

AUTONOMY FOR SENSOR-RICH VEHICLES:  
INTERACTION BETWEEN SENSING AND CONTROL ACTIONS

A DISSERTATION  
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND  
ASTRONAUTICS  
AND THE COMMITTEE ON GRADUATE STUDIES  
OF STANFORD UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

Gabriel M. Hoffmann

September 2008

© Copyright by Gabriel M. Hoffmann 2009  
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Claire J. Tomlin) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Sebastian Thrun)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Stephen M. Rock)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

---

(Dimitry Gorinevsky)

Approved for the University Committee on Graduate Studies.

# Abstract

While autonomous vehicles have the potential to enable many revolutionary technologies, assisting people through unprecedented automation, they introduce many challenges in control system design. One step toward increasing their autonomy is to formulate an optimization problem that exploits models connecting the effects of the sensing and control systems to optimize the performance of the overall system. Using these models, the robots are able to intelligently experiment with their environment to work toward achieving their goals.

This thesis presents techniques to directly exploit the connection between sensing and control, focusing on algorithms for mobile sensors and mobile sensor networks. Algorithms are developed to compute information theoretic quantities using a particle filter representation of the probability distributions over the states being estimated. To make the approach scalable to increasing network size, single-node and pairwise-node approximations to the mutual information are derived for general probability density models, with analytical bounds on the error incurred, and computation time that is polynomial in the number of sensors. The pairwise-node approximation is proven to be a more accurate objective function than the single-node approximation. A decentralized optimization algorithm is presented to implement these techniques, using a novel collision avoidance method incorporating hybrid control.

The characteristics of these algorithms are explored in simulation of autonomous search using three sensing modalities: range, bearing, and magnetic rescue beacon. For each sensing modality, the behavior of these non-parametric methods are compared and contrasted with the results of linearized methods. The proposed methods

produce similar results in some scenarios, yet they capture effects in more general scenarios not possible with linearized methods. Monte Carlo results demonstrate that the pairwise-node approximation provides superior performance to the single-node approximation.

To motivate and demonstrate these algorithms, a fleet of quadrotor helicopters is designed and used, the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC). The algorithms are implemented as an autonomous guidance system onboard these aircraft to automate search for an avalanche rescue beacon. Vehicle design, dynamics, and control are presented. Experimental results demonstrate improved vehicle control over the state-of-the-art, and the ability to autonomously search for a lost rescue beacon.

*To Katie.*

# Acknowledgements

The journey of the graduate school experience that led to this dissertation has been rewarding and fulfilling, thanks in large part to the support, encouragement, and friendship of many people around me, all of whom I owe a debt of gratitude.

In particular, I thank my advisor, Prof. Claire Tomlin, whom I have had the honor to work with. Her academic enthusiasm and inquisitive nature have been an inspiration throughout the process. She created an immensely supportive research environment to work in, and her insight and guidance helped me find the road to pursue, both to find and to follow my interests.

I would like to thank the members of my reading committee, Prof. Steve Rock, Prof. Sebastian Thrun, and Prof. Dimitry Gorinevsky. They provided valuable feedback and guidance throughout this process. Conversations with Sebastian on aspects of probabilistic robotics were always fascinating, and I also thank him for the experience that he enabled through his leadership of the Stanford Racing team. I would also like to thank Ilan Kroo for being on my defense committee, and the feedback that he provided.

The Department of Aeronautics and Astronautics was a dynamic place to work, thanks to the support of many people. In particular, I thank Prof. Bob Canon for his insight and our many discussions. I thank Sherann Ellsworth and Dana Parga who not only helped make STARMAC possible, but helped make the controls group a happy, friendly place for us all. I also thank Ralph Levine, Robin Murphy, Lynn Kaiser, and Jayanthi Subramanian for their support behind the scenes.

I have had the pleasure of working closely with many labmates in the Hybrid Systems Lab. I thank Jung Soon Jang and Rodney Teo for all they have taught me,

and passing on their experience with UAVs. I must especially thank Steve Waslander, whom I have worked with on STARMAC from the beginning of the project. Discussions with Steve helped guide my work, and I have enjoyed the process of learning through the course of our conversations. I also thank the many other students I worked with on STARMAC. In particular, Haomiao Huang, Mike Vitus, Vijay Pradeep, and Jeremy Gillula helped make the flight demonstrations presented in this dissertation possible. Additionally, Dev Rajnarayan, David Dostol, Dave Shoemaker, Paul Yu, and Justin Hendrickson helped the STARMAC design take shape. I thank Kaushik Roy, Robin Raffard, Alessandro Abate, Hamsa Balakrishnan, Alexandre Bayen, Peter Brende, Ronojoy Gosh, Gökhan İnalhan, and Meeko Oishi for helping make this lab a supportive environment, filled with interesting discussions.

I have been fortunate to collaborate with other labs on campus. My experiences working with the Stanford Racing Team on the DARPA Grand Challenge were exceptionally intriguing and rewarding. I thank Mike Montemerlo, David Stavens, and the many other collaborators from Stanford, Volkswagen, and Intel who made the team what it was. I have also had many useful discussions with friends in the Aerospace Robotics Lab. In particular, I enjoyed talking with Alan Chen and Jack Langelaan about our research. In Ilan's group, Peter Kunz and Stefan Bieniawski helped introduce me to aircraft design for UAVs.

My family and friends have been ever supportive. I thank my parents, Loren and Anita, for the love, encouragement, and strength they have provided. I thank my sister, Rebecca, whose support was invaluable. I also thank my friends. In particular, I thank James Diebel for our many fascinating conversations and adventures together, and Jill Weir for both helping me enjoy California and for helping inspire part of this research, as a former ski patrol member.

My greatest thanks go to my wife, Katie, whose loving support has been the foundation for the joy and energy that I feel every day. Her help and kindness made this journey possible. She has been a loving companion and best friend through every step of the way.



# Contents

<b>Abstract</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Autonomous Vehicle Model . . . . .	2
1.2 Mobile Sensors . . . . .	6
1.3 Outline . . . . .	8
1.4 Contributions . . . . .	9
<b>2 Robotic Testbed: STARMAC</b>	<b>11</b>
2.1 Background . . . . .	14
2.2 Vehicle Hardware . . . . .	16
2.2.1 Propulsion . . . . .	17
2.2.2 Sensors . . . . .	20
2.2.3 Control Electronics . . . . .	21
2.2.4 Frame . . . . .	23
2.3 Quadrotor Dynamics . . . . .	25
2.3.1 Thrust and Torque Generation . . . . .	25
2.3.2 Inertial Dynamics . . . . .	27
2.4 Quadrotor Control . . . . .	29
2.4.1 Attitude and Altitude Control . . . . .	29
2.4.2 Position and Trajectory Tracking Control . . . . .	33
2.5 Summary . . . . .	37

<b>3</b>	<b>Mobile Sensor Guidance</b>	<b>39</b>
3.1	System Definition . . . . .	40
3.2	Objective . . . . .	42
3.3	Information Theory with Particle Filters . . . . .	45
3.3.1	Particle Filter . . . . .	46
3.3.2	Determining Mutual Information from Particle Sets . . . . .	47
3.4	Mutual Information Approximations . . . . .	50
3.5	Distributed Control . . . . .	56
3.5.1	Iterative Optimization . . . . .	57
3.5.2	Decoupled Optimization . . . . .	59
3.6	Summary . . . . .	60
<b>4</b>	<b>Collision Avoidance</b>	<b>62</b>
4.1	Problem Formulation . . . . .	65
4.2	Optimal Control Approach . . . . .	66
4.3	Two Vehicle Collision Avoidance . . . . .	67
4.3.1	Optimal Control Input . . . . .	68
4.3.2	Avoid Set . . . . .	69
4.4	Multi-Vehicle Collision Avoidance . . . . .	72
4.4.1	Optimal Control Input . . . . .	72
4.4.2	Avoid Set . . . . .	72
4.5	Collision Avoidance Results . . . . .	76
4.5.1	Two Vehicles . . . . .	77
4.5.2	Many Vehicles . . . . .	77
4.5.3	Flight Experiments . . . . .	78
4.6	Summary . . . . .	84
<b>5</b>	<b>Mobile Sensor Applications</b>	<b>85</b>
5.1	Application to Sensing Modalities . . . . .	85
5.1.1	Bearing-Only Sensors . . . . .	86
5.1.2	Range-Only Sensors . . . . .	96
5.1.3	Magnetic Dipole Sensors (Rescue Beacons) . . . . .	100

5.2	Flight Experiments . . . . .	109
5.2.1	Vehicle Software . . . . .	109
5.2.2	System Configuration . . . . .	111
5.2.3	Flight Tests . . . . .	114
<b>6</b>	<b>Conclusion</b>	<b>123</b>
6.1	Summary . . . . .	123
6.2	Future Directions . . . . .	124
<b>A</b>	<b>Vehicle State Estimation</b>	<b>126</b>
<b>B</b>	<b>Quadrotor Aerodynamics</b>	<b>129</b>
B.1	Total Thrust . . . . .	130
B.2	Blade Flapping . . . . .	134
<b>C</b>	<b>Information Theory</b>	<b>140</b>
C.1	Kraft Inequality . . . . .	140
C.2	Optimal Codes . . . . .	141
	<b>Bibliography</b>	<b>143</b>

# List of Tables

2.1	STARMAC II Quadrotor Helicopter Components . . . . .	18
4.1	Collision Avoidance Computation Time . . . . .	77

# List of Figures

1.1	Two quadrotor helicopters from the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC) in autonomous hover control at GPS waypoints. . . . .	2
1.2	The autonomous vehicle “Stanley” traversing off-road terrain during the DARPA Grand Challenge 2005 race, and descending “Beerbottle Pass”. . . . .	3
1.3	Block diagram of the information-seeking control problem. . . . .	4
1.4	Control system components. . . . .	5
2.1	A STARMAC II quadrotor helicopter. . . . .	12
2.2	Quadrotor helicopter control method. . . . .	13
2.3	STARMAC II vehicle and its components. . . . .	17
2.4	Thrust test stand used to measure motor thrust, side force, torque, voltage and current. . . . .	19
2.5	Flow of information in the STARMAC II electronics. . . . .	21
2.6	Effect of shrouds on yaw control. . . . .	24
2.7	Free body diagram of the moments and forces acting on rotor $j$ . . . .	27
2.8	Free body diagram of a quadrotor helicopter. . . . .	28
2.9	Attitude control results from indoor flight tests for roll, pitch and yaw. . . . .	31
2.10	Altitude command tracking in indoor flight tests. . . . .	32
2.11	Indoor autonomous hover performance. . . . .	33
2.12	The quadrotor travels along path segment $\Lambda_k$ from waypoint $\mathbf{x}_k^d$ to $\mathbf{x}_{k+1}^d$ , applying along and cross-track control inputs to track the path. . . . .	34
2.13	Tracking a trajectory indoors, at 0.5 m/s, with an error of under 0.1 m. . . . .	35

2.14	Tracking a trajectory outdoors, at 2.0 m/s, with an error of under 0.5 m.	36
2.15	Three STARMAC quadrotor helicopters performing position control at GPS waypoints.	37
3.1	Directed graphical model of the estimator.	44
3.2	Graphical depiction of a 1-D particle filter.	47
4.1	A mid-air collision between two out of three remote-piloted quadrotor helicopters operating in close proximity.	63
4.2	Relative states of two acceleration constrained quadrotor helicopters.	66
4.3	The keepout set and the avoid set for two-vehicle collision avoidance, in the rotated coordinate frame.	70
4.4	Collision avoidance control input diagram.	73
4.5	Simulation of two quadrotor helicopters approaching one another, with guaranteed separation using the collision avoidance algorithm.	78
4.6	Simulation of 8 vehicles tracking randomly generated trajectories, using the many-vehicle collision avoidance law.	79
4.7	Separation distances between each vehicle for the eight vehicle simulation in Fig. 4.6.	80
4.8	Simulation of 3 rings of 16 vehicles converging toward the same point, with the outermost rings moving fastest.	81
4.9	Separation distances between each vehicle for the 48 vehicle simulation in Fig. 4.8.	81
4.10	Simulation of a ring of 64 vehicles converging toward a center.	82
4.11	Automatic collision avoidance flight experiment using $d_{min} = 2$ m with human control inputs attempting to cause collisions.	83
4.12	Separation according to GPS data for the flight experiment shown in Fig. 4.11.	84
5.1	Bearings-only measurement model.	87
5.2	Optimal configurations of range sensors with linear Gaussian estimators.	91
5.3	Optimal configurations of range sensors, with random observation sides.	92

5.4	Optimal configurations of range sensors for a prior distribution with 5 times more uncertainty in the $x$ direction than the $y$ . . . . .	92
5.5	Simulation of a bearings-only target search using particle filter mutual information and the pairwise-node approximation. . . . .	94
5.6	Simulation of a bearings-only target search using the pairwise-node approximation for distributed control and collision avoidance to allow the optimization to be decoupled. . . . .	95
5.7	Summary of results for 1000 trials of bearings-only target localization.	96
5.8	Range-only measurement model. . . . .	97
5.9	Simulation of a range-only target search using particle filter mutual information and the pairwise-node approximation. . . . .	99
5.10	Contour plot of mutual information gain for linear approximation and for particle filter. . . . .	101
5.11	Avalanche rescue beacon magnetic field . . . . .	102
5.12	Rescue beacon measurement model. . . . .	103
5.13	Simulation of a rescue beacon search using particle filter mutual information and the pairwise-node approximation. . . . .	106
5.14	Mutual information gain contours for particle filter distribution. . . .	107
5.15	Comparison of mutual information contours for single- and pairwise-node approximations. . . . .	108
5.16	STARMAC control software diagram. . . . .	110
5.17	Tracker DTS Avalanche Rescue Beacon Transceiver circuitry. . . . .	111
5.18	Example polar plot of measured received signal strength from a Tracker DTS avalanche rescue beacon. . . . .	112
5.19	Comparison of yaw angle estimate using the IMU to yaw angle estimate from the processed rescue beacon measurements. . . . .	113
5.20	A STARMAC quadrotor helicopter with a avalanche rescue beacon transceiver. . . . .	114
5.21	Flight demonstration of avalanche rescue beacon search and localization by a STARMAC quadrotor helicopter in a $6 \times 10$ m field. . . . .	116

5.22	Two STARMAC quadrotor helicopters flying with avalanche rescue beacon transceivers. . . . .	118
5.23	Flight demonstration of avalanche rescue beacon search and localization by two STARMAC quadrotor helicopters in a $9 \times 9$ m field. . . .	119
B.1	Thrust dependence on angle of attack and vehicle speed. . . . .	131
B.2	Predicted ideal thrust and measured climb thrust with vertical velocity. . . . .	133
B.3	Effect of angle of attack and velocity on thrust in flight. . . . .	134
B.4	Blade flapping with stiff rotor blades modeled as hinged blades with offset and spring. . . . .	135
B.5	Static measurement of blade flapping angle. . . . .	138
B.6	Effect of blade flapping. . . . .	138



# Chapter 1

## Introduction

The technological capabilities for components of autonomous vehicles are ever increasing, with continual advances in the capabilities of their sensor suites and computational resources. With these resources, they can perceive and analyze the world around them. With the sensing and actuation capabilities found in current robotic systems, further automation could present many benefits, from reducing the workload of the operators, to enabling increasingly complex and capable systems, to potentially increasing safety and efficiency through improved sensing and reasoning. However, for the vehicles to fully exploit these resources to autonomously interact with uncertain surroundings, their software must consider the interaction between sensing and control actions to acquire knowledge about the surroundings to achieve the mission goals.

To address this challenge, this dissertation proposes autonomous guidance and control algorithms for mobile sensors and mobile sensor networks—vehicles that exist for the purpose of acquiring useful information. Algorithms are developed such that the vehicles not only cope with the uncertainty in their surroundings, but act to reduce that uncertainty to improve system performance. The guidance algorithms exploit a model of the interaction between the sensing and control actions of the vehicles to optimize the rate at which information is acquired. Non-parametric estimators, particle filters, are used to directly estimate the information available for any observation the sensors could make. Techniques for distributed control of mobile sensor



Figure 1.1: Two quadrotor helicopters from the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC) in autonomous hover control at GPS waypoints.

networks are proposed using novel approximation techniques and collision avoidance laws. These methods are applied to autonomous search and rescue in simulations and in flight experiments of the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC), shown in Fig. 1.1, demonstrating the characteristics and the utility of these approaches.

## 1.1 Autonomous Vehicle Model

This section presents the model of autonomous vehicles used in this thesis to develop algorithms for mobile sensors. First, to motivate the focus of the model, consider currently fielded robotic systems operating in uncontrolled environments. They typically require human supervision. Even unmanned aerial vehicles (UAVs) are commonly controlled remotely by at least one pilot, and often by additional crew that interpret sensors and control auxiliary functions.

Recently, autonomous ground vehicles with technology at the forefront of current capabilities competed in the DARPA Grand Challenge, a 132 mile off-road race—with



Figure 1.2: The autonomous vehicle “Stanley” traversing off-road terrain during the DARPA Grand Challenge 2005 race, and descending “Beerbottle Pass”.

nobody in the vehicles or remotely controlling them [11, 20]. These vehicles were required to accurately navigate unknown terrain while completing the race in under 10 hours. In 2004, no teams completed the race. In 2005, out of the dozens of initial competitors, only four teams completed the race. Our entry, Stanley, demonstrated the ability to accurately perceive terrain quality and navigate around obstacles at aggressive speeds, winning the race in just under 7 hours [94]. One of the keys to Stanley’s success was programming the guidance system so that, based on testing experiences, the vehicle would be likely to acquire sufficient measurements of the surroundings to enable safe control [19, 42, 92]. However, if Stanley were to encounter a situation not experienced in testing, such as exceptionally rough terrain near a cliff edge, the planner would need to accept the high uncertainty and the outcome would be unpredictable.

The leap in technology to go from autonomous vehicles guided by GPS and inertial navigation systems to vehicles that can safely and reliably sense, interpret, and react to unknown environments requires systems to make useful measurements of the environment in order to build up knowledge of the surroundings. To automate this task requires programming the vehicles to take control actions which reduce their uncertainty.

Specifically, how can robotic systems be programmed to go beyond coping with

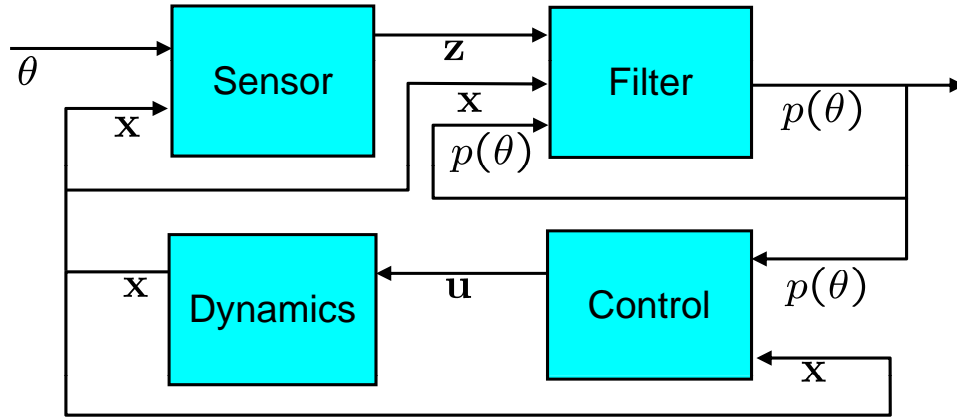


Figure 1.3: As in a standard control system, the state vectors  $\mathbf{x}$  of the vehicles are manipulated using the control inputs  $\mathbf{u}$ . Unlike a typical control system, the information-seeking controller receives the full probability distribution  $p(\theta)$  of the target state estimate vector  $\theta$ , rather than only the expected value. The future value of  $\mathbf{x}$  can be controlled such that future sensor measurements  $\mathbf{z}$  yield the greatest expected reduction in the uncertainty of  $p(\theta)$ , based on sensor models. The vehicles maximize the information gained about the target state while minimizing the number of future measurements required.

the uncertainty in their surroundings—to act to gain information to reduce uncertainty and to improve both control and overall system performance? The underlying tenet of stochastic control theory, the certainty equivalence principle, allows us to separate the design of optimal controllers and optimal estimators—for linear systems with Gaussian noise [65]. However, for robotic systems operating in unknown environments, these conditions of linearity and Gaussian noise frequently do not hold. This is what leads to coupled sensing and control problems.

The framework used to represent autonomous vehicles in this dissertation is shown in Fig. 1.3. These components comprise a closed-loop system that interacts with the surrounding environment. The vehicle dynamics determine the response of the vehicle to control inputs, the sensors make measurements of the vehicle state and the surrounding environment, the sensor measurements are used in a filter that estimates the state of the vehicle and the surrounding environment, and the control system uses

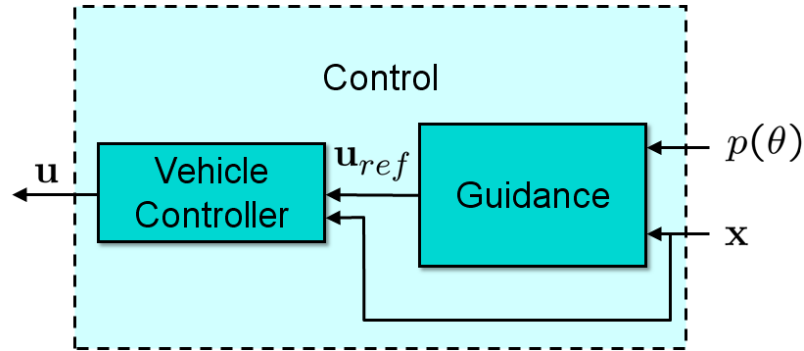


Figure 1.4: The control block in Fig. 1.3 represents a general system architecture. The algorithms proposed in this work decompose the control block into a guidance component and a vehicle controller component. The guidance component optimizes the actions of the vehicle, and the vehicle controller component tracks the resulting reference commands,  $\mathbf{u}_{ref}$ . This architecture reduces the complexity of the optimization problem for the guidance component.

the result of the filter to generate new control inputs.

The control architecture considered in this work is that of a distinct guidance system and vehicle control system, as depicted in Fig. 1.4. The task of the guidance algorithm is to ensure the vehicle accomplishes its mission using the results of the estimation system to produce reference commands for the vehicle control system. For autonomous vehicles interacting with uncertain environments, the quality of the information available from the sensor is typically a function of the actions the vehicles take.

One step toward increasing the autonomy of robotic systems is to exploit models connecting the effect of the sensing and control systems. By using such models for control, the robot is able to intelligently experiment with its environment to work toward achieving its end goal. This chapter proceeds with detailing the background and challenges for mobile sensors and mobile sensor networks. These applications are the basis for the algorithms developed in subsequent chapters.

## 1.2 Mobile Sensors

Mobile sensors are a prototypical example of coupled sensing and control problems. They have the potential to deliver exciting new capabilities, both through their ability to move sensors to vantage points that are rich sources of information, and their ability to act as networks that make simultaneous measurements in multiple locations. Mobile sensors have the potential to improve the state of the art in surveillance, reconnaissance, and scientific discovery of the environment. They could facilitate disaster prevention, through missions such as structural health monitoring, and emergency response, by providing situational awareness. The control objective is to search for information quickly, safely, and reliably. An example of one such system is the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC), presented in detail in Chapter 2. STARMAC is comprised of six quadrotor helicopters, two of which are shown in Fig. 1.1. The testbed is outfitted with avalanche rescue beacon receivers that are used for experiments in Section 5.2 to search for an avalanche rescue beacon transmitter.

To automate the task of searching for a target, several computational challenges arise. First, there is the task of representing information. Typically, there is low prior information available about the target object's state. Search regions can be complicated to represent. As the search progresses, the target state probability distribution often requires a more intricate model than can be represented by a parametric distribution, such as Gaussian. Further, the mapping between sensor observations and the physical world, even for simple sensors, is frequently a nonlinear function, such as the arctan function for bearing measurements, and the  $\mathcal{L}^2$  norm for range measurements. Second, there is the challenge of formulating the optimal control problem. The linear model, quadratic cost, and Gaussian distribution (LQG) assumptions that lead to the certainty equivalence principle, separating the estimation and control problems, are typically not valid in this problem [65]. Third, there is the difficulty of cooperatively controlling the mobile sensors. In order to improve network performance, it would be desirable to add more sensors to the network. However, the computational cost of optimizing the actions of the network must remain bounded. Also, safety requirements

must be satisfied, such as guaranteed collision avoidance between vehicles. They must maintain a safe separation distance under the constraints of their dynamics.

The goal of the search problem differs from a typical control system. The direct measurement of success is not the ability to track a trajectory. Rather, as depicted in Fig. 1.3, it is to maximize the likelihood of localizing the target as quickly as possible. This is a stochastic optimal control problem, where control inputs regulate both the dynamics of the system and the information gained by sensing, as discussed in work on the dual control problem [25] and on extremum-seeking control [4]. Several stochastic optimal control problems have been solved in the literature by simplifying sensor and motion models, such as the LQG problem mentioned above. The assumptions of the LQG problem have been extended to the mobile sensor search problem, with nonlinear models, using the Extended Kalman Filter (EKF), which linearizes the motion and measurement models [93]. Search methods with an EKF use metrics of the expected estimation covariance following a control action, often in an information theoretic context [34, 15, 29, 63]. A feedback controller has been formulated for the dual control problem using an EKF with assumptions rendering the solution suboptimal, but solvable on-line [53].

Although EKF approaches are computationally efficient, they use linearized measurement models, rely on a Gaussian noise assumption, and require a guessed initial solution. This can lead to underestimation of the covariance, biased estimates, and divergence of the filter [33, 93, 78, 84]. These drawbacks can be mitigated through a number of methods, but they cannot be eliminated [78, 32]. The EKF methods also approximate the structure of posterior distributions with only a mean and a covariance, discarding additional available information. One method to improve on EKF performance has been demonstrated using grid cell discretization for estimation, although it relies on a probability of detection model for the sensor, rather than a true sensor model [10]. The work presented in this dissertation also uses metrics of the underlying estimator, although by using a particle filter as the estimator, the nonlinear estimation performance can be improved, more information can be captured, and explicit sensor models can be incorporated.

Information theoretic cost metrics have been used to manage sensors [103], and

have led to algorithms to control sensor networks for information gathering over an area by parameterizing the motion of collectives of vehicles [60]. The optimal probing control law to minimize Shannon entropy for the dual control problem has been shown to be the input that maximizes mutual information [26]. A property relating probability distributions, the alpha-divergence, has been computed for particle filters and applied to manage sensors with binary measurements, though scalability in sensor network size was not addressed, and the Shannon entropy was only found in the limit of the presented equations [55]. Gaussian particle filters have been used with a mutual information objective function, though the technique approximates the posterior probability distribution as Gaussian at every update [102]. Particle filters have also been used with linear interpolation to approximate the posterior distribution to compute expected uncertainty [86].

The work presented in this dissertation develops methods to make the information theoretic ideas of previous work tractable and scalable for real-time control of a mobile sensor network for general sensors, dynamics, and available prior knowledge. Given a particular configuration of sensors, these techniques exploit the structure of the probability distributions of the target state and of the sensor measurements to compute control inputs leading to future observations that minimize the expected future uncertainty of the state of interest.

### 1.3 Outline

This thesis proceeds in Chapter 2 with the design and control of STARMAC quadrotor helicopters. These vehicles are used for example applications and experiments throughout the thesis. The vehicle hardware architecture is presented first. Then the vehicle dynamics are derived, and algorithms are designed for low level vehicle control given the challenges associated with autonomous control. Experimental results of control are presented.

Chapter 3 proceeds to address the pure challenge of autonomously acquiring information. Mobile sensing platforms are considered, which exist, in essence, to seek information. The task of controlling mobile sensors and networks of such sensors



is analyzed. To handle the nonlinearities and structure typical of perception tasks, particle filters are used. Information theoretic terms are derived for use with these non-parametric filters, and algorithms to enable scalable control are derived. To enable the use of algorithms for acceleration-constrained vehicles, a computationally efficient collision avoidance algorithm is presented in Chapter 4, along with simulation and experimental results.

