fig. 5-3) was implemented separately from the HI to avoid overwhelming the operator with both supervising the mission and monitoring low-level health information for each vehicle. This interface allowed an additional test safety monitor to coordinate low level critical health information with the four UAV safety pilots. As will be discussed later, the health monitoring tools on both the HI and SMI played a critical role in field testing this system. While this implementation used a second human operator for the SMI, this process could also be automated [134].

6.4.3 Typical Mission Execution

Flight tests were executed according to the cargo UAV mission scenario described in Sect. 6.2.2. UAVs started on the ground and as tasks input by the HI operator got assigned to them, the SMI operator would instruct the safety pilots to manually takeoff, climb to altitude, and turn the vehicles over to the planning system. Missions began with a desired cargo UAV resupply point and the HI operator designating a scouting area around it for the WUAV to locate potential targets. As the WUAV scouted and its sensor footprint overlapped targets, detections were triggered and displayed on the map (Fig. 6-3 (a)). The operator could then create local search tasks around the detections for a SUAV to take a closer look. If the SUAV detected a target during the search, it sent an image to the operator for classification (“hostile”, “unknown”, “friendly”) and prioritization (Fig. 6-6). For hostile targets, a destroy task was created which led the WUAV and a SUAV to coordinate through the task and path planning algorithms to engage. Once the WUAV was in position to release

its weapon (simulated in this test), it sought operator approval to fire. Then a battle damage assessment task was automatically created and assigned to a SUAV to send another picture of the target for the operator to confirm the kill. Targets classified as “unknown” created persistent tracking tasks, and “friendly” targets required no further action. Once the operator determined the area to be safe for the CUAV, a resupply task was created, and the trajectory planning algorithm generated the shortest path to the objective while treating unknown and hostile targets as dynamic obstacles to avoid. If refueling was required during the mission, as was often the case with the quadrotors, the vehicle automatically flew back to its starting location where the safety pilot regained control, landed manually, swapped the battery, and returned the UAV to the mission.

6.5 Flight Test Results

A thorough sequential build-up approach was used to ramp up the multi-vehicle experiment. It consisted of a series of simulations in the lab, numerous single vehicle integration flights, and discrete event flight test scenarios of increasing complexity and number of UAVs. Once operational, a total of 14 missions were conducted with three or four UAVs being controlled collaboratively by the HI operator and the automation. This resulted in a total of 16 hours of logged small UAV flight, and 83 aircraft sorties. Data collected from modules in the framework was logged for post process analysis. This section describes some of the challenges encountered while operating outdoors, the measured system performance, and overall lessons learned from this exercise.

6.5.1 Challenges and System Robustness

A key objective of the design was to enable the system to overcome some of the failures inherent to complexities of outdoor flight testing, and to show that the system could complete the mission even with degraded capability. Different types of system degradations were indeed observed in the field. First, UAVs occasionally failed to track waypoints or altitude due to errors in the autopilot, and required a safety pilot

Table 6.1: Summary of Failures during Trials

Type of Failures	Pauses	Disables
Altitude Tracking	3	0
Waypoint Tracking	2	4
Communications	1	2
Algorithmic Churning	3	0
Hardware	1	1
Total	10	7

to temporarily take over. Second, communications were often degraded due wireless channel dynamics described in Sect. 2.1.2, and resulted in delayed or denied messaging between vehicles and the GCS. Third, algorithmic churning was observed in task allocation and trajectory planning, as will be discussed later, and led to degraded team performance. Fourth, two occasions of hardware failures were observed when quadrotor motors ceased in flight. Despite these complications, 13 of the 14 missions attempted were successfully completed which can be attributed to the implementation of health monitoring and management tools discussed previously. The one unsuccessful mission was due to a software error in the framework, which was subsequently corrected. Table 6.1 lists the types and numbers of failures which occurred during the 14 trials and the steps taken for each occurrence.

Several occasions of “churning” were witnessed during flight testing. Churning occurs when the automated planner rapidly switches back and forth between two or more plans. This can significantly degrade the team performance as agents will not commit to carrying out their plans. In flight testing, imperfect state estimate, inaccurate waypoint tracking, and non-linear behaviors contributed to churning in the planning of both task allocation and flight trajectories. For instance, the task planner initially assumed a linear discharge of vehicle batteries to predict if there was sufficient charge to take on a task or if the UAV should instead recharge. In reality, discharge was nonlinear as voltage would suddenly drop, plateau, and even rise because of vehicle maneuvers and gust stabilization. Several instances were therefore observed where agents would take on a task, then release it to another agent to go refuel,

and then get re-assigned to it and so forth (see Fig. 6-7). On one occurrence, this type of churning repeated for over three minutes where the UAVs involved did not accomplish anything. Initially the human operator had to override the autonomy by “pausing” one of the problematic vehicles. Later a fix was implemented by ignoring the battery discharge rate and detecting a set threshold voltage value below which agents would be forced to go refuel once and for all. This simple solution fixed the churning problem and resulted in higher system performance. More accurate energy consumption models may provide a more elegant solution to better consider refueling requirements in task allocation.

Churning was also observed in decentralized trajectory planning mostly due to imperfect waypoint tracking, weather elements, inaccurate state prediction, and delayed positional reporting due to degraded communications. Vehicles would sometimes plan a path around one another one way, then transition into a state where another deconflict-ed path was more optimal, and therefore plan to go around each other another way (see Fig. 6-8). This process could repeat itself several times causing delays in the vehicles reaching their destinations, frustration for the human operator, and occasionally changes in the task assignment. In general, trajectory plan churning was always resolved by the autonomy and did not interfere as much with mission execution as the task allocation churning phenomenon. Several solutions exist to resolve this type of issue, such as deconfliction limited to altitude only, slower replan rates, filtered replanning [135], and tighter integration of the path and task planning, but further work is required to validate and integrate these techniques. These events highlight the importance of network communications in the operation of a multi-UAV system.