In Chapter 5, extensive simulations using three characteristic sensors are presented and compared to state-of-the-art methods using linearized, Gaussian schemes. Experimental results with quadrotor helicopters demonstrate the use of these particle filter methods with avalanche rescue beacons, sensors which detect magnetic dipoles.

The thesis is concluded with Chapter 6, a discussion of the results presented here, and directions for future work.

## 1.4 Contributions

The contributions of this thesis have two main themes. First, the development of a robotic testbed is presented. Second, challenges in coupled sensing and control problems are addressed. Several solutions are presented, including computing information theoretic quantities using a particle filter, performing optimal distributed information-seeking using a mobile sensor network, performing decentralized control with a mobile sensor network, and vehicle collision avoidance. These solutions are simulated and demonstrated in flight experiments. In more detail:

- **Information theoretic control algorithms**

The theoretical contributions in this area are four-fold. First, algorithms are designed to compute information theoretic quantities for a particle filter representation of probability distributions. Second, single-node and pairwise-node approximations are derived for the mutual information available in a mobile sensor network, with general probability density models, analytical bounds on the error incurred, and computation time that is polynomial in the number of sensors. Third, the pairwise-node approximation is proven to be a more accurate

objective function for mutual information optimization than the single-node approximation. Fourth, this work is extended to a decentralized control algorithm using a novel hybrid control collision avoidance method to guarantee safety.

- **Analysis of information theoretic control algorithms through simulation**

The characteristics of the information theoretic control algorithms are explored for three different sensing modalities: range, bearing, and magnetic rescue beacon. A fleet of quadrotor helicopter using these techniques is simulated, and the results are compared to those of linear Gaussian schemes. The proposed methods produce similar results in some scenarios, yet capture effects in more general scenarios not possible with linearized methods. Monte Carlo results demonstrate that the pairwise-node approximation provides superior performance to the single-node approximation.

- **Autonomous quadrotor helicopters**

The development of the STARMAC testbed is presented. This fleet of six quadrotor helicopters is capable of flying multi-agent missions indoors and outdoors, and can carry sufficient sensing and computing resources not only to localize and control the aircraft, but also to enable higher levels of vehicular autonomy. The design methodology for the vehicles is given, and the dynamics are presented based on analytical and experimental results. The control system design is capable of rejecting disturbances typical of quadrotor actuation, and experimental results demonstrate the accuracy of the system.

- **Validation of information theoretic control algorithms through experiment**

Particle filter information theoretic methods are implemented in flight software onboard the STARMAC testbed. Experimental results demonstrate the ability to quickly localize an avalanche rescue beacon. The ability to optimally sense the magnetic dipole demonstrates the benefits of computing information using the particle filter approach.

## Chapter 2

# Robotic Testbed: STARMAC

This chapter presents the Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC), a fleet of quadrotor helicopters designed to be capable mobile sensors. The use of these vehicles as mobile sensors motivates the information-seeking guidance algorithms presented in Chapter 3. In Chapter 4, a collision avoidance algorithm is developed using the acceleration-constrained dynamics of quadrotors to enable efficient use of the algorithms from Chapter 3. In Chapter 5, their dynamics are used as the basis for simulations of information-seeking guidance algorithms, and the vehicles are used to fly mobile sensor guidance experiments using these algorithms with avalanche rescue beacon receivers.

The use of quadrotor helicopters as autonomous platforms has been envisaged for a variety of applications both as individual vehicles and in multi-agent teams, including surveillance, search and rescue, and mobile sensor networks [43]. Quadrotor helicopters are rotorcraft with two pairs of counter-rotating, fixed-pitch rotors located at the four corners of the aircraft, as shown in Fig. 2.1. They are controlled by varying the rotational speed of the rotors to manipulate the thrust, as depicted in Fig. 2.2. Pitch and roll angles are controlled using moments generated by differential thrust between rotors on opposite sides of the vehicle, and the yaw angle is controlled using the difference in reaction torques between the pitch and roll rotor pairs. Vertical position is controlled with the total thrust of all rotors, and lateral acceleration is achieved through the pitch and roll of the aircraft.

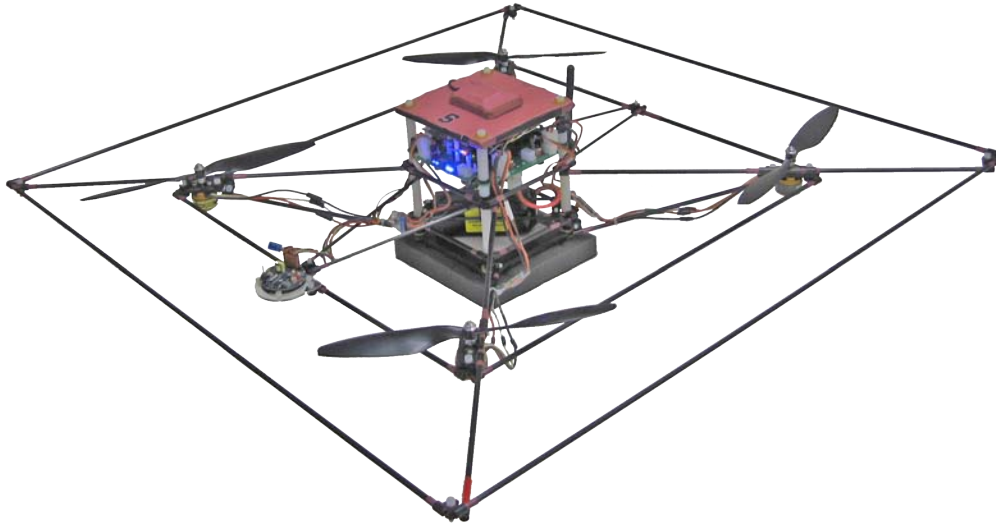


Figure 2.1: A STARMAC II quadrotor helicopter.

The quadrotor design has advantages over comparable vertical take off and landing (VTOL) UAVs, such as helicopters. First, quadrotors use variable-speed, fixed-pitch rotors for vehicle control, rather than mechanical control linkages, simplifying fabrication and maintenance. Second, the use of four rotors ensures that individual rotors are smaller in diameter than the equivalent main rotor on a helicopter. The individual rotors store less kinetic energy, mitigating the risk posed should they entrain any objects. The small rotors can also be enclosed in a protective frame, permitting flights indoors and in obstacle-dense environments with reduced risk of damaging the vehicle, its operators, or its surroundings. These benefits accelerate the design and test flight process by allowing testing to take place indoors or out, by inexperienced pilots, and with a short turnaround time for recovery from incidents.

Taking advantage of the benefits of quadrotors, STARMAC has been developed with the aim of being an easy-to-use and reconfigurable proving ground for novel algorithms for multi-agent applications. The testbed is comprised of six STARMAC II quadrotors. Improvements over previous quadrotor designs allow accurate vehicle control in both indoor and outdoor autonomous operation, with sufficient reconfigurability and excess payload capacity to carry sensing and computational resources that enable onboard execution of sophisticated multi-vehicle missions.

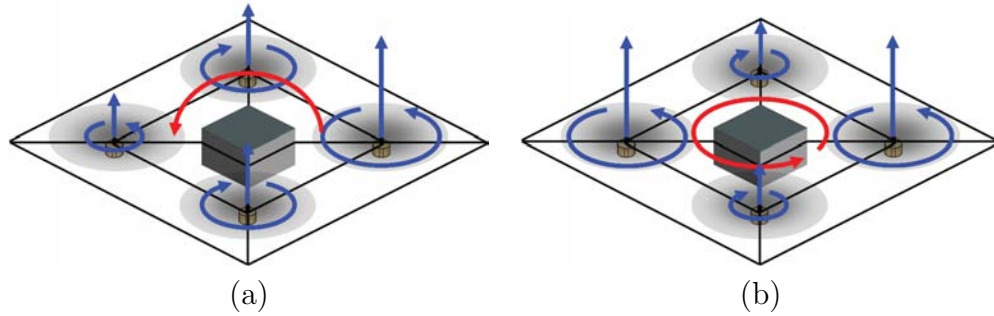


Figure 2.2: Quadrotor helicopters are controlled by varying thrust at each rotor to produce (a) roll or pitch axis torques, and (b) yaw axis torque.

Previous work in autonomous quadrotor helicopters was largely for indoor environments, with few disturbances and low speeds. The work presented in this chapter focuses on a novel quadrotor helicopter design that is capable of flying both indoors and outdoors with accurate control, and can carry sufficient sensing and computing resources not only to localize and control the aircraft, but also to enable higher levels of vehicular autonomy.

This section proceeds as follows. Section 2.1 presents a summary of the development history of quadrotors, as well as a review of related work in autonomous helicopter and quadrotor research and control. Then, the design of STARMAC II is presented in Section 2.2 to provide an understanding of the sensing and actuation capabilities for subsequent sections [41]. A nonlinear dynamic model of the quadrotor helicopter is developed in Section 2.3, based on flight experiments, bench experiments, and previous work on helicopter aerodynamics [72, 59, 81]. Finally, a comprehensive set of vehicle control system designs are presented and demonstrated in both indoor and outdoor flight tests [44]. The vehicle control system is designed to accept commands from the autonomous guidance system, removing much of the burden from potentially complex optimizations. Updates from the guidance system may occur at a slower rate than required to control vehicle dynamics, and the control system mitigates disturbances. The vehicle control system enables the more sophisticated guidance system to interpret the current state of the vehicle and the sensors to determine the correct action to take; the topic of subsequent chapters.

## 2.1 Background

The first flight-capable quadrotor designs appeared as early as the 1920's [59], though no practical versions were built until more recently. The only manned quadrotor to leave ground effect was the Curtiss-Wright X-19A, completed in 1963, though it lacked a stability augmentation system to reduce pilot work load and development was halted at the prototype stage [3]. Recent advances in microprocessor capabilities and in micro-electro-mechanical system (MEMS) inertial sensors have spawned a series of remote control (RC) quadrotor toys, such as the Roswell flyer (HMX-4) [2], and Draganflyer [23]. These vehicles often typically accept attitude rate and total thrust by human RC pilots.

Building on the successes of RC quadrotors, many groups are developing quadrotor UAVs [40, 2, 8, 35, 24, 74, 77, 79]. The vehicles range in mass from 0.3 to 4.0 kg, and demonstrate a range of designs and control techniques. Many of these groups have used proportional-derivative (PD) and proportional-integral-derivative (PID) control laws. Although many have also used nonlinear control techniques, few have demonstrated controlled flights with attitude angles out of the linear regime, or speeds fast enough to experience variation in aerodynamic effects.

Among these projects, several groups have achieved control with external tethers and stabilizing devices. One such system, using an HMX-4, was flown with horizontal motion constraints [2]. To achieve hover, backstepping control was used for the vehicle's roll, pitch, and lateral position, and linear control was used for yaw and altitude. State estimation was accomplished with an offboard computer vision system. A second project used a tethered POLYHEMUS magnetic positioning system [12]. Trajectory tracking flights were demonstrated at speeds of 0.03 m/s or less. Lateral position was controlled using a series of nested saturation functions. Altitude was controlled with a feedback linearization term comprised of a linear controller with nonlinear compensation for vehicle tilt and an offset for gravity—a useful technique common to many testbeds. A third project, with tethered power, used outward facing IR and ultrasonic rangefinders to perform collision avoidance, a robust internal-loop compensator for control, and offboard computer vision for positioning [77].

Working toward a more autonomous vehicle, many groups have demonstrated untethered position controlled flight indoors. One such project was the Cornell Autonomous Flying Vehicle, using a custom vehicle design [74]. Hover control was accomplished using an LQR controller with dead-reckoning estimation, using a human to null the integration error. A second such project was the OS4 quadrotor project, also a custom designed vehicle [8]. The work identifies several dynamic effects beyond the rigid body equations of motion, including gyroscopic torque, angular acceleration of blades, drag force on the vehicle, and rotor blade flapping, though the effects were not further analyzed. Backstepping control was used to improve on the vehicle's initial linear control law, though the linear control law neglected the reference command rate in the derivative term, a design that was found to hurt performance when tested on STARMAC II. Attitude and altitude control and obstacle avoidance were demonstrated experimentally [9]. A third project achieved autonomous hover using nested saturation with IR range measurements of wall positions [24]. The system was modified to incorporate ultrasonic sensors [52], and later incorporated two cameras for state estimation [85]. Finally, the MIT multi-vehicle quadrotor project [97] demonstrated trajectory tracking control using Draganflyer V Ti Pro quadrotors with LQR control and an offboard Vicon position system. The vehicles are capable of tracking slow trajectories throughout an enclosed area visible to the Vicon system.

Autonomous outdoor flight of quadrotor helicopters has been limited. The first platform to demonstrate autonomous outdoor hover capability was STARMAC I, the predecessor to the STARMAC II aircraft presented in this section [40, 100]. The quadrotor, derived from a Draganflyer aircraft, performed GPS waypoint tracking using an inertial measurement unit (IMU), an ultrasonic ranger for altitude, and an L1 GPS receiver. To improve attitude control, this project found that frame stiffening with cross braces between motors greatly improved attitude estimation from the IMU. Aerodynamic disturbances in altitude were also observed with this testbed, modeled using flight data, and compensated for with integral sliding mode control and reinforcement learning [100]. In addition to research platforms, some commercially available small-scale quadrotors are becoming increasingly capable. The MD4-200 quadcopter from Microdrones GmbH [67] can perform waypoint tracking with 2 m

accuracy outdoors using an onboard GPS. By contrast, the results presented in this section demonstrate autonomous path tracking with an indoor accuracy of 0.1 m and an outdoor accuracy of 0.5 m.

Much can be learned from past work on autonomous trajectory tracking for helicopters, a widely studied problem with recent efforts focusing on nonlinear methods. These include techniques such as input/output linearization using differential flatness to track trajectories [54] and backstepping controller design [28] that was used to enable aerobatic maneuvers [31] for an X-Cell 0.60 size helicopter. The backstepping controller has been extended to include robustness considerations as well [62] and in each case, flight test results were demonstrated with outdoor testbed vehicles. These control methods have inspired much of the recent work in quadrotor controller design.

Despite substantial interest in the quadrotor design for autonomous vehicle testbeds, little attention has been paid to the aerodynamic effects that result from multiple rotors, and from motion through the free stream. Exceptions include the Mesicopter project, that studied first order aerodynamic effects [56], and the X-4 Flyer project at the Australian National University [80]. The X-4 project considered the effects of blade flapping, roll and pitch damping due to differing relative ascent rates of opposite rotors, and rotor design. Preliminary results considering these aerodynamic phenomena for vehicle and rotor design showed promise in flight tests [79].

This chapter proceeds by presenting a novel quadrotor helicopter design that is capable of flying both indoors and outdoors with accurate control. It can carry sufficient sensing and computing resources not only to localize and control the aircraft, but also to enable higher levels of vehicular autonomy.

## 2.2 Vehicle Hardware

This section details the vehicle hardware design for STARMAC II. The overall design goals are to make each vehicle a complete, autonomous agent, contributing its own sensing and computing resources to the team of agents. This vehicle design is the basis for the dynamics and control developed in subsequent sections. The vehicle is comprised of: the propulsion system, the sensing system, the control electronics, and



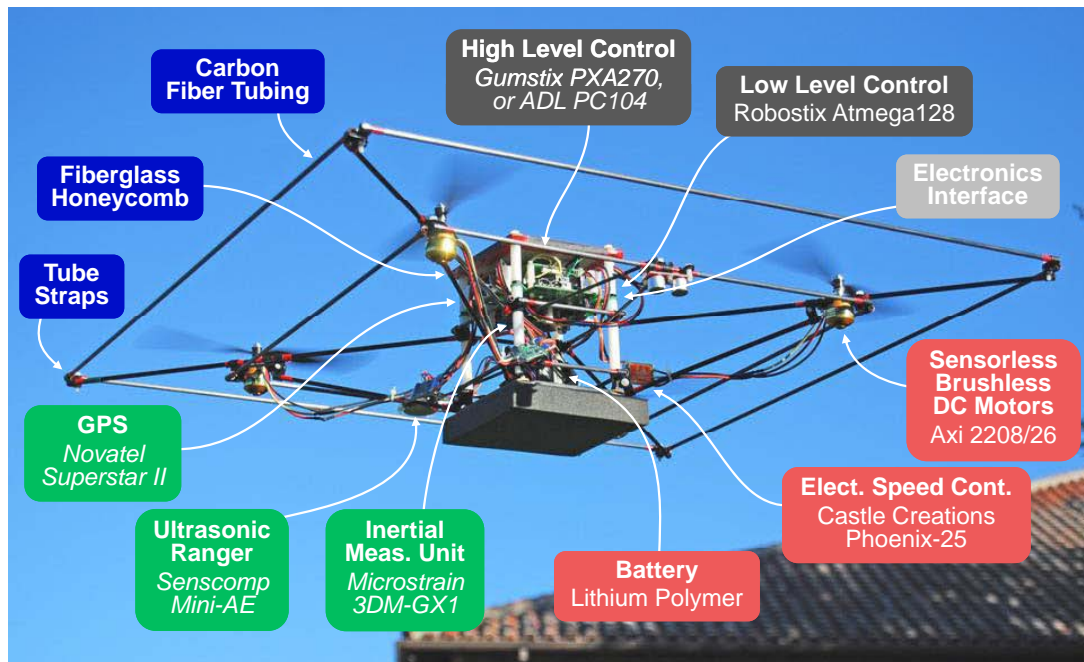


Figure 2.3: STARMAC II vehicle and its components.

the frame, shown in Fig. 2.3 and enumerated in Table 2.1. The design is based on both experimental results and on lessons learned from the first generation vehicle, STARMAC I [40, 100]. The design requirements for STARMAC II are:

1. Accurate control for indoor and outdoor flight.
2. Onboard position and trajectory tracking control.
3. Perception of the environment through onboard sensors.
4. Ability to implement onboard optimization for multi-agent algorithms.

The details of the vehicle components, enabling this autonomous operation, follow.

### 2.2.1 Propulsion

The propulsion system is composed of rotors, motors, speed controllers, and a battery. The components are selected to maximize flight time and payload capacity, while

Table 2.1: STARMAC II Quadrotor Helicopter Components

System	Component	Notes
Propulsion	Electronic Speed Controller	Requires low quantization error
	Sensorless Brushless Motor	High efficiency, direct drive
	Lithium Polymer Battery	High current, high energy density
	Rotor	Designed for low speed flight
Sensing	GPS Receiver	Outputs integrated carrier phase
	Inertial Measurement Unit	Temperature compensated
	Ultrasonic Ranger	Measures proximity of ground
	Battery State	Monitors battery power
Control Electronics	Electronics Interface	Connects all electronics
	Low Level Control Computer	Controls attitude and altitude
	High Level Control Computer	Runs guidance software
Frame	Carbon Fiber Tubing	Primary structure
	Fiberglass Honeycomb	Light protective plates
	Tube Straps	Connections, absorbs shocks

remaining small enough to fly in confined indoor environments. The desired payload capacity was a minimum of 0.4 kg for enhanced onboard computation, with 1.0 kg enabling the inclusion of a suite of environment sensors for autonomous operation in unknown surroundings. As shown in Section 2.3, for a given rotor design, the larger the diameter, the more efficient the operation. However, spatial constraints for indoor flight limited the allowable rotor size. Lithium polymer batteries are used due to their high energy density and discharge rate.

To select an efficient combination of rotors, motors, and speed controllers, preliminary selection uses manufacturer-provided efficiency charts. Rotors designs are selected that are meant for near-hover operation. To more accurately determine the characteristics for various combinations of components, a thrust test stand, shown in Fig. 2.4, measures thrust forces, torques, and efficiencies experimentally. The system is comprised of a desktop computer and Atmel microprocessor that sends waveforms to the motor electronic speed controller. The microprocessor records analog measurements of battery voltage, current and the load cell at 400 Hz, well above the Nyquist frequency of the rotation induced vibrations. This test stand also provides measurements to identify the dynamics of the propulsion system and to quantify the

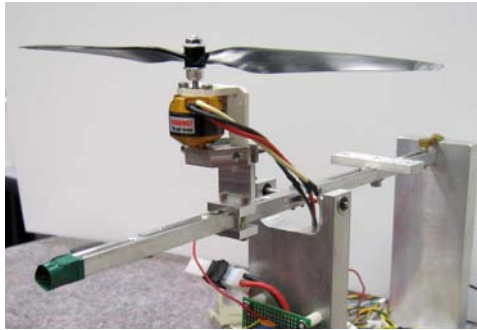


Figure 2.4: Thrust test stand used to measure motor thrust, side force, torque, voltage and current.

aerodynamic effects discussed in Section 2.3. Component combinations are evaluated considering the current draw, voltage drop, total mass of the components, and the total thrust produced. Speed controller deficiencies such as command quantization and output repeatability are also determined for several speed controller options.

The most efficient combination tested at this scale is the Axi 2208/26 sensorless brushless motor [70] with Wattage 10  $\times$  4.5 Park Flyer propellers (tractor and pusher) [101] and the Castle Creations Phoenix-25 controller [13], powered by a Thunder Power [95] 4200 mAH lithium polymer battery. Low quantization noise in this speed controller aids control system implementation, which is sensitive to small differences in relative thrust. The resulting thrust is measured to be up to 8 N per motor for a gross thrust of up to 32 N. With motor current constraints, the vehicle mass is limited to 2.5 kg. This mass limit also ensures sufficient thrust margin for control. The baseline vehicle has a mass of 1.1 kg yielding flight times in the range of 15-20 minutes, though additional batteries could be connected in parallel to extend flight time. A payload of up to 1.4 kg can be included, reducing the total flight time accordingly. The Axi 2208/26 can operate efficiently while directly driving the rotor, eliminating the need for gearing that was one cause of vibration in STARMAC I.

### 2.2.2 Sensors

The sensor suite is comprised of two categories of sensors—those used to estimate the vehicle state and those used to perceive the surrounding environment. The vehicle state sensors provide measurements for attitude, position, and path tracking control algorithms, as described in Section 2.4. The sensors for the surrounding environment provide measurements for automated search and rescue, obstacle detection, and simultaneous localization and mapping.

The vehicle is equipped with three sensors for vehicle state estimation. First, a Microstrain 3DM-GX1 inertial measurement unit (IMU) [68] provides attitude, attitude rate and acceleration through a built-in estimation algorithm using three-axis rate gyroscopes, accelerometers, and magnetometers at 76 Hz. The resulting attitude estimates are accurate to  $\pm 2^\circ$ , though sustained accelerations result in measurement drift. The extended Kalman filter (EKF) therefore uses only the raw sensor measurements, as detailed in Appendix A. Second, a downward facing Senscomp Mini-AE ultrasonic ranger [88] measures the vehicle height above the ground, with a 6 m range, 30 Hz update rate and 0.1% accuracy. The range is measured to the nearest object in a  $30^\circ$  cone, typically the closest point on the ground. Outlier rejection of the raw measurements is required to eliminate false echoes. Third, the position and velocity are estimated in three dimensions using raw integrated carrier phase (ICP) measurements from the Novatel Superstar II GPS receiver [75]. A custom code was developed in house to perform ICP double differencing [69], providing 10 Hz position and velocity estimates. The code is executed in the background in real-time on the onboard computers, saving weight and cost. The accuracy is typically 0.02-0.05 m relative to a stationary base station. For indoor flights, an overhead USB camera is used in conjunction with hue blob tracking software to provide position sensing in place of GPS. The camera system gives 0.01-0.02 m accuracy at 15 Hz, and combined with ultrasonic measurements of the range to the floor, provides a drop-in replacement for GPS input to the EKF.

To perceive the surrounding environment, the vehicle frame can be reconfigured to carry additional sensors for specific applications. Numerous additional sensors are used on the STARMAC platform, including the Videre Systems stereo vision

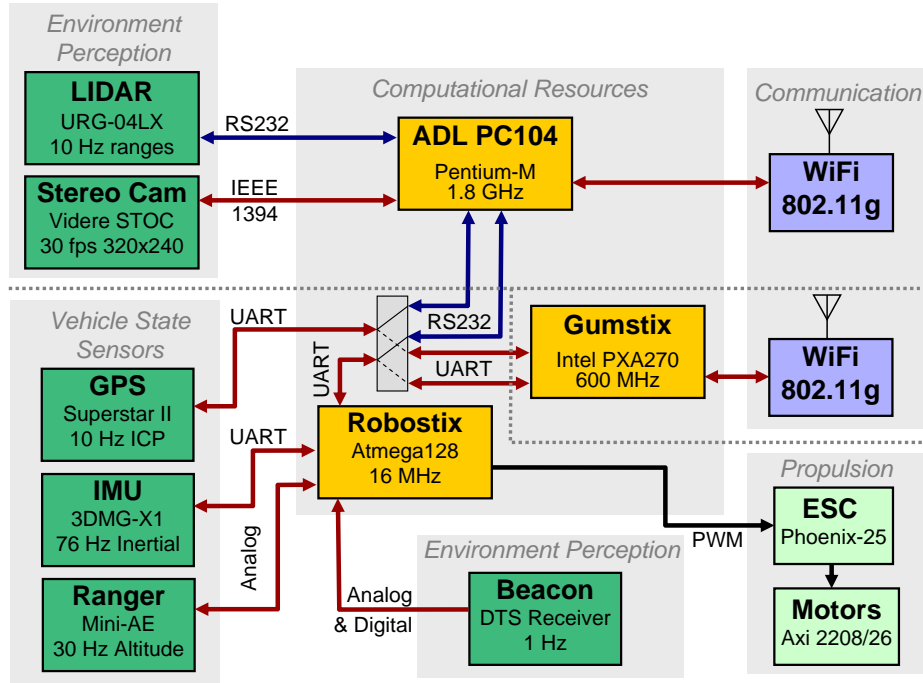


Figure 2.5: Flow of information in the STARMAC II electronics.

camera [98], various USB cameras, the Hokuyo URG-04LX laser range finder [45], and the Tracker DTS digital avalanche rescue beacon and receiver [6]. These sensors enable autonomous multi-agent missions, such as the cooperative search and rescue experiment in Section 5.2.

### 2.2.3 Control Electronics

The control electronics are comprised of an electronics interface board and three computers that manage the distribution of power, the control of the actuators, and the operation of the sensors, as shown in Fig. 2.5.

The electronics interface board is a custom printed circuit board (PCB) that connects many of the electrical components, and provides functionality for their operation. It distributes power from the battery to the motors, converts battery power to levels usable by the digital electronics, and connects to external power while batteries are replaced or the vehicle is grounded. The PCB also shifts serial signal levels,

generates analog signals of battery voltage and current, and provides LEDs to indicate the condition of the embedded computers.

Computation occurs in two stages. Low level processing controls the attitude and altitude dynamics at a fast rate, and the high level processing manages longer computations at a slower rate. The low level computing occurs on a Robostix board [36], using an Atmega128 processor programmed in C to run with no operating system. This system uses measurements from the IMU and ultrasonic ranger to control vehicle attitude and altitude at 76 Hz by sending PWM commands to the motors' electronic speed controllers (ESCs). The low level computer is programmed to be fault tolerant to communication drop-outs. This computer also measures analog and digital inputs from devices such as the rescue beacon receiver and the power monitoring circuitry on the electronics interface board. The low level computer is controlled over the serial UART port by the high level computers.

High level processing occurs on either a Gumstix Verdex single board computer (SBC) [36] running embedded Linux on a PXA270 microprocessor, or on an Advanced Digital Logic PC104 [1] running Fedora Linux; an electric switch selects the computer. The Verdex board is a light, low power device running at 600 MHz with 128 MB of RAM, 64 MB of program space, and 2 GB of external storage on a secure digital card. However, without a floating point unit the computational capabilities remain somewhat limited. The PC104 can be included to provide additional computation power to enable more optimization for more advanced guidance algorithms, at the cost of additional weight and power consumption and hence shortened flight times. The specifications for the PC104's vary; a typical example is a Pentium-M 1.8 GHz processor, 1 GB of RAM, a 16 GB solid-state drive, with USB 2.0 and IEEE 1394. With all peripherals in use, it weighs 0.4 kg and consumes less than 20 W, under 25% the power used by one motor. It enables execution of many optimization algorithms in real-time.

## 2.2.4 Frame

The vehicle frame, shown in Fig. 2.1, is designed to be light, while maintaining sufficient stiffness to ensure accurate state measurement and control actuation. The core of the vehicle, where most electronics are contained, uses fiberglass laminated 0.25 in honeycomb plates to protect the electronics and provide structural rigidity. The core has a  $0.14 \times 0.14$  m horizontal cross section, and can be vertically expanded to accommodate payloads. The electronics interface board is at the center, mounted to the frame. The remainder of the frame is constructed of carbon fiber tubes, Delran motor mounts, nylon fasteners, tube straps, foam padding for landing gear, and standoffs. These raw materials are commercially available, and most are machined using a grinding wheel, saw, and a drill; a mill is only used for motor mount fabrication. The frame is approximately 15% of the total mass of the vehicle.

The tube strap-based frame design allows the vehicle to be easily reconfigured for various experiments and payloads, and acts to absorb the energy; the tubes have some ability to slide in the straps rather than breaking, and the straps themselves can flex. This design eliminates the stress concentrations in earlier designs that used rigid aluminum or plastic channels to clamp the tubes.

During development, the STARMAC II airframe design underwent several iterations using the easily reconfigured plastic joints. Multiple carbon fiber tube crossbraces were experimented with. These minimize vibrational effects on the IMU by suppressing frequencies below the cutoff frequency, and ensure consistent thrust generation by maintaining motor alignment. These crossbraces increase the stiffness of the motor mounts with respect to the core, improving both estimation and control accuracy in flight tests.

In further experiments with the reconfigurable frame, protective shrouds around the rotors are compared to the protective carbon fiber tube exterior shown in Fig. 2.3, a  $0.75 \times 0.75$  m square frame, small enough to fit through most doorways. With protective shrouds in place, using a gap of 5% of the rotor radius, experiments reveal that yaw tracking performance is degraded to approximately  $\pm 10^\circ$  of error, as shown in Fig. 2.6a, whereas with the carbon fiber exterior frame, yaw tracking error is less than  $\pm 3^\circ$ , as shown in Fig. 2.6b. Note that the control system is a different version

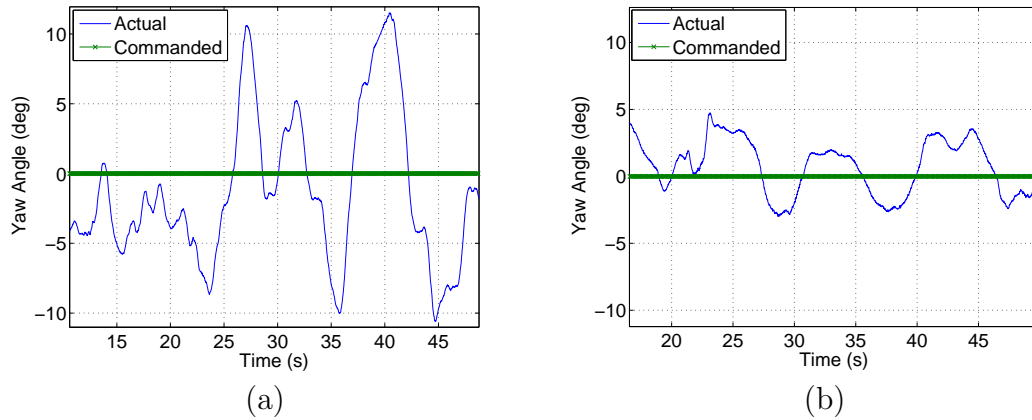


Figure 2.6: Control system performance on prototype of STARMAC II (a) with shrouds around the rotors and (b) with the shrouds removed.

than the ones presented in this dissertation. The data show that during shrouded flights, the angular acceleration of the vehicle about the yaw axis does not consistently match motor commands. One likely explanation is that the shrouds may interact with the flow of air through the rotors, causing a variable reaction torque or disturbance in the airstream.

Similarly, when the rotors are mounted close to the center of the vehicle, disturbances are introduced to the attitude control system. Experiments demonstrate that this effect is reduced when the rotors are moved away from the core of the vehicle, reducing the impingement of the downwash and tip vortices on the structures in the core. To further reduce vibration, the position of diagonal stiffener mounts, beneath the rotor planes, is varied in experiments. Moving the mount toward the blade tip verifies the hypothesis that proximity of any component of structure to the blade tips, the source of tip vortices, yields strong random disturbances, rendering the quadrotor essentially unflyable. Upon moving the mount to coincide with the center of the motor, the disturbance is eliminated. The final configuration has a diagonal spacing between motor centers of  $L = 0.61$  m.



## 2.3 Quadrotor Dynamics

The dynamics of quadrotor helicopters are now presented in two parts. First, thrust and torque generation are described in terms of the inputs to the brushless motors. Then, the nonlinear dynamics used in the development of vehicle controllers in Section 2.4 are presented.

### 2.3.1 Thrust and Torque Generation

Thrust is produced by each rotor through the torque applied by brushless DC motors, with the dynamics of each motor given by

$$Q_m = k_q i_m \quad (2.1)$$

$$V = R_a i_m + k_e \omega_m \quad (2.2)$$

where  $Q_m$  is the torque developed by the motor,  $i_m$  is the current through the motor,  $V$  is the voltage across the motor, and  $\omega_m$  is the angular rate at which the motor is spinning [27].  $k_q$ ,  $R_a$ , and  $k_e$  are motor-specific constants, where  $k_q$  relates current to torque,  $R_a$  is the total armature resistance of the motor, and  $k_e$  relates motor speed to the back EMF. Converting voltage to motor power  $P_m$  in steady state gives

$$P_m = i_m V = \frac{Q_m}{k_q} V \quad (2.3)$$

which can be related to thrust by equating the power produced by the motors to the ideal power required to generate thrust by changing the momentum of a column of air. At hover, the ideal power  $P_h$  is

$$P_h = T_h v_h \quad (2.4)$$

where  $T_h$  is the thrust produced by the motor to remain in hover and  $v_h$ , the induced velocity at hover, is the change in air speed induced by the rotor blades with respect

to the free stream velocity,  $v_\infty$ . Using conservation of momentum and energy [59],

$$v_h = \sqrt{\frac{T_h}{2\rho A}} \quad (2.5)$$

where  $A = \pi R^2$  is the area swept out by the rotor,  $\rho$  is the density of air and  $R$  is the radius of the rotor. Combining (2.4) and (2.5), the required power to remain in hover is

$$P_h = \frac{T_h^{3/2}}{\sqrt{2\rho A}} \quad (2.6)$$

This quantifies the effect of vehicle weight and rotor area on power requirements: the larger the rotor area the less power consumed per unit weight. With  $P \propto T_h^{1.5}$ , and the electrical power density of the aircraft increasing with the mass of batteries carried, the optimal tradeoff for battery payload can be determined, subject to component voltage and current limits.

For a quadrotor helicopter,  $T_h$  is equal to  $\frac{1}{4}T_{nom}$ , where the nominal total thrust  $T_{nom}$  is equal to the weight of the vehicle. The steady state torque is proportional to the thrust [59],  $Q_m = \kappa_t T$ , with a constant ratio  $\kappa_t$  that depends on blade geometry. Angular acceleration of the rotor was found to have a negligible effect on the reaction torque. The relation between applied voltage and thrust is found by equating (2.3) and (2.6), neglecting efficiency losses, yielding

$$T_h = \frac{2\rho A \kappa_t^2}{k_q^2} V^2 \quad (2.7)$$

Thrust is proportional to the square of the voltage across the motor. The thrust from rotor  $j$  is denoted  $T_j$ , and is controlled through the application of a voltage  $V_j$ . In developing control laws for the quadrotor helicopter, (2.7) is linearized about  $T_h$ .

Aerodynamic effects due to motion through the free stream alter these forces and moments. As a rotor moves translationally, the relative momentum of the airstream causes an increase in lift. The angle of attack of the rotor plane with respect to the free-stream also changes the lift, with an increase in angle of attack increasing thrust, similar to airfoils. These combined effects alter the ‘‘induced power’’ of the

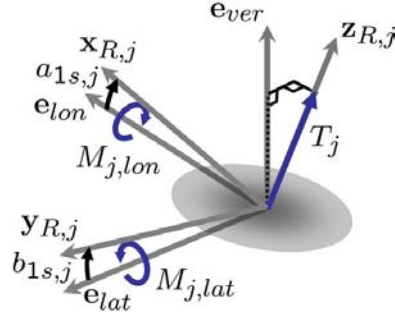


Figure 2.7: Free body diagram of the moments and forces acting on rotor  $j$ . Blade flapping causes the rotor plane to tilt by deflection angles  $a_{1s,j}$  and  $b_{1s,j}$ , and results in moments  $M_{j,lon}$  and  $M_{j,lat}$  about the rotor hub, as developed in Appendix B.

rotor. A further effect is “blade flapping”, an imbalance in lift caused by different relative velocities between advancing and retreating blades and the free stream, results in a steady state tilt of the rotor plane, deflecting the thrust vector and causing a moment at stiff rotor hubs, shown in Fig. 2.7. These aerodynamic effects are treated as disturbances in the control system. Details on the effects are developed in Appendix B using techniques from helicopter analysis, thrust test stand data, and flight experiments.

### 2.3.2 Inertial Dynamics

The nonlinear dynamics are derived in North-East-Down (NED) inertial and body fixed coordinates. Let  $\{\mathbf{e}_N, \mathbf{e}_E, \mathbf{e}_D\}$  denote unit vectors along the respective inertial axes, and  $\{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$  denote unit vectors along the respective body axes, as shown in Fig. 2.8. Euler angles to rotate from NED axes to body fixed axes are the 3-2-1 sequence  $\{\psi, \theta, \phi\}$ , referred to as yaw, pitch, and roll, respectively. The current velocity direction unit vector is  $\mathbf{e}_v$ , in inertial coordinates. The direction of the projection of  $\mathbf{e}_v$  onto the  $\mathbf{x}_B - \mathbf{y}_B$  plane defines the direction of  $\mathbf{e}_{lon}$  in the body-fixed longitudinal, lateral, vertical frame,  $\{\mathbf{e}_{lon}, \mathbf{e}_{lat}, \mathbf{e}_{ver}\}$  in Fig. 2.7. Due to blade flapping, the rotor plane does not necessarily align with the  $\mathbf{x}_B, \mathbf{y}_B$  plane, so for the  $j^{th}$  rotor let  $\{\mathbf{x}_{R,j}, \mathbf{y}_{R,j}, \mathbf{z}_{R,j}\}$  denote unit vectors aligned with the plane of the rotor and oriented with respect to the  $\{\mathbf{e}_{lon}, \mathbf{e}_{lat}, \mathbf{e}_{ver}\}$  frame. Let  $\mathbf{x}$  be defined as the position vector

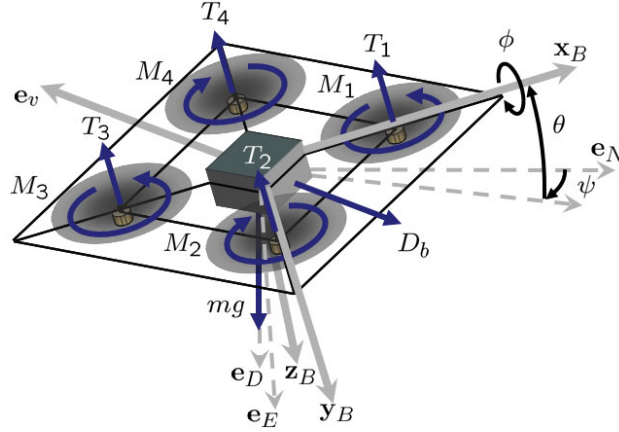


Figure 2.8: Free body diagram of a quadrotor helicopter.

from the inertial origin to the vehicle center of gravity (c.g.), and let  $\omega_B$  be defined as the angular velocity of the aircraft in the body frame.

The rotors, numbered 1 – 4, are mounted outboard on the  $\mathbf{x}_B$ ,  $\mathbf{y}_B$ ,  $-\mathbf{x}_B$  and  $-\mathbf{y}_B$  axes, respectively, with position vectors  $\mathbf{r}_j$  with respect to the c.g.. The thrust produced by the  $i^{th}$  rotor acts perpendicularly to the rotor plane along the  $\mathbf{z}_{R_j}$  axis, as defined in Fig. 2.7. The vehicle mass is  $m$ , acceleration due to gravity is  $g$ , the inertia matrix is  $I_B \in \mathbb{R}^{3 \times 3}$ , and vehicle body drag force is  $D_b = \kappa_d v_\infty^2$ . Drag constant  $\kappa_d$  varies with vehicle state. At an angle of attack of  $15^\circ$ , it is measured to be approximately  $0.02 \frac{N}{(m/s)^2}$  for STARMAC II.

The total force  $\mathbf{F}$  is,

$$\mathbf{F} = -D_b \mathbf{e}_v + mg \mathbf{e}_D + \sum_{j=1}^4 (-T_j R_{R_j, I} \mathbf{z}_{R_j}) \quad (2.8)$$

where  $R_{R_j, I}$  is the rotation matrix from the plane of rotor  $j$  to inertial coordinates. This rotation matrix includes an additional rotation for the aerodynamic effects derived in Appendix B. Similarly, the total moment  $\mathbf{M}$  is,

$$\mathbf{M} = \sum_{j=1}^4 (\mathbf{M}_j + \mathbf{M}_{bf, j} + \mathbf{r}_j \times (-T_j R_{R_j, B} \mathbf{z}_{R_j})) \quad (2.9)$$

where  $R_{R_j,B}$  is the rotation matrix from the plane of rotor  $j$  to body coordinates. This rotation matrix is due to the aerodynamic effects derived in Appendix B. Note that the drag is neglected in computing the moment. This force was found to cause a negligible disturbance on the total moment over the flight regime of interest, relative to blade flapping torques. The nonlinear dynamics are,

$$\mathbf{F} = m\ddot{\mathbf{x}} \quad (2.10)$$

$$\mathbf{M} = I_B\dot{\omega}_B + \omega_B \times I_B\omega_B \quad (2.11)$$

where the total angular momentum of the rotors is assumed to be near zero, as the momentum from the counter-rotating pairs cancels when yaw is held steady.

## 2.4 Quadrotor Control

The vehicle control system for STARMAC II uses successive loop closure. The inner loop controls attitude and altitude and executes at a higher rate than the outer loop to control the fast attitude dynamics of the vehicle. Altitude control is also performed as part of the inner loop to exploit accelerometer data, available at a high rate, for disturbance rejection. The outer loop has several modes, including hovering at a waypoint, trajectory tracking, and feeding through attitude and altitude commands from autonomous guidance algorithms. The control system design enables improved control accuracy over the state-of-the-art in the literature. This improved accuracy directly improves the performance of the autonomous guidance system. The details of the inner loop and outer loop control systems are given in the following presentation of the control system.

### 2.4.1 Attitude and Altitude Control

Attitude and altitude control are the inner loop of the vehicle control system. The equations of motion for attitude, (2.9) and (2.11), approximately decouple about each attitude axis, so control input moments about each axis,  $u_\phi$ ,  $u_\theta$ , and  $u_\psi$ , can be implemented independently. This is a good approximation when the angular velocities

are low and the roll and pitch angles are within approximately  $\pm 30^\circ$ . The inputs for each axis are added to the total thrust control input  $u_z$  to generate thrust commands  $u_1$  through  $u_4$ , for motors 1 through 4,

$$\begin{aligned} u_1 &= -u_\theta/L + u_\psi/(4\kappa_t) + u_z/4 \\ u_2 &= u_\phi/L - u_\psi/(4\kappa_t) + u_z/4 \\ u_3 &= u_\theta/L + u_\psi/(4\kappa_t) + u_z/4 \\ u_4 &= -u_\phi/L - u_\psi/(4\kappa_t) + u_z/4 \end{aligned} \quad (2.12)$$

Note that vertical acceleration in the body frame due to  $u_z$  is decoupled from the attitude loop by design of the quadrotor, though when the vehicle rotates, this acceleration is no longer in the inertial vertical direction. In controlling each attitude angle, the time delay in thrust must be included in the model. It is well approximated as a first order delay with time constant  $\tau$ , as experimentally verified [41], and found to be 0.1 s for STARMAC II. The resulting transfer function for the roll axis is

$$\frac{\Phi(s)}{U_\phi(s)} = \frac{I_\phi}{s^2(\tau s + 1)} \quad (2.13)$$

where  $I_\phi$  is the component of  $I_B$  for the roll axis. The transfer functions for the pitch and yaw axes are analogous. Note that the induced power and blade flapping effects are not included in the linear model; they are treated as disturbance forces and moments that must be rejected by the control system. They are directly compensated for in [47], though note that unknown wind disturbances, potentially variable in speed and direction, must still be rejected.

Although a standard PID controller has been shown to perform adequately [100], control design using root locus techniques reveals that adding an additional zero, using angular acceleration feedback, allows the gains to be increased by an order of magnitude, yielding higher bandwidth. Further, by using acceleration compensation, there is direct feedback on the actual thrust achieved, regardless of vortex ring state or ascent/descent dynamics given in Appendix B. The resulting control law,

$$C(s) = k_{dd}s^2 + k_d s + k_p + \frac{k_i}{s} \quad (2.14)$$

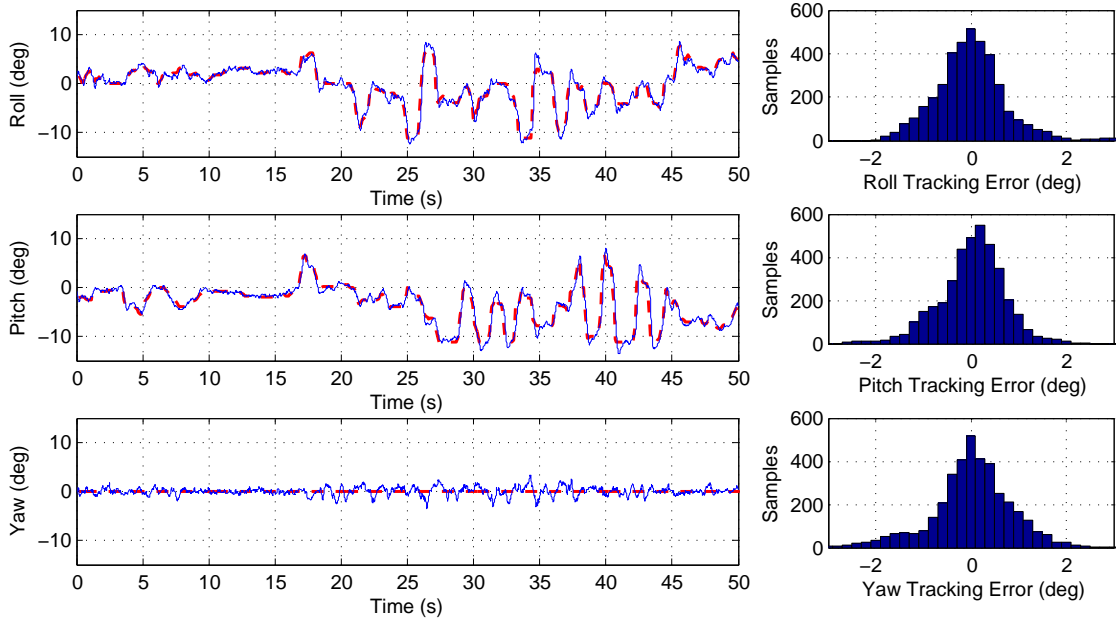


Figure 2.9: Attitude control results from indoor flight tests for roll, pitch and yaw. Histograms of tracking error on the right.

is tuned to provide substantially faster and more accurate performance than previously possible. The time-domain control input for the roll axis is then

$$u_{\phi} = k_{dd}(\ddot{\phi}_{ref} - \ddot{\phi}) + k_d(\dot{\phi}_{ref} - \dot{\phi}) + k_p(\phi_{ref} - \phi) + k_i \int_0^t (\phi_{ref} - \phi) dt \quad (2.15)$$

with the time-domain angular control inputs  $u_{\theta}$  and  $u_{\psi}$  generated accordingly. Note that anti-windup is used for the integral term.

The implementation of this control law requires some consideration. First, the angular acceleration signal must be computed by finite differencing the rate gyroscope data, a step that can amplify noise. However, in implementation, the signal resulting from differencing the current and previous angular velocity measurements at 76 Hz has sufficiently low noise for use in the controller. Second, to implement this control law,  $C(s)$  acts on the error signal to provide *reference command tracking* capability, unlike previous work [8]. To enable this, the reference command is processed by a low pass digital filter to compute first and second derivatives.

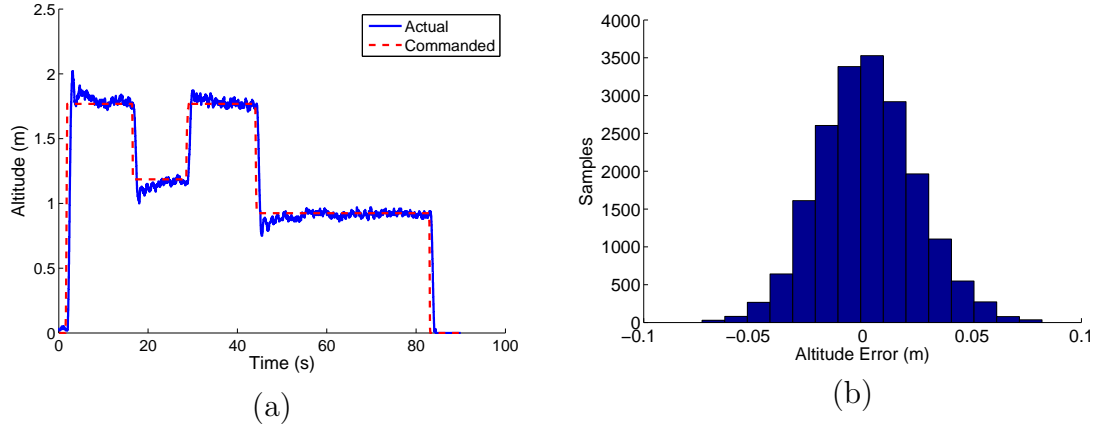


Figure 2.10: (a) Altitude command tracking in indoor flight tests. (b) Histogram of altitude error for 3 minute hover flight.

In practice, the controller can track rapidly varying reference commands, as shown in Fig. 2.9, with root mean square (RMS) error of  $0.65^\circ$  in each axis. Aggressive flights have been flown frequently, with typically up to  $15^\circ$  of bank angle. The controller has been flown up to its programmed limit of  $30^\circ$  without apparent degradation in performance. The tracking performance of this control system is the most accurate system in the quadrotor literature to the knowledge of the author.

A similar approach is used for altitude control, improving performance over a PID implementation. An accelerometer directly observes the vertical acceleration. Vehicle vibration is removed from the signal by a digital low pass filter. The utility of this noisy measurement was discovered while experimenting with various filtered signals in [100]. Feedback linearization compensates for the offset of gravity and the deflection of thrust due to tilt,

$$u_z = \frac{1}{\cos \phi \cos \theta} (k_{dd,alt}(\ddot{z}_{ref} - \ddot{z}) + k_{d,alt}(\dot{z}_{ref} - \dot{z}) + k_{p,alt}(z_{ref} - z)) + T_{nom} \quad (2.16)$$

where  $z$  is the altitude and  $z_{ref}$  is the reference command. The linearized plant model is identical in form to (2.13).

Results for the altitude control loop are presented in Fig. 2.10. RMS error in position was measured to be 0.021 m in hover flight at a constant altitude.



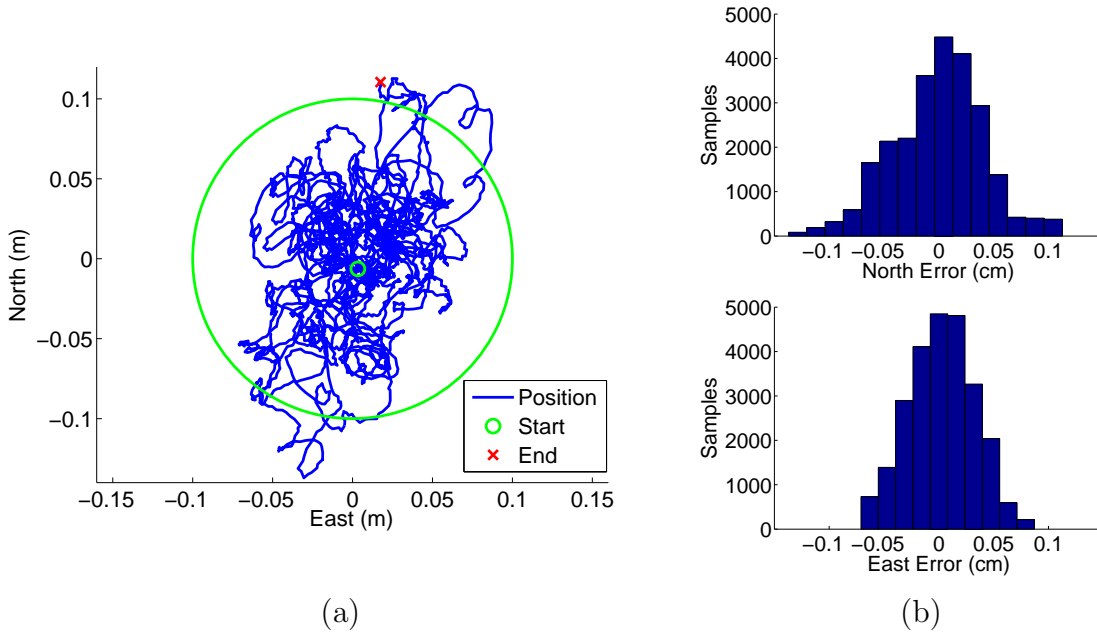


Figure 2.11: Autonomous indoor hover performance. (a) Position plot over 3 minutes of flight with 0.1 m error circle. (b) Histogram of position error in North and East directions.

## 2.4.2 Position and Trajectory Tracking Control

This outer loop control system operates in three modes, all of which send reference commands to the attitude control loop. In the first mode, position tracking, a way-point position is tracked. In the second mode, trajectory tracking, a sequence of waypoints are tracked. In the third mode, feed-through, commands from either the onboard autonomous guidance system or a human at the ground station are passed through. This section presents details of the position and trajectory tracking modes.

Consider position control in the  $\mathbf{e}_E$  direction. Using (2.11) and (2.8), when  $\psi = 0$ , the transfer function is  $\frac{X_E(s)}{\Phi(s)} = (T_{nom}/m)\frac{1}{s^2}$ . Note that this neglects drag, induced power, and blade flapping. For this control design, those effects are again treated as disturbance forces that must be compensated by the control system. Setting the control input to  $u_E = \phi_{ref}$ , the open loop plant is the convolution of the  $x_E$  dynamics

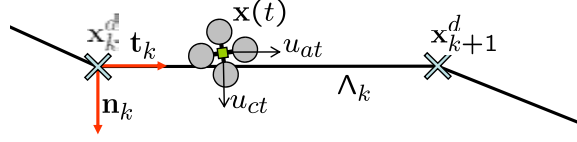


Figure 2.12: The quadrotor travels along path segment  $\Lambda_k$  from waypoint  $\mathbf{x}_k^d$  to  $\mathbf{x}_{k+1}^d$ , applying along and cross-track control inputs to track the path.

with the closed-loop dynamics of  $\phi$ , (2.13), using the feedback control law of (2.14),

$$\frac{X_E(s)}{U_E(s)} = \frac{C(s)T_{nom}I_\phi/m}{s^2(\tau s^3 + s^2 + C(s)I_\phi)} \quad (2.17)$$

The open loop plant for the  $\mathbf{e}_N$  direction is analogous, using  $\theta$  rather than  $\phi$ . When  $\psi \neq 0$ , the control inputs must be rotated accordingly.