6.5.2 System Performance

Several metrics were used to analyze the performance of the system. Missions varied from one another and the system evolved from lessons learned over the course of the 14 trials executed. For this reason, the performance is compared across three

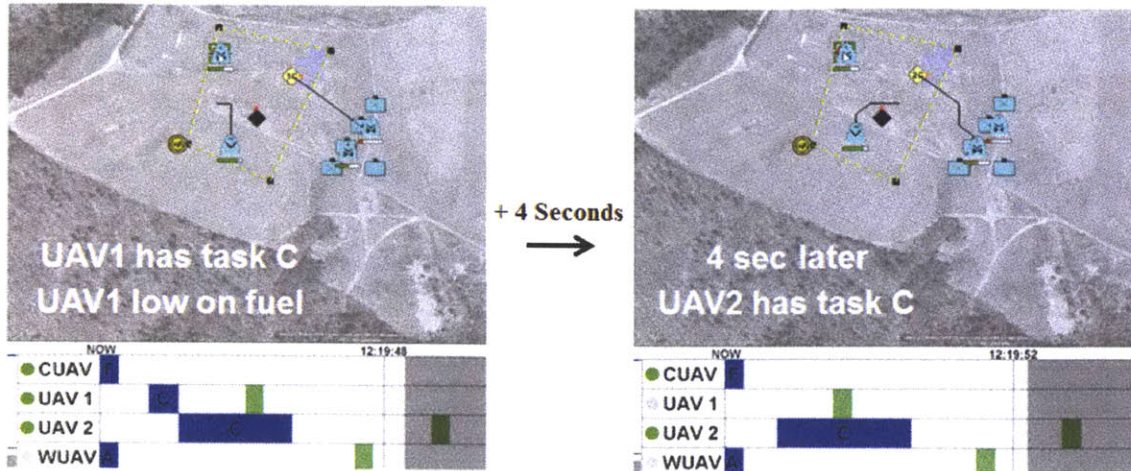


Figure 6-7: Churning in task allocation due to nonlinear voltage discharge

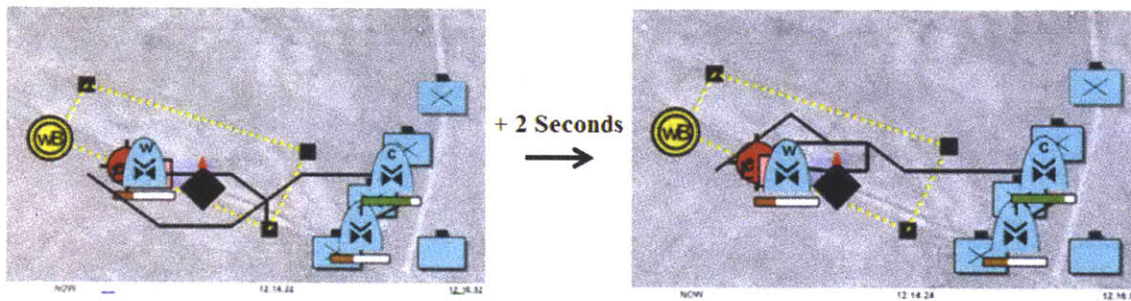
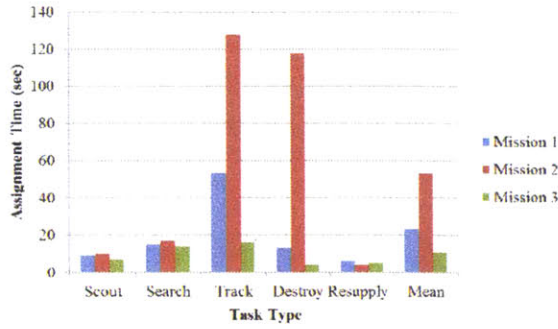


Figure 6-8: Churning in trajectory planning due to competing deconfliction paths representative missions to illustrate certain points instead of providing an average across all trials.

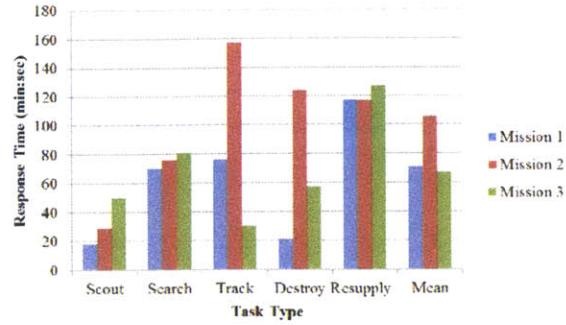
- Mission 1 was conducted early in the field campaign with three quadrotors actually flying and the WUAV operating in simulation, and therefore had less complexity than trials with four actual UAVs. During this trial one quadrotor was briefly paused due to inaccurate altitude tracking.
- Mission 2 involved all four UAVs, but suffered from several complications such as task allocation churning and three instances of waypoint tracking failures which resulted in a SUAV being disabled.
- Mission 3 also involved all four UAVs, but had the fix implemented to prevent task allocation churning, and only suffered from a quick UAV pause due to inaccurate altitude tracking.

The first metric analyzed was the task assignment time, or time to assign a task to an agent after it was created. This includes time for the operator to request and approve a plan, time for the decentralized task allocation routine to converge on a plan, and time for an agent to become available to take on the task. Task assignment time ranged from 3 to 350s with a median time of 12s. The planning algorithm in general converged within one second, so increased assignment time was mostly due to the operator not requesting (or forgetting to request) a plan, or agents being unavailable. Similarly, the task response time, or time to start a task after it has been created, was also evaluated. Task response includes time for task assignment, path planning, and actual travel. These times ranged from 15 to 411s with a median of 58s. Finally, the tracking persistence, which is the percent of time a track task was being fulfilled by a SUAV compared to the total amount of time that task was in existence, was another metric used to assess performance. Mission 1 had 65% tracking persistence, where as Mission 2 had 55% and Mission 3 was 85%. In all cases Mission 2 showed decreased performance that can be attributed to task allocation churning which created drastic operational inefficiencies, and because eventually one of the SUAVs was disabled due to failures, leaving only one remaining SUAV to take on all tasks. In Mission 2 the degraded system took nearly five times longer to assign task and 58% longer to start executing them compared to the system operating nominally in Mission 3. Track and destroy tasks were found to be the most sensitive to these issues. The difference in performance between Mission 2 and 3 is mostly due to “outdoor factors” which would not necessarily be observed in simulation or a laboratory setting.

In order to measure the cognitive workload of the Human Interface operator during the missions, a utilization metric was calculated as the ratio of the total operator “busy time” to the total mission time. For utilization, operators were considered “busy” when performing one or more of the following tasks: creating or editing tasks, identifying and designating targets, approving weapons launches, conducting battle damage assessment, interacting via the chat box, replanning in the SCT, or using the health monitoring panel. Across the 14 missions, operators had an average utilization



(a) Mean task assignment times



(b) Mean task response times

Figure 6-9: Time performance metrics compared across 3 missions

of 22.6% with a minimum of 16.8% and a maximum of 31.6%. Representative Missions 1-3 all had utilizations between 19% and 20%. Essentially, about one-fifth of the HI operator’s time was spent actively engaged with the interface, while the other four-fifths were spent monitoring the progress of the mission. Mission 2 however stood out from the other two in a few ways. First, peak utilization was determined by calculating the utilization per minute throughout the mission and taking the maximum value. Figure 6-10 shows the peak utilization during Mission 2 was 76% while the other two missions were 47% and 45% respectively. In addition, the operator chose to use the “what-if” capability in the SCT to modify the schedules generated by the planner 3 times during Mission 2, while the operator did not use the “what-if” capability in either of the other two missions if interest. These results show that the HI operator was indeed working harder during Mission 2. This may be because the operator had to deal with vehicle navigation problems and task allocation churning. It should be noted that despite this increased workload, the mission was still successfully completed and the operator’s workload rarely surpassed 70%, which is generally considered to be an acceptable, but near the upper bound limit, level of utilization for safe human supervisory control [136–139].