A PID controller is implemented using (2.17). The results for indoor flight tests are presented in Fig. 2.11. The resulting RMS east and north error is 0.036 m. Disturbances are introduced by recirculating downwash in the small flight test room. By using  $C(s)$  rather than previous PID implementations for attitude control, the aerodynamic effects are better rejected than in previous work on STARMAC II, yielding superior closed-loop position control. Aerodynamic effects are additionally rejected by the integral term in the position control loop.

To track trajectories, a path  $\Lambda \in N_p \times \mathbb{R}^3$  is defined by a sequence of  $N_p$  desired waypoints,  $\mathbf{x}_k^d$  and desired speeds of travel  $v_k^d$  along path segment  $\Lambda_k$  connecting waypoint  $k$  to  $k+1$ , as depicted in Fig. 2.12. Let  $\mathbf{t}_k$  be the unit tangent vector in the direction of travel along the track from  $\mathbf{x}_k^d$  to  $\mathbf{x}_{k+1}^d$ , and  $\mathbf{n}_k$  be the unit normal vector to the track. Then, given the current position  $\mathbf{x}(t)$ , the cross-track error  $e_{ct}$ , error rate  $\dot{e}_{ct}$  and along-track error rate  $\dot{e}_{at}$  are,

$$\begin{aligned} e_{ct} &= (\mathbf{x}_k^d - \mathbf{x}(t)) \cdot \mathbf{n}_k \\ \dot{e}_{ct} &= -\mathbf{v}(t) \cdot \mathbf{n}_k \\ \dot{e}_{at} &= v_k^d - \mathbf{v}(t) \cdot \mathbf{t}_k \end{aligned} \quad (2.18)$$

Note that only the along-track error rate is considered, and depends only on the velocity of the vehicle. This is done so that the controller does not attempt to catch

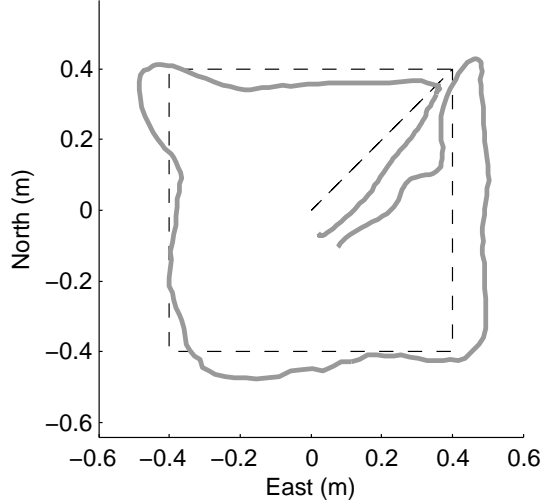


Figure 2.13: Tracking a trajectory indoors, at 0.5 m/s, with an error of under 0.1 m.

up or slow down to meet a schedule by deviating from the desired path. This design choice assumes that the desired speed is selected, and the time of achieving a waypoint is unimportant. It is straightforward to extend the following control law by including feedback on along-track position, if timing is important.

The trajectory tracking controller is implemented by closing the loop on along-track rate error and cross-track error [44]. This is essentially piecewise PI control in the along-track direction, and PID control in the cross-track direction,

$$\begin{aligned}
 u_{at} &= K_{d,at}\dot{e}_{at} + K_{i,at}\int_0^t \dot{e}_{at}dt \\
 u_{ct} &= K_{p,ct}e_{ct} + K_{d,ct}\dot{e}_{ct} + K_{i,ct}\int_0^t \dot{e}_{ct}dt
 \end{aligned} \tag{2.19}$$

where control inputs  $u_{at}$  and  $u_{ct}$  are the attitude commands for vehicle tilt in the along-track and cross-track directions, respectively. They are rotated by  $\psi$  and by the trajectory orientation to generate  $\phi_{ref}$  and  $\theta_{ref}$  commands for the inner loop. Transition from segment  $k$  to  $k + 1$  occurs when the vehicle crosses the line segment normal to the path at the end of the segment. The trajectory controller presented here is intended for use with a coarse set of waypoints, focusing on accurate line

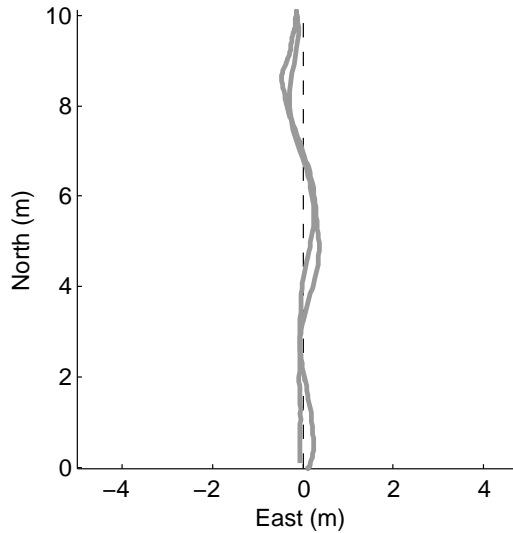


Figure 2.14: Tracking a trajectory outdoors, at 2.0 m/s, with an error of under 0.5 m.

tracking. It has been improved upon for finer resolution paths by computing feed forward inputs to follow a least-norm control input solution through the waypoints.

The controller defined in (2.19) is implemented on STARMAC II. Results for both indoor and outdoor settings are presented in Figs. 2.13 and 2.14, respectively [44]. The indoor results demonstrate tracking errors of under  $\pm 0.1$  m throughout the box shaped trajectory, and show the largest overshoot when switching from one track to the next, as the desired direction of travel suddenly switches by  $90^\circ$ . For the outdoor flight tests, the gains on the cross-track and along-track controllers are reduced by a factor of two, and the resulting errors increases to  $\pm 0.5$  m. Lower gains are used to prevent oscillations in the control system not experienced inside. The oscillations might be attributed to wind disturbances or to decreasing the position update rate from 15 Hz for indoor positioning to 10 Hz for GPS.

This trajectory controller has some interesting properties. Since the path is composed only of line segments, overshoot on sharp corners is inevitable. This is addressed by upsampling using least-norm feed forward inputs, as mentioned above, or alternatively by a constraint-meeting algorithm given in [44]. Another property arises due to the lack of mandated times to be at each waypoint. The trajectory tracking controller



Figure 2.15: Three STARMAC quadrotor helicopters performing position control at GPS waypoints.

does not deviate from the intended path to try to meet a deadline—a required feature when the path is meant to avoid obstacles.

## 2.5 Summary

This chapter described the vehicle design, dynamics and control system of the STARMAC testbed. This system is capable of accepting commands from the autonomous guidance system and following them reliably, removing much of the burden from potentially complex optimizations. Updates from the guidance system may occur at a slower rate than required to control vehicle dynamics, and the control system mitigates disturbances.

Now, the question becomes how to control a fleet of vehicles, such as STARMAC, as a cohesive mobile sensor network, to optimally acquire information. As mobile sensors are added to this fleet, it becomes challenging to beneficially incorporate the additional resources into the network, such as the network of vehicles shown in Fig. 2.15. To accomplish this, the task of information collection is analyzed in

information-theoretic terms in Chapter 3, and algorithms for guidance are designed to enable scalable implementation of the resulting objective function. In Chapter 4, a collision avoidance algorithm is derived to enable the use of the mobile sensor network guidance algorithms. They are demonstrated in simulation of quadrotor helicopters and experimented with on STARMAC II aircraft. In Chapter 5, these autonomous guidance algorithms are implemented in simulation and then on STARMAC II aircraft.

## Chapter 3

# Mobile Sensor Guidance

To develop autonomous guidance algorithms for mobile sensors, the strong coupling between estimation and control systems, introduced in Section 1.2, can be modeled and exploited. The main objective of a mobile sensor is to gather information. To accomplish this goal with any degree of optimality, the guidance algorithm must consider the effect of the control inputs on the quality of the information acquired.

This chapter proposes three techniques to address the goal of seeking information in order to yield a mobile sensor network framework which is scalable and capable of accurately capturing and using information. The first technique is to directly use particle filter estimators [33] to compute an information-seeking objective function. This enables the use of multi-modal posterior distributions, nonlinear and non-Gaussian sensor models, and the use of general prior information. This technique preserves details in the objective function that would be discarded by linearization and Gaussian approximations—by directly using a particle filter representation, it is possible to more accurately quantify the value of potential observations. The second technique is to decompose the information-seeking objective function so that as the number of vehicles increases, the vehicles can leverage one another’s positions to improve the sensing capabilities, by using approximations that discard higher order terms. A first approximation is to fully decouple the problem. We define this as the *single-node* approximation, and derive the error incurred. Its computational complexity is constant with respect to the number of sensors, yielding a fast distributed cooperative

optimization. Although the vehicles appear to cooperate due to optimization using the same target state probability distribution, the only interactions between their local optimization problems are the collision avoidance constraints. In order to enable a higher level of cooperative sensing, a new method is proposed that considers the effects of each sensor on each other sensor, pairwise, called the *pairwise-node* approximation. It incurs only a linear computational expense in the number of vehicles, and the effect of the approximation error is provably reduced from that of the single-node approximation, allowing coupled effects between the mobile sensors to be directly captured.

The mobile sensing problem considered for this chapter focuses on using mobile sensors to search for a target. The particle filter is formulated for this problem, though note that the techniques developed for particle filters are applicable to any particle filter formulation. Also note that the information-theoretic approximations are more generally applicable to any probabilistic representation.

The chapter proceeds by defining the structure of the search problem, and then proposing an information-theoretic framework for the solution. First, the mobile sensor network model is defined. Next, the goal of searching for a target is cast using the information-theoretic concept of mutual information as a utility function. This quantity can be maximized in expectation to optimally search for a target.

### 3.1 System Definition

Consider a set of  $n_v$  vehicles carrying sensors to locate a target in the search domain  $\Theta$ . The state of the  $i^{th}$  vehicle is  $\mathbf{x}_t^{(i)} \in \mathbb{R}^{n_s}$ , with  $n_s$  vehicle states, such as position, orientation, and velocity. The location of the target  $\theta_t \in \Theta \subset \mathbb{R}^{n_\theta}$  at discrete time  $t$  is unknown to the vehicles. Though stationary targets are considered here, a motion model could be used for nonstationary targets [37]. A prior distribution  $p(\theta_0)$  is provided, using any information available *a priori*.

Sensor measurements for the  $i^{th}$  vehicle  $\mathbf{z}_t^{(i)} \in Z^{(i)} \subset \mathbb{R}^{n_z}$  are taken at rate  $\frac{1}{\Delta}$ ,



where  $Z^{(i)}$  is the domain of the observations, with dimension  $n_z$ . Where the superscript is omitted,  $\mathbf{z}_t = \{\mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(n_v)}\}$ . The measurement model is

$$\mathbf{z}_t^{(i)} = h_t^{(i)}(\mathbf{x}_t^{(i)}, \theta, \eta_t^{(i)}) \quad (3.1)$$

The observation noise is  $\eta_t^{(i)} \in \mathbb{R}^{n_\eta}$  with an assumed probability distribution  $p(\eta_t^{(i)})$ . The noise distribution need not be Gaussian. The problem formulation admits a broad class of measurement models, as  $h_t^{(i)}$  could be a nonlinear or discontinuous mapping of the states and measurement noise onto the observation space. Each vehicle is provided, *a priori*, with the measurement model for all sensors in the network. This enables each vehicle to locally interpret all observations made by the sensor network, and provides each vehicle with a model for the sensing capabilities of all other sensors, making possible optimal trajectory planning, as described in the following section. Note that the measurement models for any sensor in the network must be known in order to use that sensor, so providing the measurement models of each sensor to each vehicle introduces no practical limitations in this cooperative scenario.

The motion models for each vehicle, its discrete time dynamics, are

$$\mathbf{x}_{t+1}^{(i)} = f_t^{(i)}(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)}) \quad (3.2)$$

where  $\mathbf{u}_t^{(i)} \in U^{(i)} \subset \mathbb{R}^{n_u}$  is the set of  $n_u$  control inputs,  $U^{(i)}$  is their domain, and the time duration between time steps is  $\Delta$ . A collision avoidance constraint is imposed between vehicles: vehicles are required to maintain minimum separation distance of  $d_{min}$ . This accounts for the finite expanse of the vehicles, as well as any desired safety margin. Let  $\rho_t^{(i)}$  be the subset of the  $i^{th}$  vehicle's states that corresponds to its position. The collision avoidance constraint can be written as

$$\|\rho_t^{(i)} - \rho_t^{(j)}\| \geq d_{min} \quad \forall j \in \{1, \dots, n_v : j \neq i\} \quad (3.3)$$

For computational purposes, each vehicle must store the posterior distribution

locally. To enable distributed knowledge of the posterior distribution, using non-parametric estimators, each vehicle  $i$  maintains its own instantiation of the posterior distribution of the target state  $p(\theta_t)$  incorporating all prior measurements,  $\{\mathbf{z}_1, \dots, \mathbf{z}_{t-1}\}$ . Note that the local instantiation of  $p(\theta_t)$  is a non-parametric approximation to the true posterior distribution, which is a continuous function. The system is assumed to be Markov, hence recursive updates using Bayes' rule are used. To incorporate new observations made or received between times  $t - 1$  and  $t$ ,

$$p(\theta_t|\mathbf{z}_t) = \frac{p(\theta_t)p(\mathbf{z}_t|\theta_t)}{p(\mathbf{z}_t)} \quad (3.4)$$

The target is assumed stationary for this work, so recursion is accomplished using  $p(\theta_t) = p(\theta_{t-1}|\mathbf{z}_{t-1})$ . To incorporate a nonstationary target, a motion model would provide this relationship. The posterior distribution is stored locally at each vehicle. All vehicles' distributions are based on the full history of shared observations, and can be assumed nearly identical. It is assumed that the vehicles are equipped with communication devices that enable this exchange of measurements between vehicles. One such reliable technology is demonstrated in the 802.11g network used for STARMAC [41].

## 3.2 Objective

The goal of the multi-vehicle search team differs from a typical control system. The direct measurement of success is not the ability to track a trajectory. Rather, as shown in Fig. 1.3, it is to maximize the likelihood of localizing the target as quickly as possible. The target is localized by making observations at a fixed rate. The more observations required, the slower the target is localized. Therefore, the goal can equivalently be stated as *controlling sensor locations to minimize the expected number of future observations needed to ascertain the target's state*. A set of observations can be interpreted, in an information theoretic sense, as a code word, with an alphabet comprised of all possible quantized outputs of the  $n_z$  sensors. These encode the continuous target state, which is represented numerically in software by an alphabet

of a finite number of symbols, such as 64 bits in a double precision floating point data type. Therefore, to minimize the expected number of remaining observations is to maximize the expected log-likelihood of the posterior distribution with each observation of the vehicles, as derived in [89], and summarized in Appendix C.

In order to increase this likelihood as quickly as possible at each time step, only the control actions for the current time step need be considered. However, more generally, if one considers optimizing control actions over longer time horizons, it is equivalent to using a larger code word, in information theoretic terms. A longer optimization horizon results in equal or better expected performance by the end of the time horizon, with some interesting bounds given in the literature [102]. However, the one step time horizon maximizes the current rate at which information is acquired, yielding equal or better expected results by the next time step. In accordance with the goal of acquiring information as quickly as possible for the time-critical search problem, which has diminishing returns for delayed information, one step horizons will be considered for this work.

Taking the log-likelihood of the posterior distribution given by Bayes' rule in (3.4), and using the relationship  $p(\mathbf{z}_t, \theta_t) = p(\mathbf{z}_t|\theta_t)p(\theta_t)$  yields

$$H(\theta_t|\mathbf{z}_t) = H(\theta_t) - I(\mathbf{z}_t; \theta_t) \quad (3.5)$$

where

$$H(\theta_t) = - \int_{\theta_t \in \Theta} p(\theta_t) \log p(\theta_t) d\theta_t \quad (3.6)$$

$$H(\theta_t|\mathbf{z}_t) = - \int_{\substack{\theta_t \in \Theta \\ \mathbf{z}_t \in Z}} p(\theta_t, \mathbf{z}_t) \log p(\theta_t|\mathbf{z}_t) d\theta_t d\mathbf{z}_t \quad (3.7)$$

$$I(\mathbf{z}_t; \theta_t) = \int_{\substack{\theta_t \in \Theta \\ \mathbf{z}_t \in Z}} p(\theta_t, \mathbf{z}_t) \log \frac{p(\theta_t, \mathbf{z}_t)}{p(\theta_t)p(\mathbf{z}_t)} d\theta_t d\mathbf{z}_t \quad (3.8)$$

$H(\theta_t)$  is the *entropy* of the target state distribution,  $I(\mathbf{z}_t; \theta_t)$  is the *mutual information* between the distributions of the target state and the sensors, and  $H(\theta_t|\mathbf{z}_t)$  is the *conditional entropy* of the distribution—the expected entropy of the target state when conditioning with  $\mathbf{z}_t$  (see Remark 3.1) [17]. The entropy of a probability distribution is

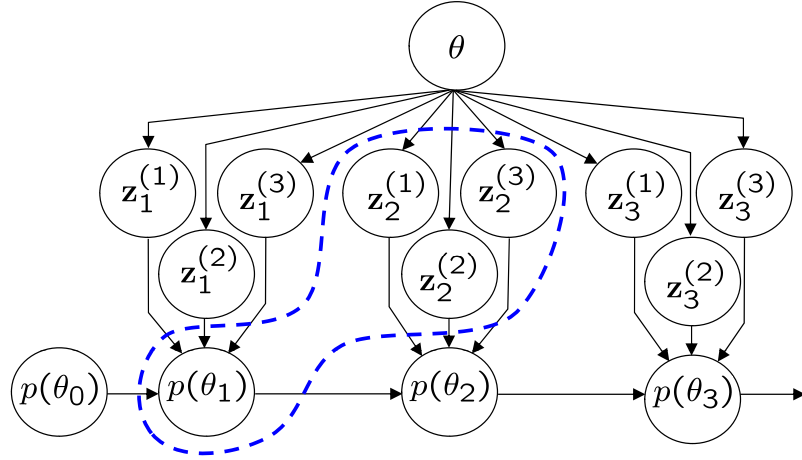


Figure 3.1: Directed graphical model of the estimator, with 3 mobile sensors. The region enclosed in the dashed line contains the random variables used in the information theoretic optimization to compute the control actions at time  $t = 1$ . The optimization selects control actions that optimally reduce the expected uncertainty of the subsequent target state model  $p(\theta_2)$ . The expected uncertainty is a direct function of the probability distributions of the random variables  $\mathbf{z}_2^{(1)}$ ,  $\mathbf{z}_2^{(2)}$ , and  $\mathbf{z}_2^{(3)}$ , and the current target state model  $p(\theta_1)$ . Although  $p(\theta_1)$  is independent of future states of the sensors, the sensor probability distributions are direct functions of the future states of the sensors, states that are selected via the control inputs chosen by the information theoretic optimization.

a metric of the uncertainty of that distribution. The mutual information the expected divergence (Kullback-Liebler) between the independent and joint distributions of  $\theta_t$  and  $\mathbf{z}_t$ . It is large when two distributions have strong interdependence, and zero when they are independent.

**Remark 3.1.** When the argument of  $H(\cdot)$  indicates a conditional relationship, e.g.  $a|b$ , it is the expected entropy of the conditional probability distribution  $p(a|b)$ , since  $b$  is a random variable. Thus,  $H(a|b) = \int_{b \in \mathcal{B}} p(b) \left( \int_{a \in \mathcal{A}} p(a|b) \log p(a|b) da \right) db$ . Because  $p(a, b) = p(a|b)p(b)$ , this is equal to the expression in (3.7). If the value of  $b$  were known to be some constant  $b_c$ , then the entropy would be written  $H(a|b = b_c)$ , which can be computed without taking an expectation.

The control inputs  $\mathbf{u}_t$  and vehicle states  $\mathbf{x}_t$  influence the observations  $\mathbf{z}_t$  through (3.2) and (3.1). To minimize the expected future uncertainty of the target state

distribution, with respect to  $\mathbf{u}_t$ , one minimizes (3.5). Note that the actual uncertainty can only be determined once the true measurement  $\mathbf{z}_t$  is made. The prior uncertainty is independent of the future control inputs, as depicted in Fig. 3.1, so to minimize the expected posterior uncertainty, one must maximize the observation information with respect to the control inputs.

In order to seek information, the network computes its control inputs by maximizing the mutual information utility function, defined as follows.

**Definition 3.1** (Mutual Information Utility Function). *The mutual information utility function is*

$$V^{(i)}(\mathbf{x}_t, \mathbf{u}_t, p(\theta_t)) = I(\mathbf{z}_{t+1}; \theta_{t+1}) \quad (3.9)$$

where the argument  $p(\theta_t)$  indicates that the data defining the probability distribution of  $\theta_t$  are used by the utility function. The arguments on the right hand side of (3.9) are random variables; to evaluate this expression requires the sensor model, (3.1), and vehicle motion model, (3.2). Both are functions of the arguments of the mutual information utility function.

Methods for computing this utility function are presented next.

### 3.3 Information Theory with Particle Filters

The goal of the search problem is to minimize the uncertainty encompassed in the posterior distribution of the target state, as represented here by a particle filter. Regardless of the specific implementation of the particle filter, the method proposed in this work focuses on controlling the vehicles such that they maneuver the sensors to make observations that reduce the uncertainty in the particle set as quickly as possible, distinguishing the likely particles from the unlikely particles.

This section proceeds by first giving a review of particle filters, as implemented in this work. Then, a method is developed to compute mutual information directly from the particle filter representation of the target state probability distribution. In order to improve the efficiency of computing mutual information as the size of the

sensor network grows, two decompositions leading to two approximations are derived next. These approximations, with analytically quantified error, permit a direct tradeoff between computational complexity and the level of cooperation between vehicles. Finally, the distributed control algorithm applied to these utility functions is presented.

### 3.3.1 Particle Filter

Particle filters are a Monte Carlo method to perform Bayesian estimation. By using this Monte Carlo method, it is possible to make direct use of nonlinear sensor and motion models, non-Gaussian noise models, and non-Gaussian posterior probability distributions. Although particle filters typically incur more computational cost than parametric methods for nonlinear estimation, a well formulated particle filter generally results in a more accurate representation of the solution [93]. The method is presented here for completeness, as an existing technology. Specific algorithms remain an active field of research. For more details, the reader is referred to [93, 22, 37].

Each vehicle approximates  $p(\theta_t)$  with an onboard particle filter, incorporating the observations shared by all vehicles, with a set of  $N$  particles  $(\tilde{\theta}_{t,k}^{(i)}, w_{t,k}^{(i)})$  indexed by  $k$ , where  $\tilde{\theta}_{t,k}^{(i)} \in \Theta$  is the state of the particle, and  $w_{t,k}^{(i)} \in \mathbb{R}_+$  is the importance weight.<sup>1</sup> The particles represent  $p(\theta_t)$  by the probability mass function,

$$\hat{p}^{(i)}(\theta_t) = \sum_{k=1}^N w_{t,k}^{(i)} \delta(\theta_t - \tilde{\theta}_{t,k}^{(i)}) \quad (3.10)$$

where  $\delta(\cdot)$  is the Dirac delta function. This approximates  $p(\theta_t)$  over intervals in  $\Theta$ , with convergence results summarized in [18].

By maintaining a set of particles locally onboard each vehicle, only the observations need to be communicated, as opposed to the values of the entire set of particles. The particle filter iteratively incorporates new observations by predicting the state

---

<sup>1</sup>The second subscript of any variable denotes the index of the particle to which the variable belongs.

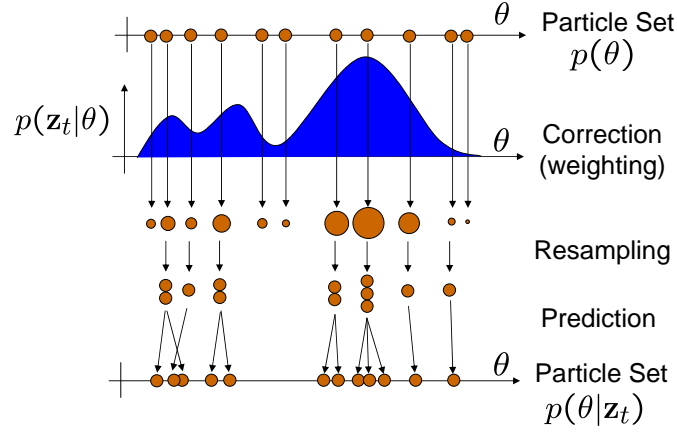


Figure 3.2: Graphical depiction of a 1-D particle filter. The filter is initialized by drawing “particles” from prior probability distribution  $p(\theta)$ , which is based on the available information. Sensor measurements  $\mathbf{z}_t$  and sensor models are used to weight particles according to their likelihood. The particles can then be resampled, according to their weights, to concentrate on regions with high likelihoods. The particle distribution is predicted for the subsequent time step, and the algorithm iterates [22].

of each particle, updating the importance weights with the likelihood of new observations, and then resampling the particles, as depicted in Fig. 3.2 and described in detail in [22, 37]. The sampling-importance-resampling algorithm is used for this work [22], with a low variance sampler [93] that has time complexity of  $\mathcal{O}(N)$ . The minimum mean square error (MMSE) estimate is

$$\hat{\theta}_t^{(i)} = \int_{\Theta} \theta_t p(\theta_t) d\theta_t \approx \sum_{k=1}^N w_{t,k}^{(i)} \tilde{\theta}_{t,k}^{(i)} \quad (3.11)$$

### 3.3.2 Determining Mutual Information from Particle Sets

To evaluate the mutual information utility function, (3.9), it can be expanded as [17],

$$\underbrace{I(\mathbf{z}_t; \theta_t)}_{\text{observation information}} = \underbrace{H(\mathbf{z}_t)}_{\text{observation uncertainty}} - \underbrace{H(\mathbf{z}_t | \theta_t)}_{\text{conditional observation uncertainty}} \quad (3.12)$$

From this expression, we see that minimizing the expected posterior uncertainty is equivalent to maximizing the difference between the uncertainty that any particular observation will be made,  $H(\mathbf{z}_t)$ , and the uncertainty of the measurement model,  $H(\mathbf{z}_t|\theta_t)$ .

To compute (3.12) using the particle filter representation, start with the first term,

$$H(\mathbf{z}_t) = - \int_{\mathcal{Z}} p(\mathbf{z}_t) \log p(\mathbf{z}_t) d\mathbf{z} \quad (3.13)$$

This cannot be directly evaluated, because  $p(\mathbf{z}_t)$  is not available as a continuous function. Rather, the observation distribution must be determined from the particle set and sensor model. First, expand the distribution as,

$$\begin{aligned} p(\mathbf{z}_t) &= \int_{\Theta} p(\mathbf{z}_t|\theta_t) p(\theta_t) d\theta_t \\ &= \int_{\Theta} \left( p(\theta_t) \prod_{j=1}^{n_v} p(\mathbf{z}_t^{(j)}|\theta_t) \right) d\theta_t \end{aligned} \quad (3.14)$$

where the second step expands the joint distribution using the requirement that the likelihoods of sensor observations are conditionally independent given the target state distribution. This requirement is exact when sensor noise is uncorrelated and due to local effects at the sensor, as is often the case, such as pixel or electrical noise. Otherwise, the conditional independence requirement can only be approximately met. Next, Monte Carlo integration techniques can be used [7]. Substituting (3.10) into (3.14) yields the particle filter approximation at the  $i^{\text{th}}$  vehicle,

$$p(\mathbf{z}_t) \approx \sum_{k=1}^N \left( w_{t,k}^{(i)} \prod_{j=1}^{n_v} p(\mathbf{z}_t^{(j)}|\theta_t = \tilde{\theta}_{t,k}^{(i)}) \right) \quad (3.15)$$

**Remark 3.2.** *The importance weights must be normalized to sum to one prior to Monte Carlo integration. That is, all  $w_{t,k}^{(i)}$  must be divided by  $\sum_{k=1}^N w_{t,k}^{(i)}$ .*

Substituting (3.15) into (3.13) yields the observation entropy of the distribution



represented by the particle filter approximation at the  $i^{\text{th}}$  vehicle,

$$H(\mathbf{z}_t) \approx - \int_{\mathcal{Z}} \left\{ \left( \sum_{k=1}^N \left( w_{t,k}^{(i)} \prod_{j=1}^{n_v} p(\mathbf{z}_t^{(j)} | \theta_t = \tilde{\theta}_{t,k}^{(i)}) \right) \right) \cdot \log \left( \sum_{k=1}^N \left( w_{t,k}^{(i)} \prod_{j=1}^{n_v} p(\mathbf{z}_t^{(j)} | \theta_t = \tilde{\theta}_{t,k}^{(i)}) \right) \right) \right\} d\mathbf{z} \quad (3.16)$$

This integration can then be performed using an appropriate numerical quadrature technique [71].