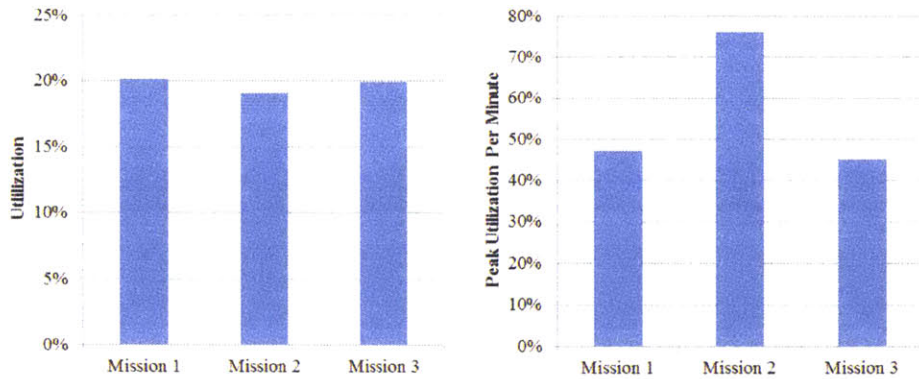


Figure 6-10: Mean and peak utilization of reference Missions 1-3

6.5.3 Lessons Learned

A key objective of this effort was to study the implications of adapting a laboratory developed decentralized multi-UAV system [110, 140] with human supervision for outdoor field operations. Some of the following lessons learned through this exercise are applicable to the overall multi-unmanned system cooperative control community.

1) The implementation generally worked well – The implementation of this system generally worked well in executing the mission. It repeatedly performed proper task allocation, planned proper trajectories, and provided adequate information to the human operator to maintain situational awareness and accomplish mission objectives. In addition, the framework was able to overcome many failures and uncertainties encountered in the field environment to allow the system to successfully accomplish the mission. The ability to “pause” or “disable” a degraded vehicle was a key enabler, since any one of the 17 failures observed which prompted these commands would have resulted in a mission abort otherwise. This also demonstrated that the CBBA algorithm was able to handle a dynamically changing number of active agents, as UAVs were removed or re-added to the team.

2) Health monitoring and management is critical but as implemented created significant operator workload – Health monitoring and management was critical to the field test success but required significant operator attention in its current implementation. Most of the failures observed were unpredictable and uncontrollable, but were able to be diagnosed and handled in real-time to keep the system moving

forward. The workload created by monitoring the health and status of the UAVs was so high that it needed to be taken on by a second human, the safety monitor, to allow the primary human operator to supervise the actual mission. Several items can be mitigated in future work by using UAVs capable of more robust altitude and waypoint tracking. Health monitoring, however, is a very significant problem for single operator control of multiple UAVs, and even with highly automated vehicles may drive workload to unacceptably high levels. More highly automated health monitoring, error detection, and self-repairs are necessary before a single operator can feasibly control multiple UAVs in such a complex environment.

3) Inaccurate state estimates, modeled dynamics, and degraded network communications resulted in “churning”

– Inaccurate and unpredictable UAV states revealed churning behaviors in task allocation and path planning. These behaviors had not been observed in previous simulations [120] or controlled indoor laboratory tests [141]. Task allocation churning was the result of an over simplified battery discharge model, and led to significant performance degradations in mission execution. Path planning churning occurred because of inaccurate waypoint tracking, the environment, inaccurate UAV state estimates, and delayed feedback due to network communication degradations. While these churning issues have been investigated previously [135, 142], it is clear that further research is required. Even though the factors causing these issues could have been simulated, identifying such factors and reproducing them with sufficient accuracy is difficult in simulations. This alone provides strong justification to perform further outdoor flight testing.

4) Human judgment was necessary and beneficial to operate the system

– Despite the tremendous recent advances in autonomy, human judgment is still necessary and beneficial to operate the system. The human operator was essential for dealing with the “fog of war”. Target position reports from the high altitude vehicle were often inaccurate, camera imagery from lower altitude Sensor UAVs were sometimes noisy or did not reveal the target, and reported vehicle positions were sometimes noticeably erroneous. The operator merged all of the information from these sources

to make decisions about mission executions and to help guide the autonomy. The human operator was active in dealing with a number of real-world subjective factors that are difficult for the algorithm to take into account. Human judgment played a critical role in assessing how “well” a vehicle was behaving and determining if it needed to be paused or disabled in the system. In addition, the human made the determination when it was “safe enough” to send the Cargo UAV to resupply based on situational awareness of the missing area.

5) The autonomy and human worked mostly well together – The system framework appeared to effectively balance roles, where the operator would create and prioritize tasks, and the autonomy would then compute the best way to execute them. However, on occasion the algorithm chose a task assignment that seemed counter-intuitive. The operator would then often times attempt to modify the plan using the Schedule Comparison Tool (SCT) to achieve a more intuitive plan. Better feedback from the algorithm could help the operator understand why the planner made the assignment a certain way. In addition, the system was implemented in this experiment to plan conservative schedules consisting of only one task in the planning horizon. This was done because some of the tasks such as target tracking and scouting had uncertain time lengths associated with them. This however required the operator to request a new plan after each UAV completed each task, which increased the workload. It also created some frustration when a vehicle would complete a very short task and then go idle unless the operator requested a new plan. While it is possible that this choice of implementation was the correct one for such dynamic missions with unbounded task lengths, future work should investigate methods to increase the planning horizon so that it is more consistent with operator expectations.

6.6 Summary

This thesis chapter presented the implementation, results, and lessons learned from outdoor flight testing of a decentralized multi-UAV system with human supervision. The framework was designed to balance the roles of the autonomy and human operator

to maximize mission performance. The human operator provided high-level mission intent by designating tasks, approving plans generated by the autonomy, classifying targets using UAV sensor feeds, and approving weapon launches. Autonomous planners associated with each UAV used the Consensus-Based Bundle Algorithm to perform real-time decentralized task allocation and then trajectory planning to efficiently and safely accomplish tasks. High level health monitoring and management tools were embedded in the Human Interface to allow the operator to identify and overcome issues by temporarily or permanently removing problematic UAVs from the mission. The system was implemented for outdoor flight testing with a team of four small heterogeneous UAVs conducting an operationally relevant mission supporting a Cargo UAV resupply.

Results support that the implemented framework worked as intended during flight testing, and 13 of the 14 missions attempted were successfully completed. As expected the implementation of the system outdoors was challenged by uncertainties and system degradations common in field testing, such as degraded network communications. Health management tools were a significant contributor to success of the effort, but created too much workload for the single mission human supervisor. The outdoor implementation revealed instances of algorithmic churning in task allocation and path planning due to delayed communications and unpredicted vehicle states, and caused performance deterioration until a fix was implemented. The balance of roles between the human and the automation in the design was found to be adequate, and human judgment was shown to be helpful in resolving subjective factors during the mission.

This chapter provides description of the system level challenges needed to operate multiple UAVs as a coordinated team, beyond effectively controlling network communications. Some of the challenges experienced in the field due to lossy network communications, which led to algorithmic churning and degraded vehicle operations, directly support the need for network communication control algorithms similar to those presented in Chapters 2-5. Furthermore lessons learned in the exercise justify the need to conduct field testing to validate laboratory assumptions, and identify areas of future research needed to further operational multi-vehicle collaborative control.

Chapter 7

Conclusion and Recommendations

7.1 Conclusions

As unmanned systems technology progresses, teams of multiple unmanned air vehicles will play an increasingly important role in a broader range of applications. The framework in this thesis enables such teams to cooperatively execute complex scenarios using dynamic mission planning to support communication requirements. A comprehensive survey of current communication control strategies for multi-agent networks was presented in Chapter 2. This relatively new field of study commonly investigates two types of problems: (1) constraining the motion of agents on a deployment objective to maintain interconnectivity [12–28], and (2) optimal deployment and control of agents pre-designed as communication relays to support a network [29–40]. Chapter 3 presented the CBBA with Relays algorithm, a unique method to control the network through distributed task allocation which ensures tasks undertaken are supported by the network and underutilized agents support the team as communication relays. The distributed algorithm uses information embedded in the Consensus Based Bundle Algorithm (CBBA) task assignment, such as task location and time of execution, to predict the network topology and plan support through the creation of relays. By explicitly coupling the task assignment and relay planning processes, the team is better able to optimize the use of agent resources to address current needs of a dynamic complex mission.

The CBBA with Relays framework was initially built using simplified and deterministic communication assumptions which resulted in data bottlenecks and unacceptable messaging error rates. Therefore, the development presented in Chapter 4 improved this approach to consider realistic wireless networking dynamics, including path loss, stochastic fading, and information routing in an uncertain environment. This improved framework enables the multi-UAV network to support data-rate requirements to send remotely sensed data back to a base station, and to maintain required inter-agent messaging bit-error-rates for cooperative task execution. Since the framework only plans network support for agents busy executing tasks, a decentralized reactive motion control policy was included to drive other agents to interconnected states. Finally, the system enables UAVs to perform decentralized estimation of wireless channel performance in uncertain environments, adjust their planning strategy to account for risk of network violations, and adapt their behavior to changes in the environment.

Four experiments were conducted in simulation to characterize the performance of the system. The first illustrated the need for proper balance in incentivizing relay tasks to achieve higher team performance through cooperation, where too low or too high of an incentive value resulted in lower mission scores. The second showed team performance improves when planning to meet data-rate requirements and results in more conservative task assignments than when considering connectivity alone. Planning to meet data-rate achieved 98.4% network supportability using real-world routing protocols and 100% with optimal routing, as opposed to only 38.6% with the original framework. Because the algorithm only predicts the network based on agents busy executing tasks, other errant agents can unpredictably alter the planned data routing. As such, this framework can only guarantee network support using a routing protocol which perfectly balances data-flow and avoids unplanned bottlenecks. This experiment also showed high performance in meeting interconnectivity BER requirements ($> 98.3\%$) using relay planning and reactive motion control. However, the framework cannot guarantee 100% continuous time BER support since planning takes place for discrete times only. The third experiment characterized risk mitiga-

tion strategies given stochastic channel fading, and demonstrated a balance in risk is needed, where behaviors too conservative or too careless resulted in decreased mission performance. Finally, the fourth experiment validated the ability to improve team performance by estimating uncertain communication environments and adapting the planning strategy to changes.

To show operational feasibility, this work was implemented in outdoor flight testing with a team of three small UAVs as presented in Chapter 5. Experiments were conducted to show (1) the improvement in team performance when using the cooperative CBBA with Relays algorithm over non-cooperative strategies, (2) the ability to further improve performance by considering data-rate and interconnectivity BER network requirements, and (3) validation of the communication estimation process and adaptive planning behavior. Six multi-UAV test scenarios were executed. Performance parameters followed trends predicted in simulation, which demonstrates the real-time performance of the algorithm as well as some robustness to the uncontrolled environment (e.g., inaccurate and delayed state estimates, environmental effects). Each test scenario revealed the need to better consider uncertainty in the task execution schedule. On multiple instances, vehicles arrived and started tasks either early or late, which changed the topology from its prediction and led to network violations. An algorithmic fix was implemented to eliminate some of these issues, however this warrants future work as described below. Despite these events, the results from flight testing and experiments conducted in simulation indicate that the overall approach presented (1) is operationally feasible, (2) improves performance of a multi-UAV team in a complex mission through cooperation, (3) enables network supportability by considering realistic networking dynamics and uncertainty, and (4) meets the intended objectives laid out in this thesis.

In addition to these experiments, another outdoor flight test campaign with a team of four heterogeneous UAVs performing decentralized planning under human supervision was described in Chapter 6. The flight tests demonstrated good performance of the framework in balancing the roles between the autonomous UAVs and the human operator, and demonstrated the ability to use the same baseline CBBA task

allocation algorithm in an operationally relevant scenario. In addition to validating this framework, two important lessons learned were: (1) the importance of mitigating impacts of sub-system degradations on team performance, and (2) the need to improve health management methods of the system to provide acceptable workloads for a single operator.

7.2 Future Work

Based on the experience acquired during this research, the following future work is recommended.

First, the CBBA with Relays algorithm and its modifications to consider realistic networking dynamics needs to be implemented using real versus simulated networking devices as a next step. Several other studies in this field [18, 35, 42], as well as empirical observations in flight testing, indicate that wireless networks tend to behave differently and more erratically than planned. Even though the framework developed in this thesis specifically aims to overcome these issues, validation is needed on an actual network.

Second, the framework developed and flight tested appears to be sensitive to task execution timing inaccuracies. As explained in Chapter 5, differences between planned and actual flight velocities changed the execution schedule which led to unpredicted network violations. Since this phenomenon is generally unavoidable, it needs to be captured in the planning process. Several recommendations were provided, such as planning for more conservative speeds, including a time-window buffer in network predictions, or forcing agents to stick to their plans or abandon their tasks. Clearly, this issue needs to be further investigated under the broader context of robust planning strategies as discussed in [107].

Third, vehicle health often degraded in flight testing due to degraded communications, inaccurate waypoint tracking, or even hardware failures. All flight tests in this work maintained safe operations because an additional operator was strictly dedicated to monitoring system health, and could communicate mitigation procedures

to safety pilots to take over manually. If teams of UAVs are going to achieve the utility envisioned for commercial and military applications, there is a critical need to improve health monitoring and management. Future work needs to further explore autonomous health management mechanisms and ensure those are robust to system failures and uncertainties themselves.