Next, similar methods are applied to compute

$$H(\mathbf{z}_t | \theta_t) = - \int_{\mathcal{Z}, \Theta} p(\mathbf{z}_t, \theta_t) \log p(\mathbf{z}_t | \theta_t) d\mathbf{z} d\theta_t \quad (3.17)$$

The joint distribution can be expanded using the chain rule and assuming conditional independence,

$$p(\mathbf{z}_t, \theta_t) = p(\theta_t) \prod_{j=1}^{n_v} p(\mathbf{z}_t^{(j)} | \theta_t) \quad (3.18)$$

Substituting (3.18) into (3.17) and applying the approximation of (3.10), yields the conditional observation entropy of the distribution represented by the particle filter approximation at the  $i^{\text{th}}$  vehicle,

$$H(\mathbf{z}_t | \theta_t) \approx - \int_{\mathcal{Z}} \sum_{k=1}^N \left\{ w_{t,k}^{(i)} \left( \prod_{j=1}^{n_v} p(\mathbf{z}_t^{(j)} | \theta_t = \tilde{\theta}_{t,k}^{(i)}) \right) \log \prod_{j=1}^{n_v} p(\mathbf{z}_t^{(j)} | \theta_t = \tilde{\theta}_{t,k}^{(i)}) \right\} d\mathbf{z} \quad (3.19)$$

Thus, the mutual information utility function, (3.9), can be found by using (3.16), (3.19), and (3.12). The step given in Remark 3.2 is also required.

**Remark 3.3.** *If  $p(\mathbf{z}_t)$  and  $p(\theta_t)$  are independent distributions, then evaluating  $I(\mathbf{z}_t; \theta_t)$  should return zero. To validate (3.16) and (3.19), define a hypothetical sensor model,*

$p(\mathbf{z}_t^{(j)}|\theta_t) = m(\mathbf{z}_t^{(j)})$ , independent of  $\theta_t$ . Then, (3.16) is

$$\begin{aligned} H(\mathbf{z}_t) &\approx - \int_{\mathcal{Z}} \left\{ \left( \sum_{k=1}^N \left( w_{t,k}^{(i)} \prod_{j=1}^{n_v} m(\mathbf{z}_t^{(j)}) \right) \right) \log \left( \sum_{k=1}^N \left( w_{t,k}^{(i)} \prod_{j=1}^{n_v} m(\mathbf{z}_t^{(j)}) \right) \right) \right\} d\mathbf{z} \\ &= - \int_{\mathcal{Z}} \left\{ \left( \prod_{j=1}^{n_v} m(\mathbf{z}_t^{(j)}) \right) \log \left( \prod_{j=1}^{n_v} m(\mathbf{z}_t^{(j)}) \right) \right\} d\mathbf{z} \end{aligned}$$

because  $m(\mathbf{z}_t^{(j)})$  can be factored out and  $\sum_{k=1}^N w_{t,k}^{(i)} = 1$ . It is similarly found that  $H(\mathbf{z}_t|\theta_t)$  is equal to the same expression. Therefore, in the limit where  $p(\mathbf{z}_t)$  and  $p(\theta_t)$  are independent distributions, (3.16) and (3.19) are equal, so  $I(\mathbf{z}_t; \theta_t)$ , given by (3.12), is zero.

This optimization remains highly coupled between the  $n_v$  vehicles, due to the integration over the domain of all sensors. Next, the degree of cooperation between the vehicles is analyzed to determine a scalable control strategy.

### 3.4 Mutual Information Approximations

The mutual information between the random variables  $\theta_t$  and  $\mathbf{z}_t$  quantifies the expected reduction in uncertainty. However, the computational complexity of using a particle set representation to evaluate this quantity grows exponentially with  $n_z$ , due to integration over each dimension. This section presents two different approximations to mutual information that can be evaluated in polynomial time with respect to the number of sensors. This makes the network scalable, yet capable of exploiting the descriptiveness of the particle filter. Note that the approximations are general; not specific to particle filters. This section proceeds by first defining the approximations and then quantifying and comparing the errors incurred.

First, consider the single-node approximation, defined as follows.

**Definition 3.2** (Single-Node Approximation). *The single-node approximation is used*

to approximate the mutual information utility function, (3.9), for optimization routines onboard the  $i^{\text{th}}$  vehicle, using:

$$V_s^{(i)}(\mathbf{x}_t, \mathbf{u}_t, p(\theta_t)) = I(\mathbf{z}_t^{(i)}; \theta_t) \quad (3.20)$$

This differs from (3.9) in that only the sensor aboard vehicle  $i$  is considered for computing the mutual information.

In the single-node approximation, a sensing node's utility function uses the previous observations of all sensing nodes, but only considers its own future observations. Although the vehicles directly cooperate through the distributed optimization, their local utility functions do not consider the effect of future observations of each other's sensors. This is equivalent to an approximation in the literature that has been applied to linearized, Gaussian estimators (e.g., [34]). The computational complexity of the single-node approximation is constant with respect to  $n_v$ .

To improve the approximation, the pairwise interactions of all vehicles are additionally considered. This more accurately captures the effect of group control inputs on mutual information (as will be proven in Theorem 3.3) with computational complexity linear in  $n_v$ . This new technique is the pairwise-node approximation,

**Definition 3.3** (Pairwise-Node Approximation). *The pairwise-node approximation is used to approximate the mutual information utility function, (3.9), for optimization routines onboard the  $i^{\text{th}}$  vehicle using:*

$$V_p^{(i)}(\mathbf{x}_t, \mathbf{u}_t, p(\theta_t)) = (2 - n_v) \left( I(\mathbf{z}_t^{(i)}; \theta_t) \right) + \sum_{\substack{j=1 \\ j \neq i}}^{n_v} \left( I(\mathbf{z}_t^{(i)}, \mathbf{z}_t^{(j)}; \theta_t) \right) \quad (3.21)$$

where  $n_v \geq 2$ ; otherwise (3.9) is readily used.

Using the pairwise-node approximation, sensing nodes additionally consider the effect of other vehicles' future observations, pairwise, on the utility of their own future observations. Whereas the single-node approximation leads to cooperative control due to emergent behavior from common knowledge of the target state distribution, the

pairwise-node approximation makes possible improved cooperation by approximating the effect of future observations of all other sensing nodes on the mutual information.

To quantify the error incurred in these approximations, a preliminary lemma is given for the subsequent theorems.

**Lemma 1** (Exchange of Conditioning Variables). *To exchange conditioning variables in mutual information, for any random variable  $a$ ,  $b$ , and  $c$ ,*

$$I(a; b|c) = I(a; b) - I(a; c) + I(a; c|b) \quad (3.22)$$

and

$$I(a; b|c) = I(a; b) - I(b; c) + I(b; c|a) \quad (3.23)$$

*Proof.* Take the difference between  $I(a; b|c)$  to  $I(a; b)$ ,

$$\begin{aligned} I(a; b) - I(a; b|c) &= H(a) - H(a|b) - H(a|c) + H(a|b, c) \\ &= I(a; c) - I(a; c|b) \end{aligned}$$

proving (3.22). Equation (3.23) follows by commuting the order of  $a$  and  $b$  in  $I(a; b)$  and  $I(a; b|c)$  above.  $\square$

To quantify the error incurred by the approximations, analytical expressions for the errors are derived and compared. First, consider the single-node approximation.

**Theorem 3.1** (Single-Node Approximation Error). *The difference between the single-node approximation for the  $i^{\text{th}}$  vehicle and the true value of (3.9) is  $\epsilon_s$ ,*

$$\epsilon_s^{(i)} = c_s^{(i)} + \sum_{\substack{j=2 \\ j \neq i}}^{n_v} \left( I(\mathbf{z}_t^{(j)}; \mathbf{z}_t^{(i)}, \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)}) \right) \quad (3.24)$$

where  $c_s^{(i)}$  encompasses the terms that are constant with respect to the  $i^{\text{th}}$  mobile sensor's control inputs.

*Proof.* Without loss of generality, consider the case of approximating the mutual information from the perspective of vehicle  $i = 1$ . The mutual information can be

expanded using the chain rule [17], and then rewritten using Lemma 1 to yield

$$\begin{aligned} I(\theta_t; \mathbf{z}_t) &= \sum_{j=1}^{n_v} I(\theta_t; \mathbf{z}_t^{(j)} | \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)}) \\ &= \sum_{j=1}^{n_v} \left( I(\theta_t; \mathbf{z}_t^{(j)}) - I(\theta_t; \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)}) + I(\theta_t; \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)} | \mathbf{z}_t^{(j)}) \right) \end{aligned}$$

For the chain rule, the intuitive interpretation is that the mutual information between the target distribution and all sensor measurement distributions is equal to the mutual information between the target distribution and the first sensor measurement distribution, plus the mutual information with the second sensor measurement conditioned on the first, and so on. Exchanging conditioning variables on the latter two terms using (3.23),

$$\begin{aligned} I(\theta_t; \mathbf{z}_t) &= V_s^{(1)}(\mathbf{x}_t, \mathbf{u}_t, p(\theta_t)) + c_s^{(1)} \\ &\quad + \sum_{j=2}^{n_v} \left( I(\mathbf{z}_t^{(j)}; \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)} | \theta_t) - I(\mathbf{z}_t^{(j)}; \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)}) \right) \end{aligned}$$

with constant  $c_s^{(i)} = - \sum_{\substack{j=1 \\ j \neq i}}^{n_v} I(\mathbf{z}_t^{(j)}; \theta_t)$ .

Applying the assumption that observations are conditionally independent given the target state, the first term in the summation is zero. Thus, generalizing to the  $i^{\text{th}}$  vehicle, the mutual information utility function can be evaluated using (3.20) with error given by (3.24).

□

Next, the error incurred by the pairwise-node approximation is derived.

**Theorem 3.2** (Pairwise-Node Approximation Error). *The difference between the pairwise-node approximation for the  $i^{\text{th}}$  vehicle and the true value of (3.9) is  $\epsilon_p$ ,*

$$\epsilon_p^{(i)} = \sum_{\substack{j=2 \\ j \neq i}}^{n_v} \left( I(\mathbf{z}_t^{(j)}; \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)} | \mathbf{z}_t^{(i)}) \right) \quad (3.25)$$

where  $n_v \geq 2$ .

*Proof.* Without loss of generality, consider the case of approximating the mutual information from the perspective of vehicle  $i = 1$ . The mutual information can be expanded using an application of the chain rule, separating the first term in the summation, and applying the chain rule again,

$$I(\theta_t; \mathbf{z}_t) = I(\theta_t; \mathbf{z}_t^{(1)}) + \sum_{j=2}^{n_v} \left( I(\theta_t; \mathbf{z}_t^{(1)}, \mathbf{z}_t^{(j)} | \mathbf{z}_t^{(2)}, \dots, \mathbf{z}_t^{(j-1)}) - I(\theta_t; \mathbf{z}_t^{(1)} | \mathbf{z}_t^{(2)}, \dots, \mathbf{z}_t^{(j-1)}) \right)$$

Exchanging conditioning variables in the summation using (3.23), canceling the resulting terms that sum to zero, and splitting the remaining summation, yields

$$I(\theta_t; \mathbf{z}_t) = V_p^{(1)}(\mathbf{x}_t, \mathbf{u}_t, p(\theta_t)) + \sum_{j=3}^{n_v} \left( I(\theta_t; \mathbf{z}_t^{(2)}, \dots, \mathbf{z}_t^{(j-1)} | \mathbf{z}_t^{(1)}, \mathbf{z}_t^{(j)}) - I(\theta_t; \mathbf{z}_t^{(2)}, \dots, \mathbf{z}_t^{(j-1)} | \mathbf{z}_t^{(1)}) \right)$$

Exchanging conditioning variables inside the summation, canceling terms summing to zero, and assuming conditional independence of observations given the target state yields,

$$I(\theta_t; \mathbf{z}_t) = V_p^{(1)}(\mathbf{x}_t, \mathbf{u}_t, p(\theta_t)) + \epsilon_p^{(1)}$$

Thus, the mutual information utility function can be evaluated using (3.21) with error given by (3.25). □

Now consider the effect that this added computational complexity has on the error terms, as a function of the value of the  $i^{\text{th}}$  vehicle's control inputs.

**Theorem 3.3** (Relative Accuracy of Mutual Information Approximations). *The magnitude of the error terms, that vary with the  $i^{\text{th}}$  vehicle's control inputs, in the single-node approximation, is greater than or equal to the magnitude of the pairwise-node*

approximation error terms, and equal only when the vehicle's observations are independent of all other vehicles. That is,

$$|\epsilon_s^{(i)} - c_s^{(i)}| \geq |\epsilon_p^{(i)}| \quad (3.26)$$

*Proof.* Subtract from the single-node approximation error, (3.24), the terms that do not vary with the  $i^{(th)}$  vehicle's control inputs,  $c_s^{(i)}$ , and apply the chain rule for mutual information [17],

$$I(\mathbf{z}_t^{(j)}; \mathbf{z}_t^{(i)}, \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)}) = I(\mathbf{z}_t^{(j)}; \mathbf{z}_t^{(i)}) + I(\mathbf{z}_t^{(j)}; \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)} | \mathbf{z}_t^{(i)}) \quad (3.27)$$

Mutual information is always greater than or equal to zero, and is equal only if the distributions are independent. So,

$$I(\mathbf{z}_t^{(j)}; \mathbf{z}_t^{(i)}, \mathbf{z}_t^{(1)}, \dots, \mathbf{z}_t^{(j-1)}) \geq I(\mathbf{z}_t^{(j)}; \mathbf{z}_t^{(2)}, \dots, \mathbf{z}_t^{(j-1)} | \mathbf{z}_t^{(i)}) \quad (3.28)$$

The magnitudes of the sums of the left and right sides of this equation, from  $j = 1$  to  $j = n_v$ , are equal to, respectively, the left and right sides of (3.26).

□

The pairwise-node approximation yields an estimate of the mutual information gain as good or better than the single-node approximation. When the vehicles' sensors make uncorrelated measurements, the single-node approximation is computationally faster and yields the same result as the pairwise-node approximation. However, if the observations are correlated, as is more frequently true, then the pairwise-node approximation yields a closer estimate.

Although the magnitude of the pairwise-node approximation error is less than that of the single-node approximation, it is not possible to guarantee that the optimization is not skewed by some systematic error between the exact solution and the single-node error. However, using the pairwise-node approximation still yields an approximate expected mutual information surpassing what seemed possible using the single-node approximation, and in experiments, the pairwise-node approximation yields better

results, as presented in Section 5.1.

In summary, no matter which method is used—the exact expression with (3.12), (3.16), and (3.19), the single-node approximation with (3.20), or the pairwise-node approximation with (3.21)—the vehicles can evaluate the mutual information utility function, in a decentralized manner, to enable them to cooperatively seek the target. By optimizing this objective function, they actively aim to reduce the uncertainty of their particle filter distributions. By using the single-node approximation, the vehicles can run the local optimization problem faster, and cooperate by trying to reduce the uncertainty of the same posterior distribution. By using the pairwise-node approximation, the computational expense of the objective function is reduced from the full problem, and the vehicles take into account the future observations that can be made collectively. In doing so, their objective functions reward them for maneuvering such that their sensor’s future observations, when combined, reduce the uncertainty of the target state distribution, at least as fast—or faster than—the single-node approximation. Next, consider how these objective functions are optimized in a distributed manner.

### 3.5 Distributed Control

The distributed control algorithm computes the output commands of the autonomous guidance system for the individual agents to implement. By distributing the problem to all agents, each agent contributes its own computational resources, improving the scalability of the system. Although it is possible to execute an optimization in a centralized manner, the computational complexity for a single iteration increases by a factor of the number of vehicles.

To perform distributed control using the information-seeking objective function, two approaches are proposed. Using the first approach, an iterative optimization, the vehicles solve local optimization problems, broadcast their solutions, and iterate until the algorithm converges. This approach ensures collision avoidance for any vehicle dynamics, and hence provides a baseline with which to implement the information-seeking algorithm. Using the second approach, a decoupled optimization, the vehicles



solve local optimization problems using the predicted location of other vehicles. This permits parallel computation without iteration, but requires an additional algorithm to ensure collision avoidance. The details of the algorithms follow.

### 3.5.1 Iterative Optimization

The mobile sensor network control problem is structured as a set of local optimal control problems for each sensing node, coupled through interconnecting constraints. The local optimization problem is formed holding the actions of the other vehicles fixed, and an iterative algorithm is implemented, based on [49], ensuring convergence to  $\epsilon$ -feasible solutions that satisfy the necessary conditions for Pareto optimality. The distributed algorithm iterates by communicating interim solutions of control inputs amongst the vehicles between local optimizations. This process can be hierarchical, synchronous, or asynchronous. Detailed algorithms and proof of convergence are in [49], for scenarios where global connectivity is assured.

To satisfy the interconnecting constraints, a penalty function [61] is defined for each vehicle as,

$$P(\mathbf{x}_t, \mathbf{u}_t) = \sum_{m=1}^{n_c^{(i)}} \max(0, g^{(i,m)}(\mathbf{x}_t, \mathbf{u}_t))^\gamma \quad (3.29)$$

where  $m$  indexes a set of  $n_c^{(i)}$  interconnecting inequality constraints,  $g^{(i,m)}$ , that affect vehicle  $i$ . The penalty function must be zero wherever the constraints are satisfied and must be differentiable. The  $n_v - 1$  collision avoidance constraints from (3.3) are written as,

$$\begin{aligned} g^{(i,m)} &= d_{min} - \|x_{t+1}^{(i)} - x_{t+1}^{(m)}\| \leq 0 \\ &\forall m \in \{1, \dots, n_v : m \neq i\} \end{aligned} \quad (3.30)$$

The penalty function is subtracted from the individual vehicle cost scaled by penalty parameter  $\beta$ , varying the tradeoff between constraint violation and the information theoretic cost, with an update  $\beta := \alpha\beta$  at each iteration, where  $\alpha \in (0, 1)$  is a design parameter. The local optimization problem based on the single-node approximation, (3.20), is

*Single-Node Local Optimization Program:*

$$\begin{aligned}
& \underset{\mathbf{u}_t^{(i)} \in U}{\text{maximize}} && V_s^{(i)}(\mathbf{u}_t, \mathbf{x}_t, p(\theta_t)) - \frac{1}{\beta} P(\mathbf{x}_t, \mathbf{u}_t) \\
& \text{subject to} && \mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \mathbf{u}_t) \\
& && \mathbf{z}_t = h(\mathbf{x}_{t+1}, \theta_t, \eta_t)
\end{aligned}
\tag{3.31}$$

The argument for this optimization program is the local control input. The other control inputs in the penalty function arguments are the current desired values communicated by other vehicles.

The single-node approximation will not vary with the control inputs of other vehicles, hence agreement between vehicles on the correct control actions for the group is not required. Only the collision avoidance constraint must be satisfied.

For the pairwise-node approximation, (3.21), the control inputs of all vehicles affect the objective function,  $V_p^{(i)}$ , hence slack variables must be added to decouple the local sensor costs, resulting in additional interconnecting constraints to include in the penalty function. Define the slack variable  $\tilde{\mathbf{u}}_t^{(i)}$  as the vector of all sensors' control inputs computed by the  $i^{\text{th}}$  sensor. Agreement among mobile sensing nodes on  $\tilde{\mathbf{u}}_t^{(i)}$  is realized through the penalty function enforced constraint,

$$\tilde{\mathbf{u}}_t^{(i)} = \tilde{\mathbf{u}}_t^{(j)} \quad \forall i, j \in \{1, \dots, n_v : j \neq i\} \tag{3.32}$$

The pairwise-node local optimization program is defined analogously to the single-node local optimization program, with optimization instead over the entire control vector  $\tilde{\mathbf{u}}_t^{(i)} \in U$ .

Potential extensions may include consideration of inter-agent coupling [64]. It is also interesting to consider cases in which there is not global connectivity, the communication is intermittent, or bandwidth is limited. These are topics of current research.

As described in Algorithm 1, the vehicles are ordered in a fixed manner that may be chosen arbitrarily. Initial solutions are determined locally by ignoring interconnected

constraints. Then, prior to local optimization, the relative weight of the penalty function with respect to the local cost is increased by a factor  $\alpha \in (0, 1)$ ,  $\beta := \alpha\beta$ , which increases the penalty of violating the interconnecting constraints gradually as the vehicles iterate on the solution. Each vehicle, starting with vehicle 1, solves the local information-seeking optimization problem with the current preferred solution and penalty parameter, and subsequently passes that solution and parameter onto the next vehicle. The optimization concludes when the solution agreed to is within  $\epsilon > 0$  of feasible and the local cost functions satisfy an appropriate convergence criteria. The algorithm is summarized below, for both the single-node approximation with local control input  $\mathbf{u}_t^{(i)}$  and the pairwise-node approximation with local network control vector  $\tilde{\mathbf{u}}_t^{(i)}$ .

---

**Algorithm 3.1** Single Time Step Distributed Optimization

---

- 1: Define  $\mathbf{x}_t$
  - 2: Initialize  $\mathbf{u}_t$  or  $\tilde{\mathbf{u}}_t$
  - 3: **repeat**
  - 4:    $\beta \leftarrow \alpha\beta$
  - 5:   **for**  $i = 1$  to  $n_v$  **do**
  - 6:     Transmit  $\mathbf{u}_t$  or  $\tilde{\mathbf{u}}_t$  to vehicle  $i$
  - 7:     Perform local optimization at vehicle  $i$
  - 8:     Update  $\mathbf{u}_t^{(i)}$  or  $\tilde{\mathbf{u}}_t^{(i)}$  for vehicle  $i$
  - 9:   **end for**
  - 10: **until** Convergence criteria satisfied
- 

This algorithm ensures that all vehicles, with control constraints, maintain collision-free operation while maximizing the information gain at each time step. Through the single-node and pairwise-node approximations, the algorithm can be computed in real-time and implemented for actual scenarios.

### 3.5.2 Decoupled Optimization

The distributed optimization algorithm of the previous section is capable of handling general vehicle dynamics. However, the computational cost of the optimization can be reduced by removing the need for the group to reach a consensus. The optimization

objectives at each vehicle already use local estimates of their contribution to the global utility function. By removing the need to reach agreement across the entire network, the optimization is decoupled, though the objective function still reflects the same coupled effects it had before.

The collision avoidance constraint must still be satisfied, but if the need for collision avoidance can be removed from the optimization, agents' optimizations can be fully decentralized, reducing the computational cost substantially. This eliminates the need to consider the control inputs of other agents, directly reducing the dimension of the problem. It also removes the need to iteratively communicate plans with other agents. The resulting optimization can be run faster, and hence at a higher rate.

In general, collision avoidance for groups of vehicles cannot be guaranteed. However, for some classes of vehicle dynamics, algorithms can be derived, as is the case for rotorcraft. In Chapter 4, a decentralized control law for quadrotor helicopters is derived. It uses a two-mode hybrid control strategy. In the standard mode, vehicles actively optimize their local approximations to the global information theoretic objective function. In the second mode, vehicles individually take collision avoidance action when necessary to prevent violating the collision avoidance constraint of other vehicles.

The decoupled optimization approach is used for the work presented in Chapter 5 with a collision avoidance algorithm derived in that chapter. It was found through simulation that, although collision avoidance actions are suboptimal, the performance of the algorithm was not adversely affected, and the run time complexity decreased.

## 3.6 Summary

This chapter has addressed the challenge of autonomous guidance of mobile sensors and mobile sensor networks. A novel approach was used to compute the information available to sensors using a particle filter to represent knowledge about the quantity being estimated. This approach overcomes approximations found in prior work to directly quantify the full potential performance for a sensor. To enable scalability

for mobile sensor network control, two approximation techniques were presented for computing the information available, and a distributed control law was presented to implement the information objective function with a network of vehicles.

Using these approaches, Chapter 5 discusses a series of applications. Then three different sensing modalities are simulated using the decoupled optimization to expose the characteristics of various algorithms. Finally, the results of flight experiments are presented. First, in Chapter 4, a decentralized collision avoidance scheme is derived for quadrotor helicopters to enable use of the decoupled optimization.

# Chapter 4

## Collision Avoidance

The decoupled optimization method for information theoretic control, presented in Chapter 3, enables cooperative control of a network of vehicles to work as a group to acquire information. However, this algorithm requires a separate collision avoidance method to assume control of the vehicles when necessary. To provide that capability, this chapter presents collision avoidance algorithms for mobile agents with the simplified dynamics of rotorcraft: second order dynamics with acceleration constraints. These algorithms are the ones used in Chapter 5 for simulations and flight experiments. The ability to ensure collision avoidance is critical—as the number of vehicles in a network increases, safety becomes challenging, even with human pilots, as shown anecdotally in Fig. 4.1.

The algorithms presented here use a decentralized cooperative switching control strategy for collision avoidance between vehicles. This rule-based approach is derived using optimal control and reachability analysis, as proposed previously for general system dynamics [96]. The algorithm allows higher level control logic to either have an added layer of safety, or to leave collision avoidance maneuvering to the collision avoidance control scheme. The goal of the presented scheme is to be minimally invasive; to only affect the control inputs when required to avoid a collision. As a result, the overall trajectories are likely to be suboptimal whenever a collision avoidance action is taken. However, when no collision is imminent, the presented control system does not interfere.



Figure 4.1: A mid-air collision between two out of three remote-piloted quadrotor helicopters operating in close proximity. The proposed algorithm would assume control when required to prevent imminent collisions.

Related work has been performed in the literature on flocking and general multi-agent systems, using rule- and optimization-based approaches. One example of a rule-based approach is potential methods. These have been used for an interesting sensor network control application, though without consideration for control input constraints [76]. Multi-agent systems with second order dynamics have been studied with a proposed decentralized control law using a detection shell and “gyroscopic” forces [14]. However, control constraints are not considered, and it is assumed that only one vehicle is in the detection shell at any time. First order dynamics have been used to formulate collision avoidance laws for aircraft, maintaining provable spacing between vehicles [21]. Virtual attractive-repulsive potentials have been used for cooperative control with second order dynamics, though input constraints are not considered [73]. Switching rules have been proposed for decentralized control, but collision avoidance was not considered [91].

Much related work has solved distributed and centralized optimization problems, though tradeoffs are made between computational efficiency for real-time execution

and guarantees of collision avoidance. An iterative distributed multi-agent optimization was formulated for general dynamics in [49], as used in Section 3.5.1. However, this exterior point method generates  $\epsilon$ -feasible solutions, and iterations between vehicles are time consuming as the size of the network increases. A numerical method to ensure aircraft collision avoidance was found using computational geometry, with guarantees when the solution is centralized [46]. One approach for distributed collision avoidance is to formulate a mixed integer linear program (MILP) using constraints on speed and acceleration [87]. The vehicles are ordered centrally and the optimizations are distributed, though run sequentially. By using loiter patterns, collision avoidance is guaranteed. Developments have been made to simplify the method, though a computationally intensive MILP step is still required [57]. A centralized nonlinear program was formulated for aircraft collision avoidance, though the computational expense scales poorly with the number of vehicles [83]. The multi-agent formation control problem has been decentralized by formulating the dual problem, with provable computational savings, though collision avoidance is not considered [82]. Decentralized nonlinear model predictive control was formulated for rotorcraft control, though collision avoidance is only enforced through potential functions, with no guarantees [90].