Fourth, many of the lessons learned in flight test were useful in highlighting flaws in the assumptions and designs of the work which would not necessarily have been identified in simulation. These experiments revealed how quickly system degradations and variations in state can impact the performance of planning algorithms, and helped modify their design to become operationally feasible. There is a general need to increase the amount of field testing conducted in this type of research to fully validate assumptions and enable these technologies to transition to the operational world.

Bibliography

- [1] U.S. Air Force Chief Scientist (AF/ST), “Technology horizons: A vision for air force science & technology during 2010-2030,” tech. rep., United States Air Force, 2010.
- [2] S. . Anderson, R. Simmons, and D. Goldberg, “Maintaining line of sight communications networks between planetary rovers,” in *International Conference on Intelligent Robots and Systems*, October 2003.
- [3] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang, “Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay,” *Journal of Field Robotics*, vol. 26(6), pp. 718–740, 2010.
- [4] D. M. Stipanović, G. Inalhan, R. Teo, and C. J. Tomlin, “Decentralized overlapping control of a formation of unmanned aerial vehicles,” *Automatica*, vol. 40, pp. 1285–1296, 2004.
- [5] H. Choi, L. Brunet, and J. P. How, “Consensus-based decentralized auctions for robust task allocation,” *IEEE Transaction on Robotics*, vol. 25, pp. 443–455, 2009.
- [6] P. R. Chandler, M. Pachter, and S. Rasmussen, “UAV Cooperative Control,” in *American Control Conference (ACC)*, June 2001.
- [7] P. Chandler and M. Pachter, “Hierarchical control for autonomous teams,” in *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2001.
- [8] C. Marshall, M. Mears, and S. Rasmussen, “2010 ice-t cooperative control flight testing,” in *AIAA Infotech@Aerospace Conference*, April 2011.

- [9] Office of the Secretary of Defense, “FY2009-2034 Unmanned Systems Integrated Roadmap,” 2009.
- [10] A. Goldsmith, *Wireless Communications*. New York, NY: Cambridge University Press, 2005.
- [11] E. Modiano, “16.36: Communication systems and networks.” Lecture Notes, Massachusetts Institute of Technology, Cambridge MA, 2012.
- [12] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, “Distributed memoryless point convergence algorithm for mobile robots with limited visibility,” *IEEE Transactions on Robotics and Automation*, vol. 15(5), pp. 2418–2428, 1999.
- [13] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, “Graph theoretic connectivity control of mobile robot networks,” *IEEE Journal*, vol. 99(9), pp. 1525–1540, 2011.
- [14] Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of a state-dependent graph laplacian,” in *American Control Conference (ACC)*, July 2005.
- [15] M. C. D. Gennaro and A. Jadbabaie, “Decentralized control of connectivity for multi-agent systems,” in *IEEE Conference on Decision and Control*, December 2006.
- [16] M. M. Zavlanos and G. J. Pappas, “Potential fields for maintaining connectivity of mobile networks,” *IEEE Transactions on Robotics*, vol. 23(4), pp. 812–816, 2007.
- [17] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, “Maintaining connectivity in mobile robot networks,” *Springer Tracts on Advanced Robotics*, vol. 54, pp. 117–126, 2009.
- [18] M. A. Hsieh, A. Cowley, R. V. Kumar, and C. J. Taylor, “Maintaining network connectivity and performance in robot teams,” *IEEE Journal*, vol. 25(1-2), pp. 111–131, 2008.
- [19] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, “Decentralized estimation and control of graph connectivity for mobile sensor networks,” *Automatica*, vol. 46, pp. 390–396, 2010.

- [20] A. Ajorlou, A. Momeni, and A. G. Aghdam, "A class of bounded distributed control strategies for connectivity preservation in multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 55(12), pp. 2828–2833, 2010.
- [21] D. V. Dimarogonas and K. H. Johansson, "Bounded control of network connectivity in multi-agent systems," *IET Control Theory and Applications*, vol. 4(8), pp. 1330–1338, 2010.
- [22] A. Ghaffarkhah and Y. Mostofi, "Communications-aware navigation functions for cooperative target tracking," in *American Control Conference (ACC)*, June 2009.
- [23] T. Razafindralambo and D. Simplot-Ryl, "Connectivity preservation and coverage schemes for wireless sensor networks," *IEEE Transactions on Automatic Control*, vol. 56(10), pp. 2418–2428, 2011.
- [24] J. Reich, V. Misra, D. Rubenstein, and G. Zussman, "Spreadable connected autonomic networks (scan) - technical report cucs-016-08," tech. rep., Columbia University, New York, NY, 2008.
- [25] G. Hollinger and S. Singh, "Multi-robot coordination with periodic connectivity," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010.
- [26] M. Lindhe and K. H. Johansson, "Adaptive exploitation of multipath fading for mobile sensors," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2010.
- [27] A. Ghaffarkhah, Y. Yan, and Y. Mostofi, "Dynamic coverage of time varying environments using a mobile robot - a communication aware perspective," in *IEEE Globecom*, December 2011.
- [28] Y. Mostofi, "Decentralized communication-aware motion planning in mobile networks: An information-gain approach," *Journal of Intelligent Robot Systems*, vol. 56(2), pp. 718–740, 2009.
- [29] R. C. Palat, A. Annamalai, and J. H. Reed, "Cooperative relaying for ad-hoc ground networks using swarm uavs," in *IEEE Military Communications Conference*, 2005.

- [30] T. Schouwenaars, E. Feron, and J. How, “Multi-vehicle path planning for non-line of sight communication,” in *Proc. American Control Conference*, p. 6pp., 14–16 June 2006.
- [31] K. Kulling, “Optimal and receding-horizon path planning algorithms for communications relay vehicles in complex environments,” Master’s thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, June 2009.
- [32] Y. Yan and Y. Mostofi, “Robotic router formation in realistic communication environments- a bit error rate approach,” in *IEEE Military Communications Conference*, 2010.
- [33] N. Bezzo and R. Fierro, “Decentralized connectivity and user localization via wireless robotic networks,” in *IEEE Globecom*, December 2011.
- [34] C. Dixon and E. Frew, “Decentralized extremum-seeking control of nonholonomic vehicles to form a communication chain,” *Lecture notes in Control and Information Sciences*, vol. 369, p. 311, 2007.
- [35] C. Dixon, *Controlled Mobility of Unmanned Aircraft Chains to Optimize Network Capacity in Realistic Communication Environments*. PhD thesis, University of Colorado at Boulder, Boulder, CO, 2010.
- [36] A. S. Ibrahim, K. Seddik, and K. J. R. Liu, “Connectivity-aware network maintenance via relays deployment,” *IEEE Transactions on Wireless Communications*, vol. 8(1), pp. 356–366, January 2009.
- [37] N. Atay and B. Bayazit, “Mobile wireless sensor network connectivity repair with k-redundancy,” in *Workshop on the Algorithmic Foundations of Robotics*, December 2008.
- [38] H. Zhu, A. L. Swindlehurst, and K. Liu, “Optimization of manet connectivity via smart deployment/movement of unmanned air vehicles,” *IEEE Journal of Vehicular Technology*, vol. 58(7), pp. 3533–3546, 2009.
- [39] A. Srinivas and E. Modiano, “Joint node placement and assignment for throughput optimization in mobile backbone networks,” in *IEEE INFOCOM*, December 2008.

- [40] E. Craparo and E. M. J. P. How, “Throughput optimization in mobile backbone networks,” *IEEE Transactions on Mobile Computing*, March 2010.
- [41] N. Goddemeier, S. Rohde, J. Pojda, and C. Wietfeld, “Evaluation of potential fields mobility strategies for aerial network provisioning,” in *IEEE Globecom*, December 2011.
- [42] D. Tardioli, A. Mosteo, L. Riazuelo, J. Villarroel, and L. Montano, “Enforcing network connectivity in robot team missions,” *The International Journal of Robotics Research*, vol. 29, pp. 460–480, April 2010.
- [43] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*. Chichester, U.K.: John Wiley and Sons, 2005.
- [44] Y. Mostofi, A. G. Ruiz, A. Ghaffarkhah, , and D. Li, “Characterization and modeling of wireless channels for networked robotic and control systems - a comprehensive overview,” in *IEEE International Conference on Intelligent Robots and Systems*, 2009.
- [45] G. Warwick, “Afrl aims for mav lead.” Available: www.aviationweek.com, Accessed July 2011, 2009.
- [46] N. Wagle and E. Frew, “Spatio-temporal characterization of airborne radio frequency environments,” in *IEEE Globecom*, December 2011.
- [47] F. Xue and P. Kumar, “The number of neighbors needed for connectivity of wireless networks,” *Wireless Networks*, vol. 10, pp. 169–181, 2004.
- [48] Y. Zhang and M. Quilling, “Optimal backbone generation for robotic relay networks,” in *IEEE International Conference on Computer Communications and Networks*, 2011.
- [49] N. Ahmed, S. Kanhere, and S. Jha, “Link characterization for aerial wireless sensor networks,” in *IEEE Globecom*, December 2011.
- [50] Q. J. Krueger, “Electromagnetic interference and radio frequency interference shielding of carbon-filled conductive resins,” Master’s thesis, Michigan Technological University, Houston, MI, Feb. 2002.
- [51] M. Schwager, N. Michael, V. Kumar, and D. Rus, “Time scales and stability in networked multi-robot systems,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

- [52] S. Ponda, J. Redding, H.-L. Choi, J. P. How, M. Vavrina, and J. Vian, “Decentralized planning for complex missions with dynamic communication constraints,” in *American Control Conference (ACC)*, June 2010.
- [53] L. B. Johnson, S. Ponda, H.-L. Choi, and J. P. How, “Asynchronous decentralized task allocation for dynamic environments,” in *Proceedings of the AIAA Infotech@Aerospace Conference*, (St. Louis, MO), March 2011.
- [54] A. N. Kopeikin, A. Clare, O. Toupet, J. P. How, and M. L. Cummings, “Flight testing a heterogeneous multi-uav system with human supervision,” in *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2012.
- [55] H. C. Christmann and E. N. Johnson, “Design and implementation of a self-configuring ad-hoc network for unmanned aerial systems,” in *AIAA Infotech@Aerospace Conference*, May 2007.
- [56] H. T. Kung, C.-K. Lin, T.-H. Lin, S. J. Tarsa, and D. Vlah, “Measuring diversity on a low-atlitude uav in ground-to-air wireless 802.11 mesh network,” in *IEEE Globecom*, December 2010.
- [57] E. Yanmaz, R. Kuschnig, and C. Bettstetter, “Channel measurements over 802.11a-based uav-to-ground links,” in *IEEE Globecom*, December 2011.
- [58] A. Ribeiro, “Stochastic learning algorithms for optimal design of wireless fading networks,” in *IEEE Workshop on Signal Processing and Advances in Wireless Communications*, June 2011.
- [59] R. Olfati-Saber, A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *IEEE Transactions on Automatic Control*, vol. 95, pp. 215–233, January 2007.
- [60] J. N. Al-Karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: A survey,” *IEEE Wireless Communications*, vol. 12, pp. 6–28, 2004.
- [61] F. Kaabi and F. Filali, “Channel allocation and routing in wireless mesh networks: A survey and qualitative comparison between schemes,” *International Journal of Wireless and Mobile Networks*, vol. 2(1), pp. 104–111, 2010.
- [62] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, “Ieee 802.11s: The wlan mesh standard,” *IEEE Wireless Communications*, vol. 2, pp. 104–111, 2010.

- [63] C. Perkins and E. Belding-Royer, "Ad-hoc on-demand distance vector (aodv) routing," in *IEEE Workshop on Mobile Computing Systems*, February 1999.
- [64] C. Perkins and E. Belding-Royer, "Dynamic source routing in ad-hoc wireless networks," in *Kluwer Academic*, 1996.
- [65] C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "The optimised routing protocol for mobile ad-hoc networks: protocol specification - report n5145," tech. rep., Institut National de Recherche en Informatique et en Automatique, France, 2004.
- [66] R. G. Garroppo, S. Giordano, and L. Tavanti, "Experimental evaluation of two open source solutions for wireless mesh routing at layer two," in *International Symposium on Wireless Pervasive Computing*, 2010.
- [67] T. Salonidis, M. Garetto, A. Saha, and E. Knightly, "Identifying high throughput paths in 802.11 mesh networks: a model-based approach," in *IEEE International Conference on Network Protocols*, October 2007.
- [68] D. Decouto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks - Springer Science and Business Media Inc.*, vol. 11, pp. 419–434, 2005.
- [69] A. Ribeiro, "Notes on implementing a iee 802.11s mesh point," in *Euro NGI Workshop, Barcelona, Spain*, January 2008.
- [70] A. Bhatia and P. Kaushik, "A cluster based minimum battery cost aodv routing using multipath route for zigbee," in *IEEE Conference on Networks*, (St. Louis, MO), December 2008.
- [71] S. Toumpis and S. Gitzenis, "Load balancing in wireless sensor networks using kirchhoffs voltage law," in *IEEE INFOCOM*, April 2009.
- [72] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Transactions on Communications*, vol. 52(7), pp. 1136–1144, 2004.
- [73] R. L. Cruz and A. Santhanam, "Optimal routing, link scheduling and power control in multihop wireless networks," in *IEEE INFOCOM*, April 2003.