In the following, two control laws are presented, one for two vehicles, and one for  $n_v$  vehicles. In both, the vehicles compute analytical avoid set boundaries with respect to each other vehicle. When any vehicles are on the boundary of their avoid sets, collision avoidance action is taken. In the two vehicle scheme, a boundary is computed using optimal control that is proven safe analytically, and computed with trigonometric functions. The optimal action is computed numerically. Run time for two simulated vehicles was 0.2 ms. In the  $n_v$  vehicle scheme, pairwise avoid sets can be computed by linear algebra. A control law is presented with safety proven analytically for three vehicles, and validated for  $n_v > 3$  in simulation and analysis of scenarios. Computational complexity is  $O(n_v)$ . Run time for 200 interacting simulated vehicles was 9.2 ms. These control laws are applied in simulations of 2 to 200 vehicles. Finally, they are demonstrated in quadrotor helicopter flight experiments.



## 4.1 Problem Formulation

Consider a set of  $n_v$  vehicles where the state of the  $i^{th}$  vehicle is

$$\mathbf{x}_i = [x_i \ y_i \ \dot{x}_i \ \dot{y}_i]^T$$

Note that this formulation neglects potential spatial trajectories in  $\mathbb{R}^3$  for aerial, underwater, and space vehicles. The methods developed here can be extended to such scenarios, though it is often desirable to restrict the problem to maintaining in-plane spacing due to operational concerns such as downwash interactions.

The vehicles are modeled to have undamped second order dynamics with acceleration control inputs  $\mathbf{u}_i = [\theta_i \ a_i]^T$ , where the acceleration direction is  $\theta_i \in [0, 2\pi)$  and the magnitude is  $a_i \in [0, a_{max}]$ . This model was found in experiments to adequately approximate the dynamics of quadrotor helicopters [44], and is similar to that of many rotorcraft. Note that  $\theta_i$ , not bold-faced, is a control input used throughout this section, not to be confused with the vector used as the target state elsewhere in this dissertation.

To analyze vehicle collision avoidance, define the relative state of vehicle  $j$  with respect to  $i$  as  $\mathbf{x}_{i,j} = \mathbf{x}_j - \mathbf{x}_i$ . Define the distance between  $i$  and  $j$  to be  $d_{i,j} = \sqrt{x_{i,j}^2 + y_{i,j}^2}$ . The collision avoidance requirement is that

$$d_{i,j} \geq d_{min} \quad \forall \{i, j | i \in [1, n_v], j \in [1, n_v], j \neq i\} \quad (4.1)$$

where  $d_{min}$  is the minimum allowed distance between vehicle centers. Define the speed of  $i$  to be  $v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2}$  and the relative speed to be  $v_{i,j} = \sqrt{\dot{x}_{i,j}^2 + \dot{y}_{i,j}^2}$ . The equations of motion of the relative dynamics for any pair of vehicles are

$$f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) = \frac{\partial}{\partial t} \begin{bmatrix} x_{i,j} \\ y_{i,j} \\ \dot{x}_{i,j} \\ \dot{y}_{i,j} \end{bmatrix} = \begin{bmatrix} \dot{x}_{i,j} \\ \dot{y}_{i,j} \\ -a_i \cos \theta_i + a_j \cos \theta_j \\ -a_i \sin \theta_i + a_j \sin \theta_j \end{bmatrix} \quad (4.2)$$

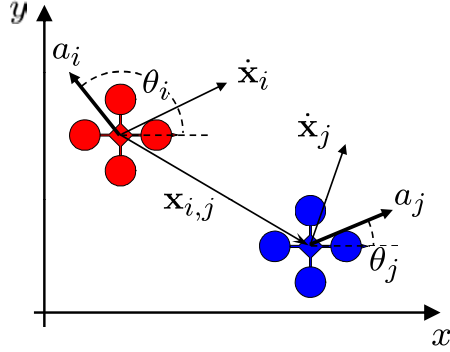


Figure 4.2: Relative states of two acceleration constrained quadrotor helicopters. The control inputs for aircraft  $i$  and  $j$  are accelerations  $a_i \in [0, a_{max}]$  and  $a_j \in [0, a_{max}]$  in directions  $\theta_i$  and  $\theta_j$ , respectively.

where  $\mathbf{u}_{i,j} = [\mathbf{u}_i^T \quad \mathbf{u}_j^T]^T$ .

## 4.2 Optimal Control Approach

The switched control approach is an adaptation of a pursuit-evasion game for non-cooperative control, where the pursuer's worst case control inputs are the worst case disturbances that the evader must be able to handle [96].

The goal for safe operation is to prevent the pairwise relative states from entering “keepout set”  $K$ , defined by (4.1). The pairwise loss function  $l(\mathbf{x}_{i,j})$  is defined such that

$$K = \{\mathbf{x}_{i,j} \in \mathbb{R}^4 | l(\mathbf{x}_{i,j}) < 0\} \text{ and } \partial K = \{\mathbf{x}_{i,j} \in \mathbb{R}^4 | l(\mathbf{x}_{i,j}) = 0\} \quad (4.3)$$

The set  $Pre_t(K) \subset \mathbb{R}^4$  can be computed from which the control strategy causes a vehicle to enter  $K$  in at most  $t$  time. The problem is formulated as a two-person, zero-sum dynamical game, where the “losing” states are calculated for the vehicles. The value function is taken to be  $J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t) = l(\mathbf{x}_{i,j}(0))$ . This is the cost of a trajectory that starts at  $t \leq 0$ , evolves according to the dynamics and control inputs, and ends at the final state  $\mathbf{x}_{i,j}(0)$ .

First, define the unsafe portion of  $\partial K$ , the boundary of  $K$ , as those states for which the control strategy results in the state flowing into  $K$ . The safe portion is

conversely defined. The outward pointing normal to  $K$  is defined as  $\nu^T = Dl(\mathbf{x}_{i,j})$ , hence the safe portion of  $\partial K$  is where  $\nu^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) \geq 0$  and the unsafe portion is where  $\nu^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) < 0$ .

The optimal control inputs for the pursuer-evader assumption are computed by the pursuer minimizing  $l(\mathbf{x}_{i,j}(0))$ , and the evader maximizing  $l(\mathbf{x}_{i,j}(0))$ . The main goal of either is to render  $l(\mathbf{x}_{i,j}(0))$  negative or positive, respectively. The optimization problem can be posed in terms of Hamilton's equations. To optimize with respect to  $J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$ , the control strategies for pursuer and evader must respectively maximize and minimize its time derivative [96]. This is

$$\frac{\partial J}{\partial t} = \left( \frac{\partial J}{\partial \mathbf{x}_{i,j}} \right)^T \frac{\partial \mathbf{x}_{i,j}}{\partial t} = \mathbf{p}^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) \quad (4.4)$$

where the costate,  $\mathbf{p} = \nabla_{\mathbf{x}_{i,j}} J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$ , with elements  $p_k : k \in [1, 4]$ . At  $t = 0$ , the game terminates, so the value of  $\mathbf{p}(0)$  is the gradient of  $J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$  at the terminal condition,  $l(\mathbf{x}_{i,j}(0))$ . That is,  $\mathbf{p}(0) = \nabla_{\mathbf{x}_{i,j}} l(\mathbf{x}_{i,j}(0))$ .

By maximizing (or minimizing)  $\mathbf{p}^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j})$ , with respect to the control inputs, the rate of approach of (or retreat from) the next level set can be controlled. The level sets can be back-propagated from the terminal state at  $\partial K$ .

An analytic solution is found in Section 4.3 for optimal control inputs that are required in a region of minimum size. For more than two vehicles, computing the reachable set becomes computationally expensive. Consequently, a suboptimal control law is presented in Section 4.4, inspired by the optimal control law, that has low computational overhead, and is trivially decentralized.

### 4.3 Two Vehicle Collision Avoidance

This section presents the optimal switching control law for two vehicle collision avoidance with acceleration constraints. First, the optimal control inputs are derived. Then, the set of relative states for which  $l(\mathbf{x}_{i,j}(0)) \leq 0$  is found, the ‘‘avoid set’’,  $A_{i,j}$ . Its boundary,  $\partial A_{i,j}$ , is the surface along which vehicles transition from nominal control to collision avoidance.

### 4.3.1 Optimal Control Input

The final time for the game is the time of closest approach, at  $t = 0$ . The cost function is,

$$J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t) = l(\mathbf{x}_{i,j}(0)) = d_{i,j}^2 - d_{min}^2 \quad (4.5)$$

The objective function, the Hamiltonian, is the rate of change of  $J(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}, t)$ , given by (4.4). Expanding it yields

$$\begin{aligned} H(\mathbf{x}_{i,j}, \mathbf{p}) &= \mathbf{p}^T f(\mathbf{x}_{i,j}, \mathbf{u}_{i,j}) \\ &= p_1 \dot{x}_{i,j} + p_2 \dot{y}_{i,j} + p_3 (-a_1 \cos \theta_i + a_2 \cos \theta_j) \\ &\quad + p_4 (-a_1 \sin \theta_i + a_2 \sin \theta_j) \end{aligned}$$

The optimization problem for evasion-evasion (analogous to pursuit-evasion) is

$$\begin{aligned} H^*(\mathbf{x}_{i,j}, \mathbf{p}) &= \\ \max_{\mathbf{u}_1} \cdot \max_{\mathbf{u}_2} \cdot &((p_1 \dot{x}_{i,j} + p_2 \dot{y}_{i,j}) - a_1 (p_3 \cos \theta_i + p_4 \sin \theta_i) \\ &+ a_2 (p_3 \cos \theta_j + p_4 \sin \theta_j)) \end{aligned} \quad (4.6)$$

To solve for each player separately, first note that at the extrema of the objective,  $a_1 = a_2 = a_{max}$ . Then, the derivative is taken with respect to the remaining control inputs. Consider the perspective of vehicle  $j$ ,

$$\frac{\partial}{\partial \theta_j} (p_3 \cos \theta_j + p_4 \sin \theta_j) = -p_3 \sin \theta_j + p_4 \cos \theta_j \quad (4.7)$$

The extrema, then, are the solutions to

$$-p_3 \sin \theta_j^* + p_4 \cos \theta_j^* = 0 \quad (4.8)$$

Thus,

$$\theta^* = \arctan\left(\frac{p_4}{p_3}\right) + n\pi \quad (4.9)$$

From the calculus of variations,

$$\dot{\mathbf{p}} = -\frac{\partial}{\partial \mathbf{x}_{i,j}} H^*(\mathbf{x}_{i,j}, \mathbf{p}) = \begin{bmatrix} 0 & 0 & -p_1 & -p_2 \end{bmatrix}^T \quad (4.10)$$

and

$$\mathbf{p}(0) = \frac{\partial}{\partial \mathbf{x}_{i,j}} J(\mathbf{x}_{i,j}(0)) = \begin{bmatrix} 2x_{i,j}(0) & 2y_{i,j}(0) & 0 & 0 \end{bmatrix}^T \quad (4.11)$$

Then, integrating to find the optimal control inputs yields

$$\mathbf{p}(t) = \begin{bmatrix} 2x_{i,j}(0) \\ 2y_{i,j}(0) \\ -\int_t^0 p_1 dt \\ -\int_t^0 p_2 dt \end{bmatrix} = \begin{bmatrix} 2x_{i,j}(0) \\ 2y_{i,j}(0) \\ -tx_{i,j}(0) \\ -ty_{i,j}(0) \end{bmatrix} \quad (4.12)$$

Substituting (4.12) into (4.9) yields the optimal input,

$$\theta_j^* = \arctan\left(\frac{y_{i,j}(0)}{x_{i,j}(0)}\right) + n\pi \quad (4.13)$$

Substituting (4.13) into (4.6) it is found that  $n = 1$  maximizes the optimization. Thus, it is optimal to accelerate away from the point of closest approach on  $\partial K$ .

### 4.3.2 Avoid Set

The boundary of the avoid set is the locus of points along which collision avoidance control is required. The collision avoidance control input is constant acceleration perpendicular to  $\partial K$  at the point of closest approach, so the path in relative coordinates is a parabola, as depicted in Fig. 4.3.

For analysis, a change of coordinates is performed. The change of coordinates rotates the relative coordinate frame such that in the rotated frame,  $\dot{x}_{i,j} = 0$  and  $\dot{y}_{i,j} < 0$ . Define the rotation angle of the avoid region, with respect to the relative

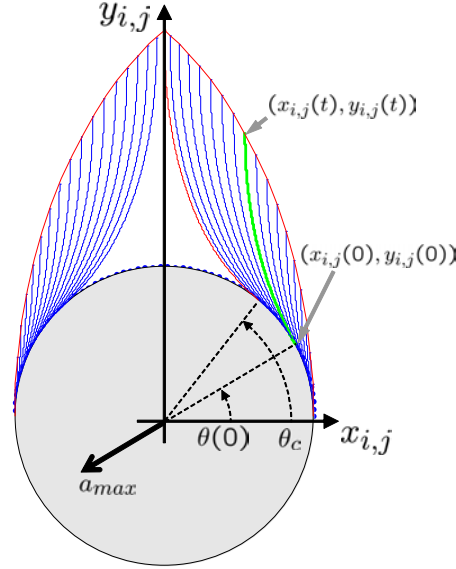


Figure 4.3: The keepout set (gray) and the avoid set (white) for two-vehicle collision avoidance, in the rotated coordinate frame. The parabolas are the trajectories that a vehicle follows from any point on the boundary. The avoid set is time varying and never crossed.

coordinate frame, to be<sup>1</sup>

$$\phi_{i,j} = \text{atan2}(\dot{y}_{i,j}, \dot{x}_{i,j}) - \frac{\pi}{2} \quad (4.14)$$

where the  $\frac{\pi}{2}$  offset orients the avoid set along the  $y_{i,j}$  axis for analysis; an arbitrary choice. Analysis of the avoid set for any  $\phi_{i,j}$  can be performed in the coordinates shown in Fig. 4.3, removing the need to consider  $\dot{x}_{i,j}$  in this frame.

For the remainder of this section, the relative coordinate frame is rotated by  $-\phi_{i,j}$ . To avoid over complicating the notation, the same variable names are used for the rotated frame; to return results to the original coordinate frame, they must be rotated by  $\phi_{i,j}$ , as depicted in the results in Fig. 4.5.

Boundary  $\partial A_{i,j}$  of the avoid set can be found in the rotated frame, as shown in Fig. 4.3, such that use of the optimal control input results in a closest approach of  $d_{min}$ . The conditions of the rotated frame can be used:  $\dot{x}_{i,j} = 0$ ,  $\dot{y}_{i,j} = -v_{i,j}$ . The

<sup>1</sup>As in ANSI-C,  $\text{atan2}(y, x)$  is similar to  $\arctan(\frac{y}{x})$ , but by using the signs of  $x$  and  $y$ , it returns the angle in the domain  $(-\pi, \pi]$  rather than  $(-\frac{\pi}{2}, \frac{\pi}{2}]$ .

parabola is rotated by  $\theta(0) - \pi/2$ , and has a second derivative in that direction of  $2a_{max}$ , resulting from collision avoidance action by both vehicles. Thus, the usable part of  $\partial A_{i,j}$  is defined by,

$$x_{i,j}(t) = d_{min} \cos \theta(0) - \frac{v_{i,j}(t)^2}{2a_{max}} \sin^2 \theta(0) \cos \theta(0) \quad (4.15)$$

$$y_{i,j}(t) = \left( d + \frac{v_{i,j}(t)^2}{a_{max}} \right) \sin \theta(0) - \frac{v_{i,j}^2}{2a_{max}} \sin^3 \theta(0) \quad (4.16)$$

To numerically test if vehicle  $j$  has crossed the boundary, the region can be approximated by a polygon with vertices generated using discrete values of  $\theta(0)$  from 0 to the critical angle,  $\theta_c$ , and then mirroring about the  $y_{i,j}$ -axis. The angle  $\theta_c$  is the one past which no points of closest approach occur, given the use of optimal control. It can be shown that  $\theta_c < \frac{\pi}{2}$  if  $2d_{min}a_{max} < v_{i,j}^2$ . To find  $\theta_c$ , (4.15) is solved with  $x_{i,j} = 0$ ,

$$\theta_c = \begin{cases} \arcsin \sqrt{\frac{2d_{min}a_{max}}{v_{i,j}(t)^2}} & \text{if } 2d_{min}a_{max} < v_{i,j}^2 \\ \pi/2 & \text{otherwise} \end{cases} \quad (4.17)$$

Then,  $(x_{i,j}, y_{i,j})$  can be tested to see if it is in the polygon.<sup>2</sup> If it is, then collision avoidance action must be taken.

To find the optimal control, the point of closest approach must be found. This can be found by solving (4.16) for  $\theta(0)$  using the current value of  $y_{i,j}$ . Note that by substituting  $\zeta = \sin \theta(0)$ , this is cubic function, with roots easily found numerically; the only physical root is in the domain  $[-1, 1]$ .

Note that synchronous control is not required, though to strictly avoid entering  $K$ ,  $A_{i,j}$  must be grown to account for all possible control inputs. When implemented in discrete time, the control law must test if the vehicles are at  $\partial A_{i,j}$ , which requires crossing the boundary of the avoid set by at most  $v_{i,j}\Delta t$ , where  $\Delta t$  is the discrete time step. To guarantee strict avoidance,  $d_{min}$  must be increased by this quantity.

The optimal collision avoidance action is to accelerate away from  $\theta(0)$ , or in the original relative coordinate frame,  $\theta(0) + \phi_{i,j}$ . This control law guarantees that no collisions will occur between two vehicles, and is required for a minimum portion of

---

<sup>2</sup>Standard algorithms can test if  $(x_{i,j}, y_{i,j})$  is in the polygon, though here it is more efficient to use a cross product test with the points on the boundary horizontally closest to  $x_{i,j}$ .

the physical area. However, to address many interacting vehicles, further development is required, as considered next.

## 4.4 Multi-Vehicle Collision Avoidance

### 4.4.1 Optimal Control Input

Following the same methods used to derive (4.13), more vehicle interactions can be included, yielding the optimal control law for many vehicles,

$$\theta_i^* = \arctan \left( \frac{y_{i,1}(0) + y_{i,2}(0) + \dots}{x_{i,1}(0) + x_{i,2}(0) + \dots} \right) + \pi \quad (4.18)$$

However, it is computationally expensive to find  $\mathbf{x}_{i,j}(0)$  for  $n_v > 2$ , so this control law will not be used here. Rather, it is used to inspire a proposed alternate control strategy,

$$\theta_i = \arctan \left( \frac{y_{i,1}(t) + y_{i,2}(t) + \dots}{x_{i,1}(t) + x_{i,2}(t) + \dots} \right) + \pi \quad (4.19)$$

where the vehicles considered are those at the boundaries of their respective avoid sets,  $A_{i,j}$ . Again, use  $a_i = a_{max}$ . The control law is equivalent to accelerating away from the centroid of all vehicles that must be actively avoided. Additional logic is required as described below.

This control law is suboptimal in the sense that its use may be required in regions for which another collision avoidance control law would not require action. However, it yields enormous computational savings, and will be shown to yield collision avoidance using a reasonably small avoid set. Now the avoid sets must be found.

### 4.4.2 Avoid Set

The proposed separate avoid sets for each vehicle are shown in Fig. 4.4. They are inspired by the avoid set for two vehicle collision avoidance, and are again aligned



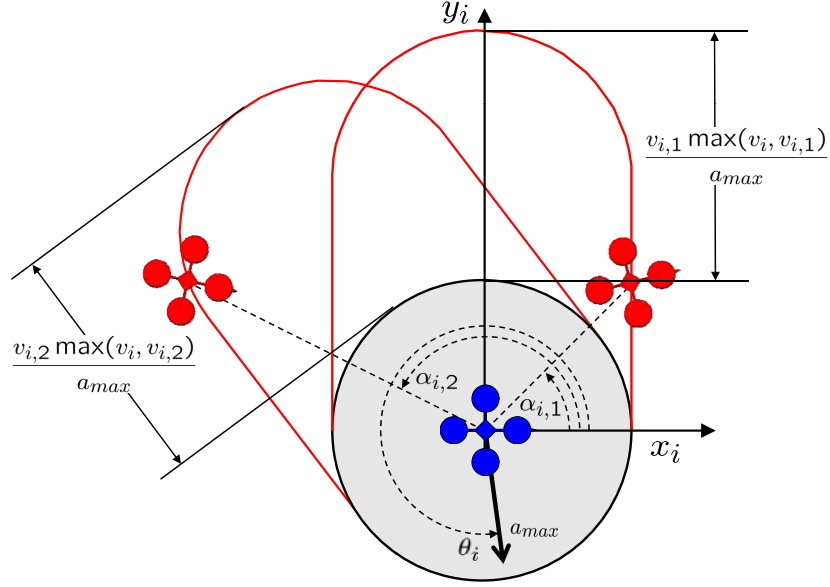


Figure 4.4: Vehicle  $v$  accelerates at  $a_{max}$  in direction  $\theta_v$  to avoid vehicles 1 and 2, which are at the edge of their respective avoid sets. The orientation of the avoid set is that of the relative velocity vectors.

with the relative velocity direction. The length of the proposed region is

$$L_{i,j} = \frac{v_{i,j} \max(v_i, v_{i,j})}{a_{max}} \quad (4.20)$$

in order to maintain the minimum separation distance, as shown in the remainder of the section. This avoid set is designed such that the control law provably works for two and three vehicles. It is validated analytically and numerically for  $n_v > 3$  in selected configurations assumed to be most challenging to the algorithm.

This section analyzes the dynamics of vehicles on their respective  $\partial A_{i,j}$  to validate that they do not cross into  $A_{i,j}$ . Because  $K \subset A_{i,j}$ , this validates that no vehicle can enter  $K$ .

**Behavior on avoid set straight-edge boundaries.** To analyze each of the  $n_v - 1$  avoid sets for vehicle  $i$ , again a change of coordinates rotates the relative coordinate frame by  $\phi_{i,j}$ , as defined in (4.14). The rotation rate of this frame of reference is found by differentiating, using (4.2), and using the fact that  $\dot{x}_{i,j} = 0$  in

this rotated frame.

$$\dot{\phi}_{i,j} = \frac{\dot{y}_{i,j}}{v_{i,j}^2}(-a_i \cos \theta_i + a_j \cos \theta_j) \quad (4.21)$$

Therefore, if vehicle  $j$  is on a straight edge of  $\partial A_{i,j}$ , to guarantee collision avoidance, it must accelerate away from that edge more than vehicle  $i$  accelerates toward it. For two vehicles, this is clearly the case; the vehicles accelerate away from the centroid, yielding the required behavior. For three vehicles, this is achieved by adding additional logic to accelerate directly away from the interior when an edge is crossed. For more than three vehicles, the same logic is applied. In this case, the number of permutations of active constraints makes an analytical proof of safety difficult: simulation results were used to validate safety of this algorithm along the straight-edged boundary.

**Behavior on avoid set curved boundaries.** From (4.20),  $L_{i,j} \geq v_{i,j}^2/a_{max}$ . In the rotated frame,  $\dot{L}_{i,j} \leq 2\dot{y}_{i,j}\ddot{y}_{i,j}/a_{max}$ . To prove that  $d_{i,j} \geq d_{min}$  is not violated, it is sufficient to show that  $\dot{L}_{i,j} \leq \dot{y}_{i,j}$  because when  $L_{i,j} = 0$ ,  $A_{i,j} = K$ . Using (4.2), this is equivalent to

$$\sin \theta_j \geq \sin \theta_i + 1/2 \quad (4.22)$$

This is clearly satisfied by substituting in (4.19) for two or three vehicles; at worst for three vehicles, one vehicle is directly between two other vehicles, in which case margin still exists in the above inequality. For  $n_v > 3$ , again the permutations of active constraints make an analytical proof of safety difficult. Analytical verification for challenging scenarios is presented in Section 4.4

**Behavior for combined boundary types.** For three vehicles, with vehicle 1 on the straight-edge boundary of  $A_{i,1}$  and vehicle 2 on the curved boundary of  $A_{i,2}$ , the vehicle on the straight edge is alone capable of ensuring that the rotation rate of the avoid set given by (4.21) rotates the set away from the vehicle, or at least keeps the rotation angle stationary. The vehicle on the curved boundary is capable of satisfying (4.22) provided vehicle  $i$  does not accelerate toward it. This final condition is satisfied by design of the discrete time implementation to prevent chattering effects. As mentioned previously, the boundary must be crossed to be detected for numerical implementation. Rather than chatter arbitrarily, vehicle  $i$

uses the input corresponding to the boundary crossed by the largest distance, which guarantees separation in the three vehicle scenario. Again, for  $n_v > 3$ , simulation is used to validate safety for this configuration.

**Analysis for more than three vehicles.** To analyze the effect of including more vehicles, consider two configurations with the potential to cause failure: a large circle of vehicles converging toward one point within the circle, and a large line of vehicles converging toward one point on the line. The algorithm is proven analytically to guarantee that separation is maintained in both cases.

To analyze the converging circle, let  $d$  be the spacing between any two adjacent vehicles in the circle, and their initial speeds be  $v$ . To find the initial  $d$  at time  $t_0$  such that the vehicles can apply  $a_{max}$  to stop by  $t = 0$  with  $d(0) = d_{min}$ , start by computing the relative speed. The angle between the velocity vectors for  $n_v$  vehicles lying equally spaced on a circle is  $2\pi/n_v$ . For adjacent vehicles,

$$v_{i,j} = \dot{d}(t_0) = -2v \sin(\pi/n_v) \quad (4.23)$$

When the vehicles begin to apply  $a_{max}$ ,  $\dot{v} = -a_{max}$ , so  $\ddot{d} = 2a_{max} \sin(\pi/n_v)$ . Integrating this and using (4.23),

$$\dot{d}(t) = -2v(t_0) \sin(\pi/n_v) + 2a_{max}t \sin(\pi/n_v) \quad (4.24)$$

The rate of change of spacing at the point of closest approach is zero, so by setting (4.24) to zero it is found that  $t_f = v(t_0)/a_{max}$ . Integrating once more,

$$d(t) = d(t_0) - 2v(t_0)t \sin(\pi/n_v) + a_{max}t^2 \sin(\pi/n_v) \quad (4.25)$$

Setting  $d(t_f) = d_{min}$ ,

$$d(t_0) = d_{min} + \frac{v(t_0)^2}{a_{max}} \sin(\pi/n_v) \quad (4.26)$$

Therefore, violation is avoided if

$$L \geq \frac{v(t_0)^2}{a_{max}} \sin(\pi/n_v) = \frac{v_i v_{i,j}}{2a_{max}} \quad (4.27)$$

This length is no more than half that of (4.20), hence the proposed control law is guaranteed to maintain  $d > d_{min}$ .

To analyze the converging line of vehicles, consider vehicles  $i = 1$ ,  $j = 2$ , with  $y_{1,2} = 0$ ,  $x_1 > x_2$ ,  $x_1 > 0$ , and  $\dot{x}_1 > \dot{x}_2$ . Then,  $\dot{L}_{1,2} = \frac{\dot{x}_1(\dot{x}_1 - \dot{x}_2)}{a_{max}}$ , so

$$\dot{L}_{1,2} = \frac{1}{a_{max}} (2\dot{x}_1\ddot{x}_1 - \ddot{x}_1\dot{x}_2 - \ddot{x}_2\dot{x}_1) \quad (4.28)$$

To guarantee collision avoidance, it is sufficient for this scenario that the  $L_{1,2}$  shrinks faster than the approach rate of the two vehicles;  $\dot{L}_{1,2} \leq \dot{x}_2 - \dot{x}_1$ . The inequality becomes

$$\ddot{x}_2 \geq a_{max}\left(1 - \frac{\dot{x}_2}{\dot{x}_1}\right) + \ddot{x}_1\left(2 - \frac{\dot{x}_2}{\dot{x}_1}\right) \quad (4.29)$$

If  $\ddot{x}_1 = -a_{max}$ ,

$$\ddot{x}_2 \geq -a_{max} \quad (4.30)$$

which is true due to constraints. Therefore, one vehicle chasing another vehicle always has the ability to stop by applying maximum acceleration away from that vehicle when using (4.20). Thus, any string of vehicles chasing one another, similarly, can stop.

## 4.5 Collision Avoidance Results

The proposed control laws were tested in simulation and flight experiments. All simulated vehicles are quadrotor helicopters with second order dynamics and  $\ell^2$ -norm constraints on acceleration. Simulations were run in Matlab using one core of a Core Duo 2.16 GHz. The number of vehicles demonstrated would be computationally overwhelming for other methods known to the authors. The timing results for the simulations are shown in Table 4.1.

Flight experiments used STARMAC II quadrotor helicopters tracking attitude commands from human pilots [44]. These experiments validated the results from simulations. Note that the collision avoidance algorithm does not prevent deadlock. However, deadlock was most severe when relative states have high symmetry to numerical precision—an improbable condition in real systems, as observed in flight

Table 4.1: Collision Avoidance Computation Time

Algorithm	# Vehicles	Time (ms)
Two-Veh.	2	0.23
Many-Veh.	2	0.16
Many-Veh.	8	0.41
Many-Veh.	32	1.6
Many-Veh.	64	3.6
Many-Veh.	100	4.6
Many-Veh.	200	9.2

experiments.

### 4.5.1 Two Vehicles

The two vehicle collision avoidance algorithm was used for two trajectory tracking vehicles flown toward each other with a variety of crossing paths, speeds, and acceleration constraints. The circular set  $K$  and avoid set  $A_{i,j}$  are shaded in Fig. 4.5. As the control action is taken, the relative velocity changes. This causes the avoid sets to morph, keeping the vehicles on each other's boundaries. The vehicles never enter the avoid set; the avoidance action is required while the vehicles are on each other's boundaries. At the end of the avoidance maneuver, they resume line tracking.

### 4.5.2 Many Vehicles

Sets of 2 to 200 vehicles were simulated doing trajectory tracking using many-vehicle collision avoidance on a variety of collision-courses. One scenario is in Fig. 4.6, with the separation distance between each vehicle in Fig. 4.7. The vehicles navigate past one another, with deadlocks resolved by asymmetries when the trajectory tracking control is allowed to resume. Timing from these simulations, in Table 4.1, shows that run time at each vehicle was approximately  $n_v \times 0.05$  ms.

Many challenging scenarios were simulated to validate performance in unreasonably complicated situations. One such case shown in Fig. 4.8 has three rings of 16 vehicles converging, with the outermost rings traveling faster. The  $n_v - 1$  avoid sets

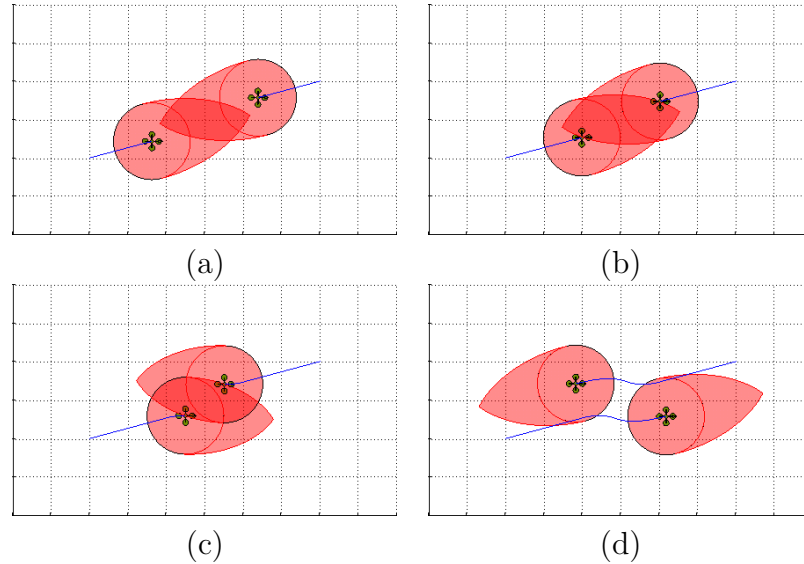


Figure 4.5: Simulation of two quadrotor helicopters approaching one another while tracking trajectories (a). When the vehicles (b) touch each other’s avoid set, the sets extending from the circular unsafe sets, they (c) apply collision avoidance control inputs. After the conflict is resolved, they (d) resume their previous trajectories.

at each vehicle are omitted from the plot for clarity. Collision avoidance must prevent lines of vehicles from piling up, and rings of vehicles being wedged to together. Spacing is maintained as shown in Fig. 4.7. Due to numerical precision, the vehicles are pushed off of their trajectories and approach their destination, the origin.

Another challenging case is a large circle of vehicles converging, as shown in Fig. 4.10, where the time history of relative distances again verified that separation was maintained. The vehicles eventually slide past one another and follow their trajectories. In both the 48 and 64 vehicle scenarios, deadlock was observed and lasted for differing times, though it was eventually resolved due to machine precision. When deadlocked, the system maintained separation, as was the case for all simulations.

### 4.5.3 Flight Experiments

The many-vehicle algorithm was implemented in the onboard computers of the STAR-MAC testbed. The software on each quadrotor broadcasts vehicle states to all other

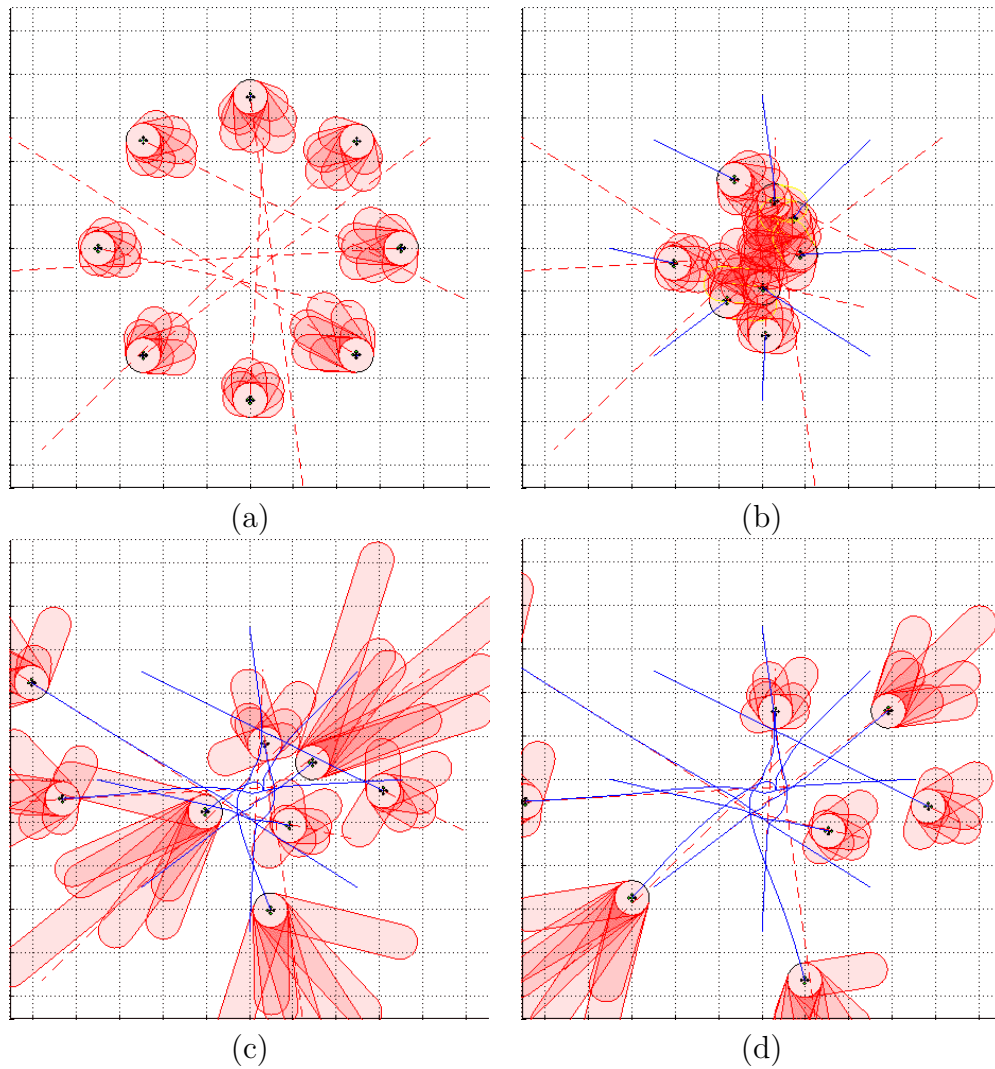


Figure 4.6: Simulation of 8 vehicles tracking randomly generated trajectories, using the many-vehicle collision avoidance law. The keepout sets are circles, and shaded long regions are avoid sets. Only a few regions touch the vehicles to which they correspond, leading to collision avoidance action, as highlighted in (b).

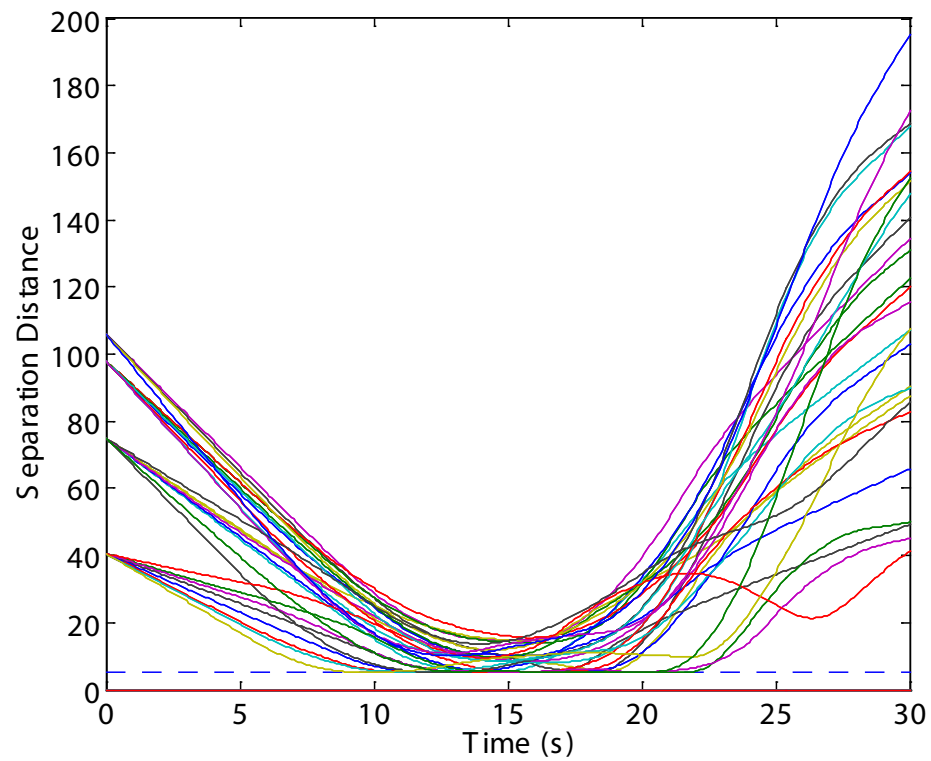


Figure 4.7: Separation distances between each vehicle for the eight vehicle simulation in Fig. 4.6, and a line showing the minimum allowed distance,  $d_{min} = 2$ . Separation is maintained throughout the simulation.



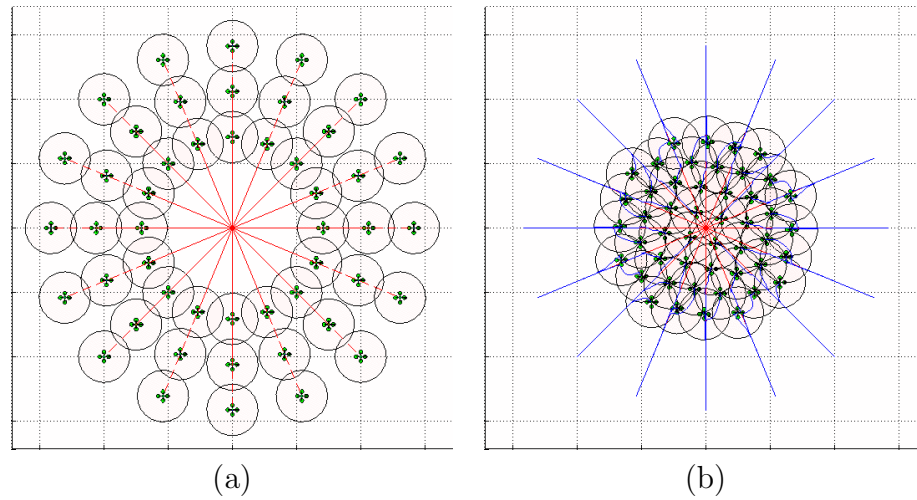


Figure 4.8: Simulation of 3 rings of 16 vehicles converging toward the same point, with the outermost rings moving fastest, (a) at  $t = 0$  and (b) at  $t = 10$ . Many avoid sets are active for each vehicle due to range of speeds of the rings.

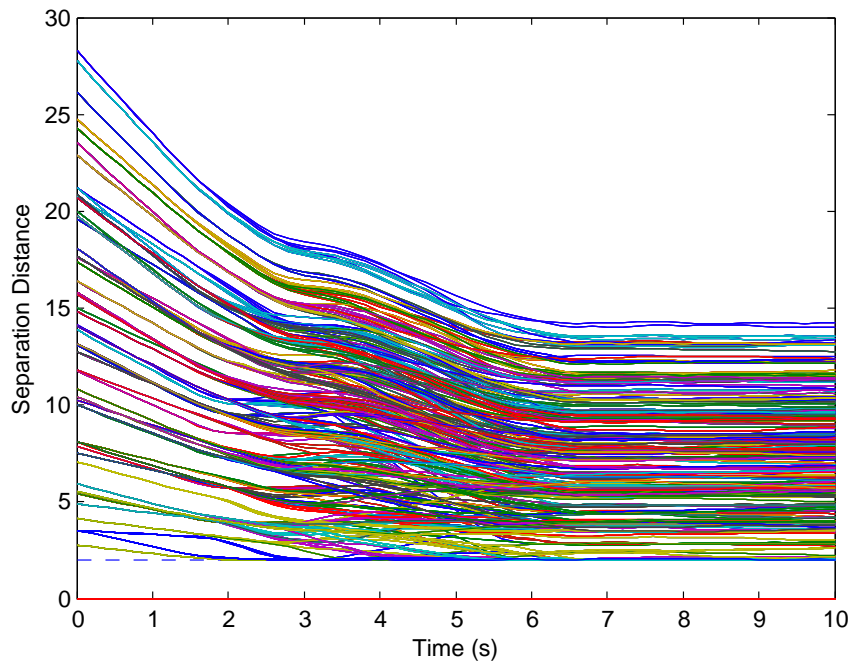


Figure 4.9: Separation distances between each vehicle for the 48 vehicle simulation in Fig. 4.8, and a line at the minimum allowed distance,  $d_{min} = 2$ . Separation is maintained throughout the simulation.

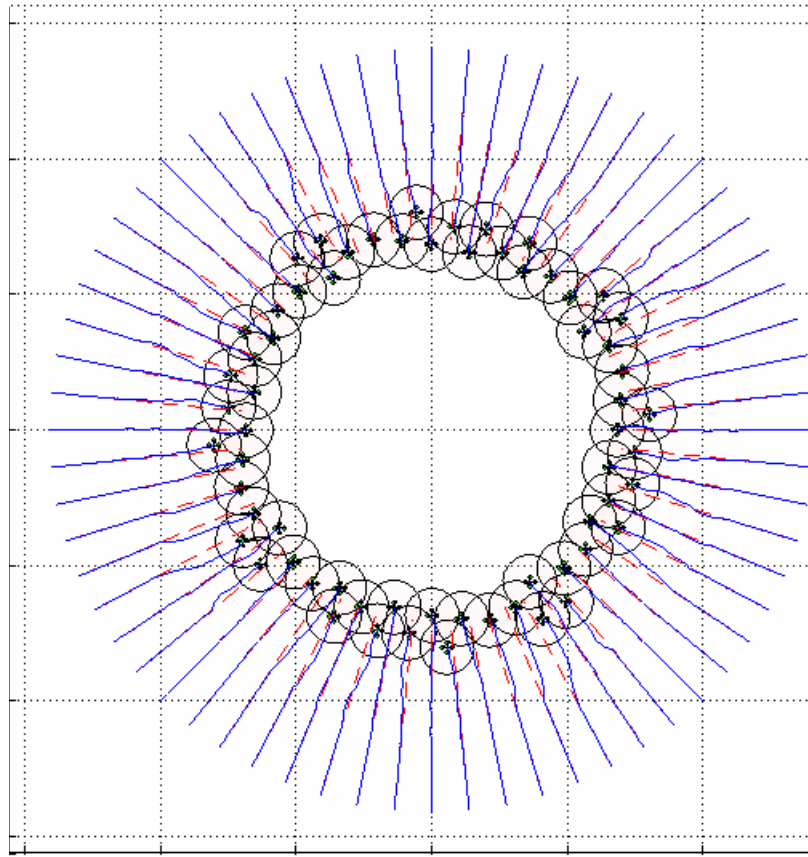


Figure 4.10: Simulation of a ring of 64 vehicles converging toward a center. The many-vehicle collision avoidance algorithm safely prevents the vehicles from being wedged together.

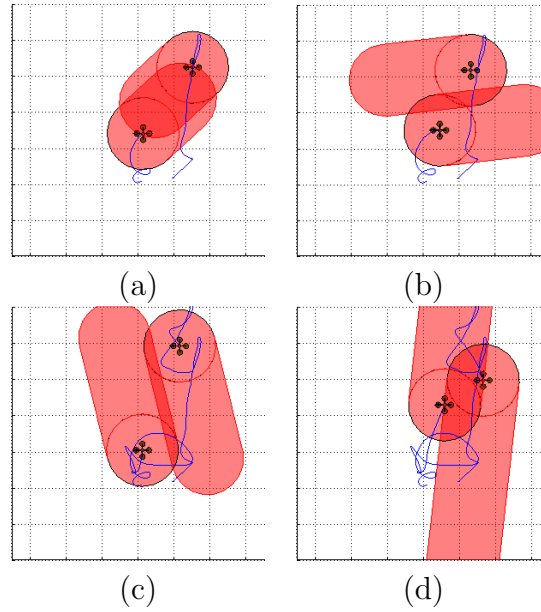


Figure 4.11: Automatic collision avoidance flight experiment using  $d_{min} = 2$  m with human control inputs attempting to cause collisions. (a) A conflict is detected at  $t = 25$  s and (b) recovered from by  $t = 26$  s. (c) The aircraft approach at  $t = 35$  s, resulting in (d) the conflict at  $t = 36$  s.

quadrotors at 10 Hz. The algorithm uses  $d_{min} = 2$  m,  $a_{max} = 1.7 \frac{m}{s^2}$ , and runs on the 600 MHz PXA270 on the aircraft.

The vehicles nominally used either attitude reference commands from a human pilot, or a waypoint tracking controller. The human pilots issued malicious commands to attempt to instigate collisions, which the collision avoidance algorithm successfully prevented, as shown in Figs. 4.11 and 4.12. The effect of time discretization was not included in  $d_{min}$  for the flight software implementation due to the fast update rate relative to vehicle speeds. In all but one instance, the minimum separation distance was maintained. Once incident occurred where it was violated by 0.1 m when  $v_{1,2} = 5 \frac{m}{s}$ . This is less than the  $v_{1,2} \Delta t$  safety margin required for time discretization, verifying the proposed approach.

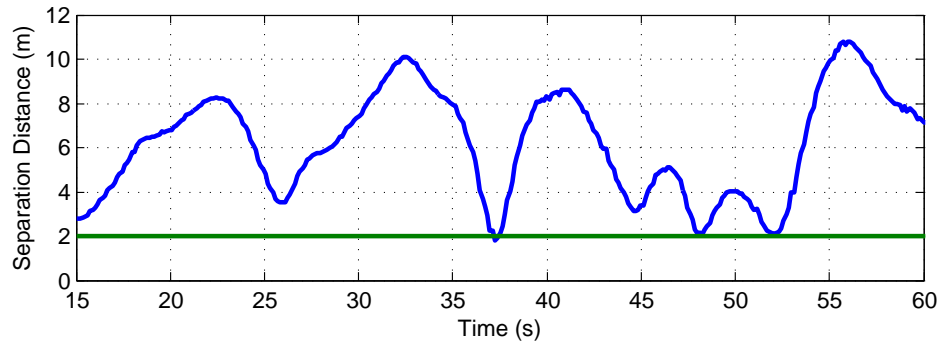


Figure 4.12: Separation according to GPS data for the flight experiment shown in Fig. 4.11. Even without extending  $d_{min}$  to account for time discretization, at  $v_{1,2} = 5 \frac{m}{s}$ ,  $d_{min}$  was violated by 0.1 m, less than  $v_{1,2}\Delta t$ .

## 4.6 Summary

Decentralized collision avoidance algorithms were presented for systems with second order dynamics and acceleration constraints. By using a switching control law with easily computed avoid sets, the control laws provide safety while using any desired external control law. Two control laws were presented, one for two vehicles and one for more than two vehicles. The vehicles computed an avoid set with respect to each other vehicle. When one or more vehicles are on the boundary of their avoid sets, collision avoidance action is taken by both vehicles. These control laws were applied in simulations of scenarios for which existing techniques either fail or are computationally expensive. The techniques can be used to simplify the distributed optimization for information theoretic mobile sensor network control, greatly reducing the computational complexity.

# Chapter 5

## Mobile Sensor Applications

To evaluate the characteristics of the information-seeking algorithms from Chapter 3, this chapter explores their application in an actual mobile sensor network, in simulation and in experiment. The guidance algorithms from Chapter 3 are applied to three different sensing modalities to explore the characteristics of the algorithms. These results are compared and contrasted with linear Gaussian methods. The differences in quantification of information demonstrate improvement beyond the state-of-the-art. The algorithms are then applied to an experiment with the STARMAC testbed in Section 5.2, demonstrating autonomous search for an avalanche rescue beacon.

### 5.1 Application to Sensing Modalities

This section explores three sensing modalities. The first category is bearings-only sensors, such as cameras and directional antennae [16], in which the direction to the target is measured. This modality permits comparison to related work. The second category is range-only sensors, including techniques such as signal strength measurements and time-of-flight measurements [39], in which distance to the target is measured. Localization using this modality is more prone to error using standard linearization techniques, but control actions generated using the non-parametric algorithms presented here match the analytically optimal behavior. The final category is

personal radio beacons, such as those used for avalanche rescue [51]. Here, measurements are made of the dipole magnetic field emitted by the beacon, with a nonlinear, periodic structure, where the prior probability distribution can be substantially non-Gaussian in structure. In all scenarios, the mobile sensor dynamics are modeled to be the dynamics of the quadrotor helicopters in STARMAC presented in Chapter 2.

The following examples use the decoupled optimization from Chapter 3 with the many-vehicle switching control law from Chapter 4 for collision avoidance. Only vehicle states and observations need to be communicated. Then local optimizations are performed to estimate the vehicle's ability to contribute to the global information of the system. The many-vehicle switching control law is used to ensure the satisfaction of collision avoidance constraints.

Simulations of the complete algorithm performed well. An example showing collision avoidance regions for information-seeking vehicles is shown in Fig. 5.6. It was found that the performance of the decoupled optimization, as measured by decrease in entropy, experienced no significant change from the iterative optimization, even though the collision avoidance regions required suboptimal control inputs when encroached upon. Further, the computational savings of executing all optimization algorithms locally and simultaneously decreased the required run time by a factor of  $n_v$ . The resulting system demonstrated consistent collision avoidance and fast localization of the search target.

### 5.1.1 Bearing-Only Sensors

For this example, consider sensors that measure the direction to the target, such as cameras [30] or directional antennae [16]. It is demonstrated that the non-Gaussian posterior probability distribution can be captured using the particle filter representation and directly used by the particle filter mutual information utility function. Results using the single-node approximation show that emergent behavior due to prior information can be sometimes beneficial, but sometimes counterproductive. The new pairwise-node approximation yields more consistent behavior that results in better performance, on average. By using particle filter methods, the bias, underestimated

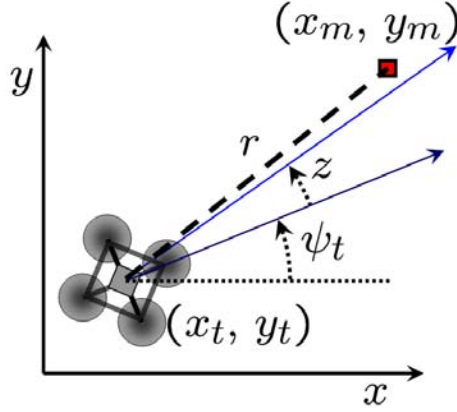


Figure 5.1: Bearings-only measurement model, where  $z$  is a measurement of the direction from the position of the sensor  $(x_t^{(i)}, y_t^{(i)})^T$  to the position of the target  $(x_m^{(i)}, y_m^{(i)})^T$ . It differs from the true direction due to additive noise, given by (5.1). Examples of such sensors include cameras and directional antennae.

covariance, and divergence associated with EKF's [58, 84, 78], can be avoided. No divergence was encountered in simulations of the methods, and the mutual information optimization successfully maneuvers the vehicle to extract information about the probability distribution represented by the particle set.

### Measurement Model

Consider searching for a target in the  $xy$ -plane. The location of the  $i^{th}$  search vehicle at time  $t$  is  $(x_t^{(i)}, y_t^{(i)})^T$ , components of  $\mathbf{x}_t^{(i)}$ . The state of the target is its location  $\theta = (x_m, y_m)^T$ . The bearing measurement model is

$$h_b^{(i)}(\mathbf{x}_t^{(i)}, \theta, \eta_t^{(i)}) = \arctan\left(\frac{y_m - y_t^{(i)}}{x_m - x_t^{(i)}}\right) - \psi_t^{(i)} + \eta_t^{(i)} \quad (5.1)$$

where  $h_b$  is the model of the bearing measurement with noise, as shown in Fig. 5.1,  $\psi_t^{(i)}$  is the  $i^{th}$  vehicle's heading and  $\eta_t^{(i)} \sim \mathcal{N}(0, \sigma_b^2)$  is the measurement noise. Although any measurement model could be used for this particle filter implementation, such as one including pixel noise in a camera, or one including signal attenuation with range for a directional antenna, the additive noise model used here allows comparison to

previous work (e.g., [34, 29, 63]). The variance is  $\sigma_b^2$ , which for simplicity is chosen to be the same for all sensors.

### Predicted Behavior

For bearings-only sensors, the optimal control actions for a linearized, Gaussian approximation of the system provide a reasonable predictor of the behavior of the actual, optimal system, with some exceptions. With this approximation, trends in optimal sensor placement and the effect of increasing  $n_v$  can be derived. Let  $p(\theta_t)$  be approximated as Gaussian with mean  $(\hat{x}_m, \hat{y}_m)^T$  and covariance  $\Sigma$ . The Jacobian of (5.1) is

$$J_b^{(i)} = \frac{1}{r^{(i)}} \begin{bmatrix} \sin \xi^{(i)} & -\cos \xi^{(i)} \end{bmatrix} \quad (5.2)$$

where  $\xi^{(i)} = \arctan 2(\hat{y}_m - y_t^{(i)}, \hat{x}_m - x_t^{(i)})$  and  $r^{(i)} = \sqrt{(\hat{x}_m - x_t^{(i)})^2 + (\hat{y}_m - y_t^{(i)})^2}$ . The goal is to minimize the conditional entropy, as in (3.5). Using the entropy formula for Gaussians [17], and the covariance update for an EKF [93], the conditional entropy for the linearized problem is

$$H(\theta_t | \mathbf{z}_t) = \frac{1}{2} \log \left( (2\pi e)^2 \left| \left( \Sigma^{-1} + \sum_{i=1}^{n_v} J_b^{(i)T} (\sigma_b^2)^{-1} J_b^{(i)} \right)^{-1} \right| \right) \quad (5.3)$$

where

$$J_b^{(i)T} (\sigma_b^2)^{-1} J_b^{(i)} = \frac{1}{\sigma_b^2 (r^{(i)})^2} \begin{bmatrix} \sin^2 \xi^{(i)} & -\frac{1}{2} \sin 2\xi^{(i)} \\ -\frac{1}{2} \sin 2\xi^{(i)} & \cos^2 \xi^{(i)} \end{bmatrix} \quad (5.4)$$

To minimize the uncertainty, given by (5.3), it is equivalent to maximize

$$U_b^{(i)}(\xi, \mathbf{r}) = \left| \Sigma^{-1} + \sum_{i=1}^{n_v} J_b^{(i)T} (\sigma_b^2)^{-1} J_b^{(i)} \right| \quad (5.5)$$

Equation (5.5) provides two insights into the behavior of information-seeking bearings-only sensors.