- [74] N. Jodeh and M. Mears, "An overview of the cooperative operations in urban terrain (counter) program," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2008.
- [75] Defense Research and Development Canada, "Human factors issues for controlling uninhabited aerial vehicles," tech. rep., 2010.
- [76] A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *The IEEE Transactions on Automatic Control (to appear)*, 2004.
- [77] J. N. Al-Karaki and A. E. Kamal, "Leader-to-formation stability," *IEEE Transaction on Robotics and Automation*, vol. 20(3), pp. 443–455, 2004.
- [78] X. Liu, S. S. Mahal, A. Goldsmith, and J. K. Hedrick, "Effects of communication delay on string stability in vehicle platoons," in *IEEE International Conference on Intelligent Transportation Systems*, August 2001.
- [79] N. Michael, M. Schwager, V. Kumar, and D. Rus, "An experimental study of time scales and stability in networked multi-robot systems," *Springer Tracts on Advanced Robotics - International Symposium on Experimental Robotics*, 2010.
- [80] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, pp. 655–661, May 2005.
- [81] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49(9), pp. 1520–1533, 2004.
- [82] M. Alighanbari, L. Bertuccelli, and J. How, "A Robust Approach to the UAV Task Assignment Problem," in *IEEE Conference on Decision and Control (CDC)*, pp. 5935–5940, 13–15 Dec. 2006.
- [83] M. Alighanbari and J. How, "Decentralized task assignment for unmanned aerial vehicles," in *IEEE Conference on Decision and Control (CDC)*, pp. 5668–5673, 12–15 Dec. 2005.
- [84] S. Sariel and T. Balch, "Real time auction based allocation of tasks for multi-robot exploration problem in dynamic environments," in *Proceedings of the AIAA Workshop on Integrating Planning Into Scheduling*, 2005.

- [85] A. Ahmed, A. Patel, T. Brown, M. Ham, M. Jang, and G. Agha, "Task assignment for a physical agent team via a dynamic forward/reverse auction mechanism," in *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, 2005.
- [86] M. L. Atkinson, "Results analysis of using free market auctions to distribute control of UAVs," in *AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*, 2004.
- [87] J. L. Ny, A. Ribeiro, and G. J. Pappas, "Adaptive communication-constrained deployment of unmanned vehicle systems - (internet draft)," tech. rep., University of Pennsylvania, Philadelphia, PA., 2011. [http://www.seas.upenn.edu/aribeiro/preprints/2012 leny etal.pdf](http://www.seas.upenn.edu/aribeiro/preprints/2012%20leny%20etal.pdf).
- [88] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics*, vol. 24(6), pp. 1416–1428, 2008.
- [89] T. Jonson, J. Pezeshki, V. Chao, K. Smith, and J. Fazio, "Applications of delay tolerant networking (dtn) in airborne networks," in *IEEE MILCOM*, 2008.
- [90] Federal Communications Commission, "The role of deployable aerial communications architecture in emergency communications and recommended next steps," tech. rep., Public Safety and Homeland Security Bureau, Washington, D.C., 2011.
- [91] T. Zajkowski, S. Dunagan, and J. Eilers, "Small uas communications mission," in *11th Biennial USDA Forest Service Remote Sensing Application Conference*, April 2006.
- [92] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, MA: Athena Scientific, 1997.
- [93] S. S. Ponda, L. B. Johnson, H.-L. Choi, and J. P. How, "Ensuring network connectivity for decentralized planning in dynamic environments," in *Proceedings of the AIAA Infotech@Aerospace Conference*, (St. Louis, MO), March 2011.
- [94] D. Bertsimas and R. Weismantel, *Optimization over integers*. Dynamic Ideas Belmont, MA, 2005.
- [95] P. R. Chandler, M. Pachter, D. Swaroop, J. M. Fowler, J. K. Howlett, S. Rasmussen, C. Schumacher, and K. Nygard, "Complexity in UAV Cooperative Control," in *American Control Conference (ACC)*, (Anchorage AK), May 2002.

- [96] S. S. Ponda, L. B. Johnson, A. N. Kopeikin, H.-L. Choi, and J. P. How, “Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams,” *IEEE Journal on Selected Areas in Communications (to appear)*, 2012.
- [97] A. S. Clare, M. L. Cummings, and L. F. Bertuccelli, “Identifying suitable algorithms for human-computer collaborative scheduling of multiple unmanned vehicles,” in *Accepted at AIAA Aerospace Sciences Meeting*, 2012.
- [98] S. S. Ponda, L. B. Johnson, A. Geramifard, and J. P. How, *The Handbook of Unmanned Aerial Vehicles (to appear)*, ch. Cooperative Mission Planning for Multi-Agent Teams. Springer, 2012.
- [99] S. Karaman, M. Faied, E. Frazzoli, and A. Girard, “Specification and planning of interactive uav missions in adversarial environments,” in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2009.
- [100] J. Boskovic, N. Knoebel, N. Moshtagh, J. Amin, and G. Larson, “Collaborative mission planning and autonomous control technology (compact) system employing swarms of uavs,” in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2009.
- [101] O. Toupet and J. P. How, “Collaborative sensor fusion and management for multiple uavs,” in *AIAA Infotech@Aerospace Conference*, 2011.
- [102] J. Wood and J. K. Hedrick, “Space partitioning and classification for multi-target search and tracking by heterogeneous unmanned aerial system teams,” in *AIAA Infotech@Aerospace Conference*, 2011.
- [103] L. Bertuccelli, H. Choi, P. Cho, and J. How, “Real-time multi-UAV task assignment in dynamic and uncertain environments,” in *AIAA Guidance, Navigation, and Control Conference (GNC)*, (AIAA 2009-5776), Aug. 2009.
- [104] L. B. Johnson, H.-L. Choi, S. S. Ponda, and J. P. How, “Allowing non-submodular score functions in distributed task allocation,” in *IEEE Conference on Decision and Control (CDC)*, Dec 2012 (submitted).
- [105] L. B. Johnson, S. Ponda, H.-L. Choi, and J. P. How, “Improving the efficiency of a decentralized tasking algorithm for UAV teams with asynchronous communications,” in *AIAA Guidance, Navigation, and Control Conference (GNC)*, August 2010.

- [106] S. Ponda, H.-L. Choi, and J. P. How, “Predictive planning for heterogeneous human-robot teams,” in *AIAA Infotech@Aerospace*, April 2010.
- [107] S. S. Ponda, L. B. Johnson, and J. P. How, “Distributed chance-constrained task allocation for autonomous multi-agent teams,” in *American Control Conference (ACC)*, June 2012 (to appear).
- [108] D. Casbeer and M. Argyle, “Hierarchical distributed task assignment for uav teams,” in *AIAA Infotech@Aerospace Conference*, 2011.
- [109] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *IEEE Proceedings*, vol. 95, pp. 215–233, January 2007.
- [110] J. P. How, B. Bethke, A. Frank, D. Dale, and J. Vian, “Real-time indoor autonomous vehicle test environment,” *IEEE Control Systems Magazine*, vol. 28, pp. 51–64, April 2008.
- [111] M. A. Hsieh, L. Chaimowicz, A. Cowley, B. Grocholsky, J. Keller, and V. Kumar, “Adaptive teams of autonomous aerial and ground robots for situational awareness,” *Journal of Field Robotics*, vol. 24(11), pp. 991–1014, 2007.
- [112] M. Dille, B. Grocholsky, S. T. Nuske, M. Moseley, and S. Singh, “Air-ground collaborative surveillance with human portable hardware,” in *AUVSI North America*, 2011.
- [113] I. Maza, K. Kondak, M. Bernard, and A. Ollero, “Multi-uav cooperation and control for load transportation and deployment,” *Journal of Intelligent and Robotic Systems*, vol. 57(1), pp. 419–449, 2010.
- [114] L. Di, “Cognitive formation flight in multi-unmanned aerial vehicle-based personal remote sensing systems,” Master’s thesis, Utah State University, Logan, UT, 2011.
- [115] I. Maza, K. Kondak, M. Bernard, and A. Ollero, “An unmanned aircraft system for automatic forest fire monitoring and measurement,” *Journal of Intelligent and Robotic Systems*, vol. 65(1), pp. 533–548, 2012.
- [116] K. D. Jones, V. Dobrokhodov, I. Kaminer, T. H. Chung, M. R. Clement, M. Kolsch, and R. Zaborowski, “Cooperative autonomy for the masses - fundamental steps toward enabling complex multi-asset missions with simple point-and-click tasking,” in *AUVSI North America*, 2011.

- [117] D. Cole, A. Goktogan, and S. Sukkarieh, “The demonstration of a cooperative control architecture for uav teams,” *Springer Tracts in Advanced Robotics - Experimental Robotics*, vol. Springer, pp. 501–510, 2008.
- [118] I. Maza, F. Caballero, J. Capitan, J. M. de Dios, and A. Ollero, “Experimental results in multi-uav coordination for disaster management and civil security applications,” *Journal of Intelligent and Robotic Systems*, vol. 61(1), pp. 563–585, 2011.
- [119] A. Huang, E. Olson, and D. Moore, “LCM: Lightweight communications and marshalling,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2010.
- [120] A. Kopeikin, O. Toupet, A. Clare, M. L. Cummings, and J. P. How, “Design of a decentralized multi-uav system with human supervision,” in *AUVSI North America*, August 2011.
- [121] M. L. Cummings, J. P. How, A. Whitten, and O. Toupet, “The Impact of Human-Automation Collaboration in Decentralized Multiple Unmanned Vehicle Control,” *Proceedings of the IEEE*, pp. 1–12, December 2011.
- [122] M. L. Cummings, “Autonomous aerial cargo utility systems.” Office of Naval Research. Available: <http://www.onr.navy.mil/Media-Center/Fact-Sheets/Autonomous-Aerial-Cargo-Utility-Systems.aspx>.
- [123] N. S. Board, “Space partitioning and classification for multi-target search and tracking by heterogeneous unmanned aerial system teams,” in *National Research Council*, 2005.
- [124] “Unmanned aircraft systems (uas) roadmap, 2005-2030,” in *OSD*, 2005.
- [125] M. L. Cummings, S. Bruni, S. Mercier, and P. J. Mitchell, “Automation architecture for single operator-multiple UAV command and control,” *The International Command and Control Journal*, vol. 1, pp. 1–24, 2007.
- [126] D. Southern, “Human-Guided Management of Collaborating Unmanned Vehicles in Degraded Communication Environments,” master’s thesis, MIT Department of Electrical Engineering and Computer Science, June 2010.
- [127] J. C. Ryan, “Assessing the performance of human-automation collaborative planning systems,” Master’s thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, June 2011.

- [128] M. L. Cummings, A. S. Clare, and C. S. Hart, "The role of human-automation consensus in multiple unmanned vehicle scheduling," *Human Factors*, vol. 52(1), pp. 17–27, 2010.
- [129] S. J. Rasmussen, M. D. Holland, and A. P. Bry, "Counter cooperative control algorithms: Challenges and lessons learned," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2008.
- [130] J. L. Franke, V. Z. Moffitt, D. Houston, J. G. Clark, and A. Lizardi, "Icarus: The construction of and lessons learned from a general-purpose autonomy system," in *AIAA Infotech@Aerospace Conference*, 2009.
- [131] W. J. Hennigan, "Military uses drone helicopter on supply mission for first time," *Los Angeles Times*, vol. January 9, 2012.
- [132] P. Smith, E. McCoy, and C. Layton, "Brittleness in the design of cooperative problem-solving systems: The effects on user performance," in *IEEE Transaction on Systems*, 1997.
- [133] A. Clare, "Dynamic Human-Computer Collaboration in Real-time Unmanned Vehicle Scheduling," master's thesis, MIT Department of Aeronautics & Astronautics, June 2010.
- [134] M. Valenti, B. Bethke, J. How, D. P. de Farias, and J. Vian, "Embedding Health Management into Mission Tasking for UAV Teams," in *American Controls Conference*, (New York, NY), June 2007.
- [135] M. Alighanbari, L. F. Bertuccelli, and J. P. How, "Filter-Embedded UAV Task Assignment Algorithms For Dynamic Environments," *AIAA Guidance, Navigation, and Control Conference (GNC)*, 2004.
- [136] W. B. Rouse, *Systems engineering models of human machine interaction*. New York, NY: North Holland, 1983.
- [137] D. K. Schmidt, "A queuing analysis of the air traffic controller's workload," *IEEE Transaction on Systems*, vol. 8(6), pp. 492–498, 1978.
- [138] M. L. Cummings and S. Guerlain, "Developing operator capacity estimates for supervisory control of autonomous vehicles," *Human Factors*, vol. 49(1), pp. 1–15, 2010.

- [139] M. L. Cummings and C. E. Nehme, “Modeling the impact of workload in network centric supervisory control settings,” *Neurocognitive and physiological factors during high-tempo operations*, vol. Surrey, UK, pp. 23–40, 2010.
- [140] J. P. How, C. Fraser, K. C. Kulling, L. F. Bertuccelli, O. Toupet, L. Brunet, A. Bachrach, and N. Roy, “Increasing autonomy of UAVs,” *Robotics & Automation Magazine, IEEE*, no. 2, pp. 43–51, 2009.
- [141] A. K. Whitten, “Decentralized planning for autonomous agents cooperating in complex missions,” Master’s thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, Sept. 2010.
- [142] J. Curtis, “Churning: Repeated optimization and cooperative instability,” *Recent Developments in Cooperative Control and Optimization*, p. 105, 2004.