First, as  $r^{(i)}$  decreases,  $U_b^{(i)}(\xi, \mathbf{r})$  increases—it is beneficial to be close to the target.



This is due to the decreased effect of additive direction noise at close range—a perspective effect noticeable in our own vision. Note that as  $r^{(i)} \rightarrow 0$ , the true Bayesian posterior probability distribution has nonzero uncertainty, whereas linearization error in the EKF causes  $U_b^{(i)}(\xi, \mathbf{r}) \rightarrow \infty$ . If a linear estimator incorporates a measurement made at the mean, the most informative location, the covariance of the estimate becomes singular.

To observe a second insight, consider the case in which values of  $r^{(i)}$  are equal and nonzero for all  $i$ . An analytical solution for optimal values for  $\xi^{(i)}$  can be found when  $\Sigma = \sigma^2 I$ , where  $\sigma$  is the standard deviation in the target state estimate in both axes and  $I$  is a  $2 \times 2$  identity matrix. By taking the gradient of (5.5), the optimal values of  $\xi^{(i)}$  can be shown to be those that satisfy

$$\sum_{i=1}^{n_v} \cos 2\xi^{(i)} = 0 \quad \text{and} \quad \sum_{i=1}^{n_v} \sin 2\xi^{(i)} = 0 \quad (5.6)$$

Thus, two solutions always satisfy (5.6), 1) spacing the vehicles with equal angles of  $\frac{\pi}{n_v}$ , and 2) grouping all vehicles into pairs or triplets that are at  $90^\circ$  or  $60^\circ$ , respectively. For  $n_v > 4$ , a continuum of other solutions exist that achieve minimum conditional entropy. Example optimal configurations for vehicles at equal ranges are shown in Fig. 5.2. Note that  $\cos 2\xi^{(i)} = \cos(2\xi^{(i)} + n\pi) \forall n \in \mathbb{Z}$ , so optimally configured vehicles may be on either side of the target with the same benefit—a consequence of the linearization. Fig. 5.3 shows optimal configurations with  $180^\circ$  randomly added to the solution directions yielding equivalent optimal solutions. The complete optimal solution causes the vehicles to fan out to satisfy the optimal direction criteria, and approach the origin, as a result of the  $\frac{1}{r^{(i)}}$  perspective effect.

The optimal configuration can be numerically found for situations where the uncertainty of the target location is not equal in both directions. For example, as shown in Fig. 5.4, if there is more uncertainty in the  $y$  direction, optimal configurations of bearing sensor tend to cluster about the  $x$ -axis.

To determine the benefit of increasing  $n_v$  for optimally spaced vehicles, (5.3) can be simplified using (5.6), with all vehicles at the same range,  $r^{(i)} = r$ . The conditional

entropy is then

$$H(\theta_t|\mathbf{z}_t) = \frac{1}{2} \log \left( (2\pi e)^2 \left( \frac{1}{\sigma^2} + \frac{n_v}{2\sigma_b^2 r^2} \right)^2 \right) \quad (5.7)$$

As more sensors are added to the network, they increase performance logarithmically. Increasing  $n_v$  is equivalent to proportionally decreasing  $\sigma_b^2$ . The worst case can similarly be computed for the configuration in which all vehicles are at the same angle with respect to the target,

$$H(\theta_t|\mathbf{z}_t) = \frac{1}{2} \log \left( (2\pi e)^2 \frac{1}{\sigma^2} \left( \frac{1}{\sigma^2} + \frac{n_v}{\sigma_b^2 r^2} \right) \right) \quad (5.8)$$

Consider the ratio of the arguments in the log expressions in (5.7) and (5.8). The ratio of the best case configuration to the worst is

$$\kappa = 1 + \frac{n_v^2}{4\sigma_b^4 r^4 p^{-1} \left( \frac{1}{\sigma^2} + \frac{n_v}{\sigma_b^2 r^2} \right)} \quad (5.9)$$

As  $n_v$  increases,  $\kappa$  increases—that is, cooperation has more benefit. As the prior uncertainty  $\sigma$  decreases, so does  $\kappa$ —decreasing the benefit of cooperation as the target is better localized.

The patterns of cooperation given by (5.6) will be apparent in the subsequent results using particle filters, as will the benefits of approaching the expected target location. However, the particle filter will be shown to handle nonlinear effects for measurements near the target, rather than risking divergence due to linearization error. The logarithmic benefit of increasing  $n_v$  is apparent in the Monte Carlo results.

### Particle Filter Results

Bearings-only simulations were run with the particle filter mutual information utility functions to determine the empirical behavior of the algorithms, and to obtain Monte Carlo results. For all bearings-only target search simulations, the search was simulated to take place over a square search region, as shown in Fig. 5.5. A uniform prior probability distribution was used over the search region to initialize each vehicle's

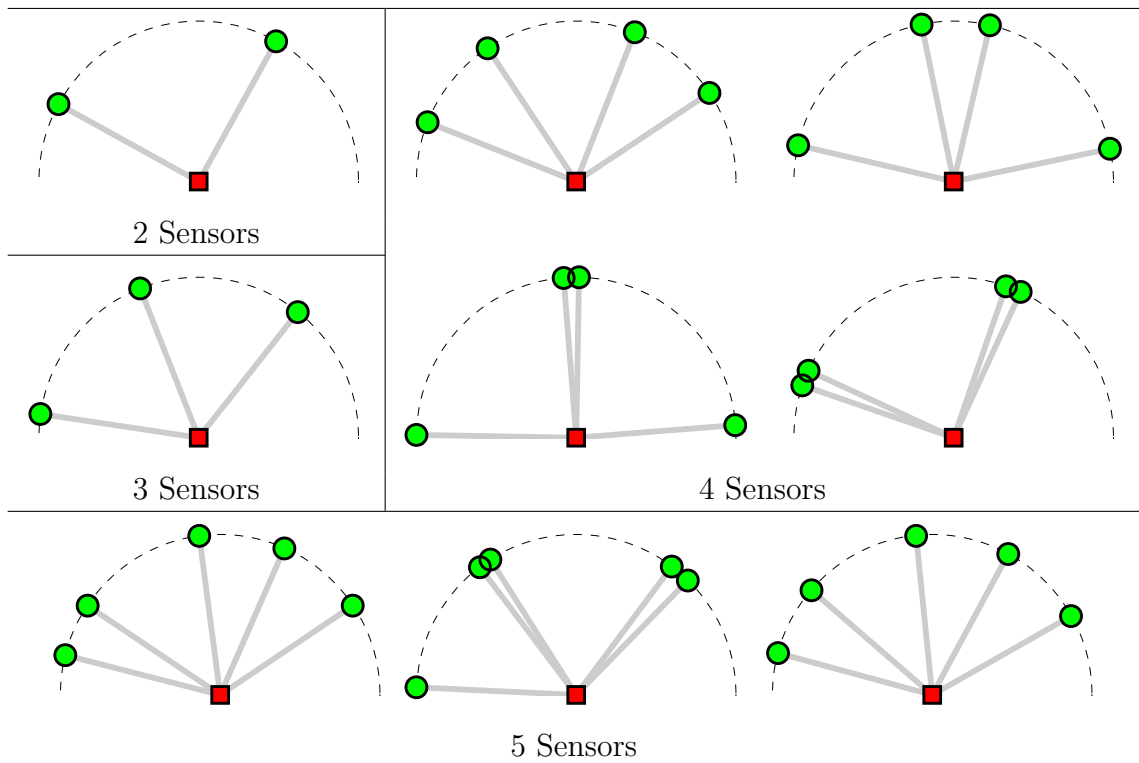


Figure 5.2: Optimal configurations of range sensors using linearized-Gaussian estimators about the mean of the prior distribution of the search target's position (red square) for a prior distribution with equal information in the  $x$  and  $y$  directions.

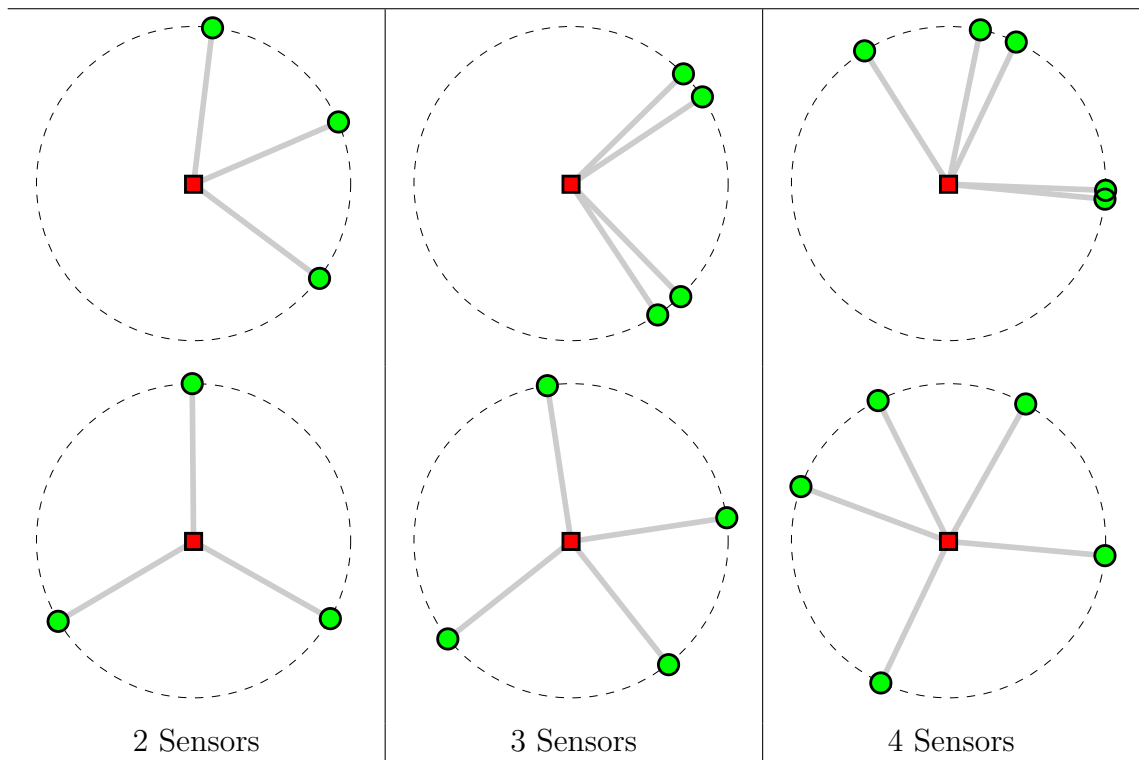


Figure 5.3: Optimal configurations of range sensors using linearized-Gaussian estimators about the mean of the prior distribution of the search target's position (red square) for a prior distribution with equal information in the  $x$  and  $y$  directions. Here, the sensors are randomly flipped to the opposite side to make equivalent observations.

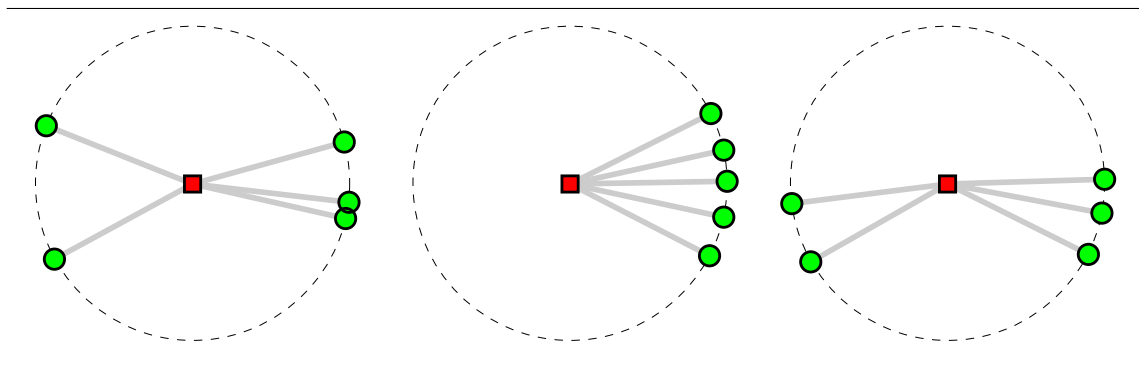


Figure 5.4: Optimal configurations of range sensors using linearized-Gaussian estimators about the mean of the prior distribution of the search target's position (red square) for a prior distribution with 5 times more uncertainty in the  $x$  direction than the  $y$ .

prior probability distribution. This represents the prior knowledge that the target is contained in the region, with complete uncertainty of its location within. The vehicles are modeled as quadrotor helicopters, with the associated dynamic capabilities and constraints. They are further constrained to avoid colliding with one another, as defined by a minimum separation distance.

Empirically, it was observed that the particle filter based algorithms result in the rapid localization of the target, despite complete prior uncertainty over the search region. This demonstrates strong robustness for the active sensing system. As expected from the predictions above, the vehicles, initialized near one another, fan out at first. The mutual information objective function indicates they will gain the most information in this manner. At the same time, the objective function balances the tradeoff of needing to approach the target. This results in the vehicles eventually flying to the target, as predicted from the linear approximation. However, when the vehicles reach the target, the information gain remains finite, unlike the fictitiously high information gain predicted by the linear approximation. They hover near the target as the search converges, continuing to acquire information, and maintaining their minimum separation distances.

These results were consistent through a large number of trial runs, as demonstrated by Monte Carlo results. Sets of 1000 trials were performed for several sizes of mobile sensor networks, with both the single-node and pairwise-node approximations. As shown in Fig. 5.7a, the use of the pairwise-node approximation resulted in a reduced time-to-convergence compared to the single-node approximation, on average. The pairwise-node approximation also yielded more consistent performance compared to the single-node approximation, hence the narrower error bands. This result demonstrates the benefit of considering the effects vehicles have on each other while performing the optimization, rather than relying on emergent behavior for vehicle cooperation.

The result of using the pairwise-node approximation for an increasing number of vehicles is shown in Fig. 5.7b. The particle filter based information-seeking algorithm successfully exploits the additional availability of sensors. The time-to-convergence is reduced, on average, as vehicles are added to the fleet. Next, the use of range-only

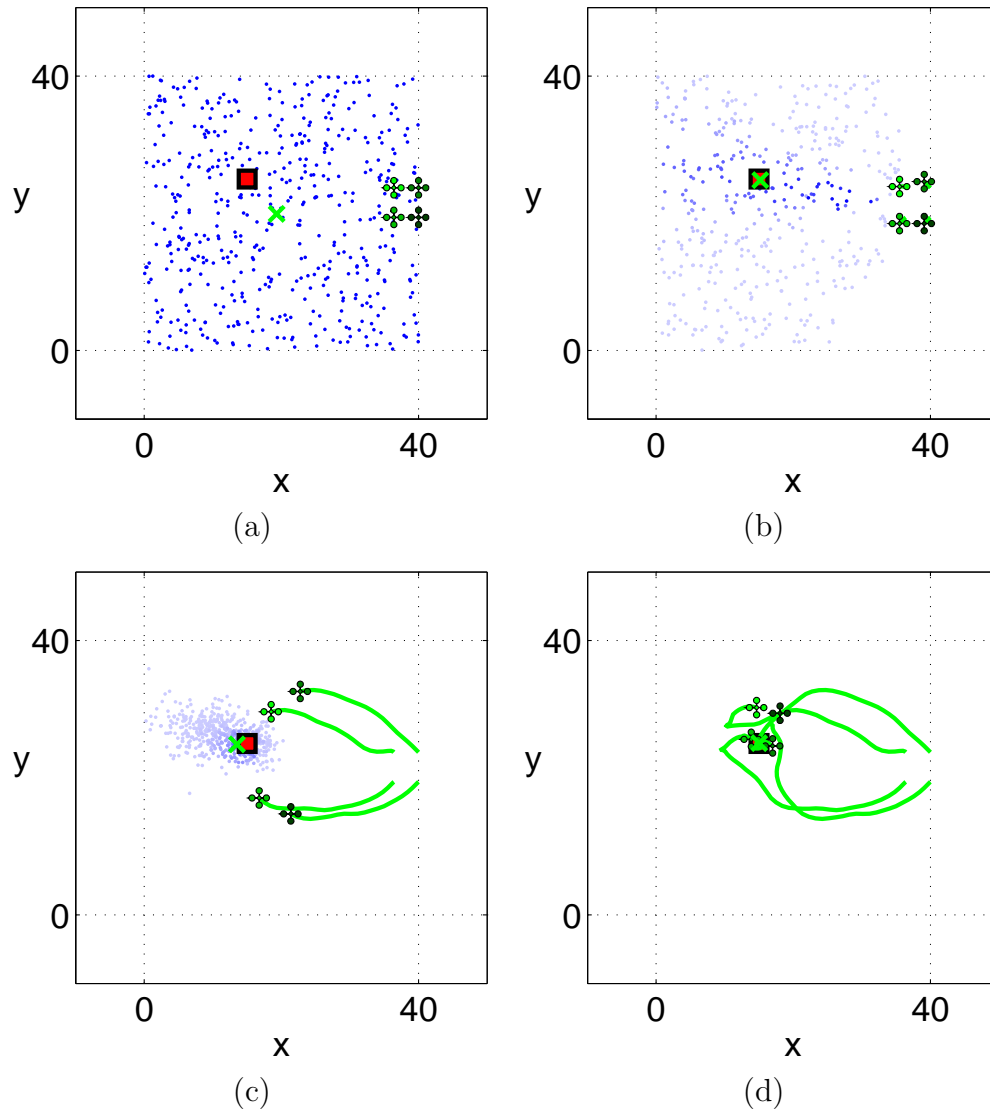


Figure 5.5: Simulation of a bearings-only target search with 4 mobile sensors using the particle filter distribution to compute mutual information and the pairwise-node approximation for distributed control. By directly using the particle filter distribution, there is bias from linearization, as there is in an EKF. The sensor has additive directional noise of  $\sigma = 0.3$  radians, the target's true location is the small square, the MMSE estimate is the X, and the lines behind the vehicles are trajectory histories. The particles, dots, with darkness proportional to their weights, were initialized from a uniform distribution over the search region. Plots (a)-(d) show times 0, 3, 10, and 50. The vehicles spread out and approach the expected target location as predicted in (5.6), only to gain the maximum information possible.

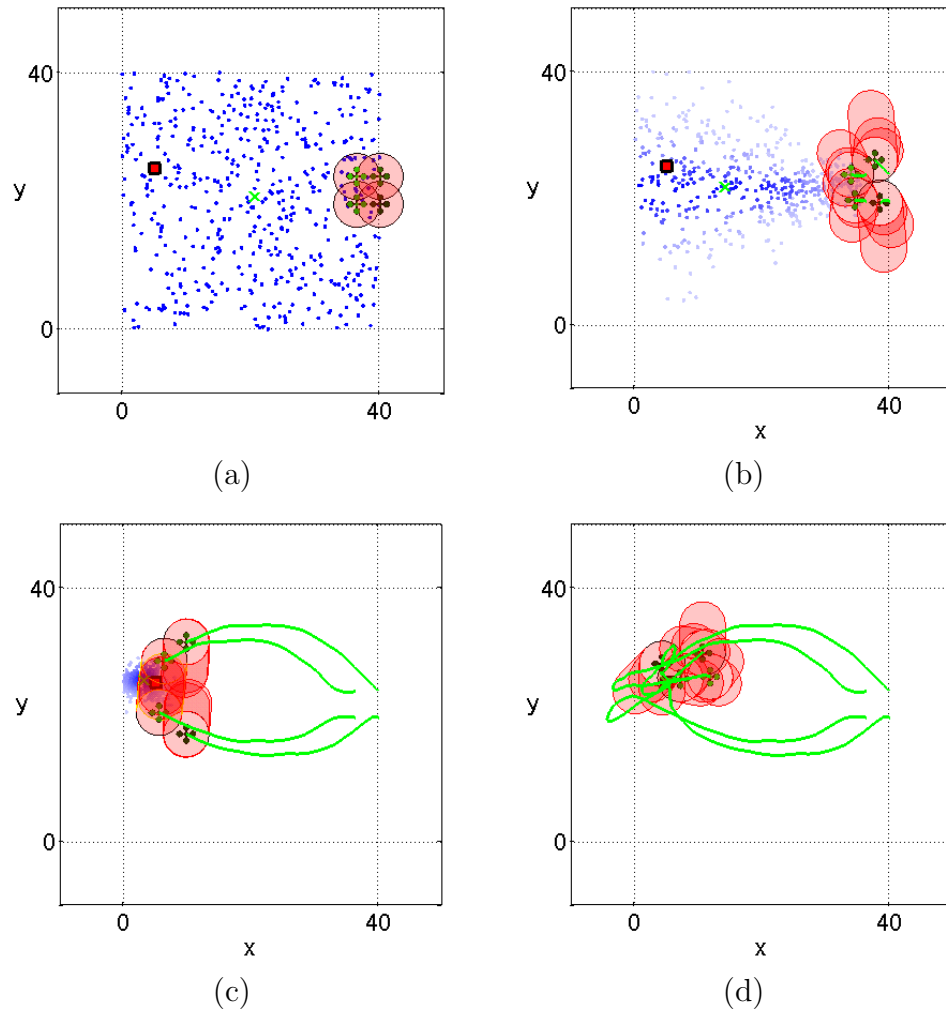


Figure 5.6: Simulation of a bearings-only target search similar to Fig. 5.5. The avoid sets used by the collision avoidance algorithm are shown around the vehicles; in other figures in this section, the sets are omitted from the plots. Experimentally, the optimization was not adversely effected, and run time is much less than the iterative method.

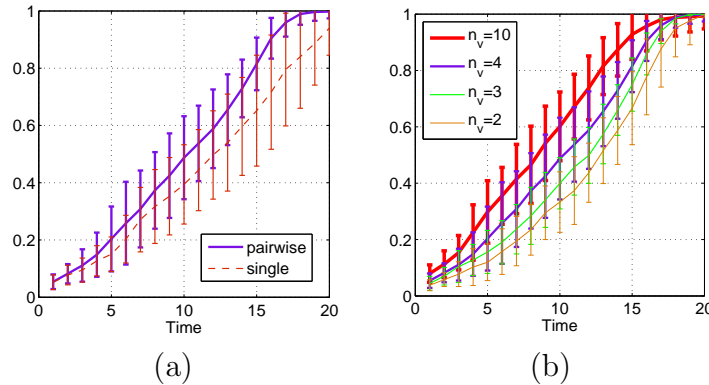


Figure 5.7: Mean and quartile bars of the probability that the true target state is within 1 unit of the MMSE estimate, for sets of 1000 trials of bearings-only target localization. The difference between the single-node and pairwise-node approximations are shown in (a), with  $n_v = 4$ . The single-node approximation is the dashed line, and the pairwise-node approximation is solid line. The pairwise-node results are more predictable and result in better expected performance. The effect of utilizing more sensors is shown in (b), comparing the effect of using the pairwise-node approximation for varying number of search vehicles: 2, 3, 4 and 10.

sensors is analyzed.

### 5.1.2 Range-Only Sensors

For this example, consider sensors that measure the distance to the target, using sensors such as wireless communication devices [39]. It is shown that, in addition to avoiding the problems associated with EKFs described previously, the use of particle filters makes it possible to quantify effects on information gain not possible with a linearized method. Although in the linearized model the optimal range to the target will be shown to be inconsequential, the particle filter information formulation demonstrates that there exists an optimal range, due to minimum and maximum ranges for the sensor, that cannot be captured by the linearized model.



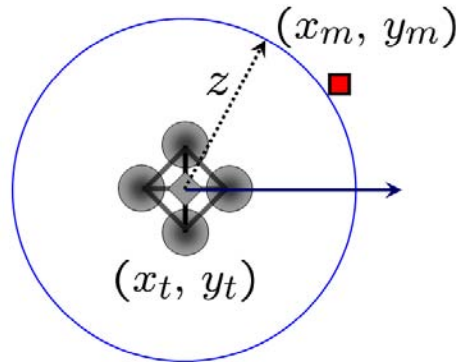


Figure 5.8: Range-only measurement model where  $z$  is a measurement of the distance from the position of the sensor  $(x_t^{(i)}, y_t^{(i)})^T$  to the position of the target  $(x_m^{(i)}, y_m^{(i)})^T$ . It differs from the true direction due to additive noise, given in (5.10). An example of such a measurement is the time of flight of wireless communication signals.

### Measurement Model

Again, consider searching for a target in the  $xy$ -plane, with the same states as the previous, bearings-only, example. The range measurement model is

$$h_r^{(i)}(\mathbf{x}_t^{(i)}, \theta, \eta_t^{(i)}) = \sqrt{(x_m - x_t^{(i)})^2 + (y_m - y_t^{(i)})^2} + \eta_t^{(i)} \quad (5.10)$$

where  $h_r$  is the model of the range measurement with noise, as shown in Fig. 5.8, and  $\eta_t^{(i)} \sim \mathcal{N}(0, \sigma_r^2)$  is the measurement noise. Although any measurement model could be used, such as noise proportional to range, due to clock drift, the additive noise model is chosen to permit comparison with other work (e.g., [15]). A key characteristic of this sensor is the lack of directional information. Hence, a single measurement provides an axisymmetric probability distribution of potential target locations.

### Predicted Behavior

To gain insight into the behavior of the optimally controlled system, again consider the case of an accurately localized target, with a probability distribution that can be approximated as Gaussian, with a covariance matrix equal to a scaled identity matrix. In this condition, the optimal placement of the sensors can be solved for, as

was done for the bearings-only sensor in Section 5.1.1. The Jacobian of the sensor model is

$$J_r^{(i)} = \begin{bmatrix} \cos \xi^{(i)} & \sin \xi^{(i)} \end{bmatrix} \quad (5.11)$$

where  $\xi$  is the angle from the  $i^{\text{th}}$  vehicle the mean of the target estimate, as in Section 5.1.1. The uncertainty following a sensing action is

$$H(\theta_t | \mathbf{z}_t) = \frac{1}{2} \log \left( (2\pi e)^2 \left| \left( \Sigma^{-1} + \sum J_r^{(i)\text{T}} (\sigma_r^2)^{-1} J_r^{(i)} \right)^{-1} \right| \right) \quad (5.12)$$

where  $\Sigma$  is the covariance of the target state distribution. Unlike the bearings-only sensor, the posterior uncertainty is not a function of range, in this linearized case. Therefore, only the direction from the target to the vehicles need be optimized. Numerical solutions show an expected behavior—the vehicles tend to cluster near the elongated axis of the confidence ellipse corresponding to any posterior distribution.

If the posterior distribution has equal uncertainty in the  $x$  and  $y$  directions, then the optimal angles for the bearings-only sensors, given by the conditions of (5.6), are also the optimal angles for range-only sensors. Either sensor type provides measurements that can be used to “triangulate” a measurement—they are simply providing measurements rotated by  $90^\circ$  from each other—though there is no scaling due to perspective for a range-only sensor. However, as is seen in the analysis using particle filters, which capture the effect of saturation of the sensor and the curvature of the range measurements, the optimal range of the range-only sensors is, in fact, important.

### Particle Filter Results

Range-only simulations were run with the particle filter mutual information utility function and pairwise-node approximation to determine the empirical behavior of the algorithms, and evaluate how the mutual information utility of measurements vary with sensing locations. For all range-only target search simulations, the search scenario was the same as for the bearings-only simulations described in Section 5.1.1.

The particle filter based algorithms again result in the rapid localization of the

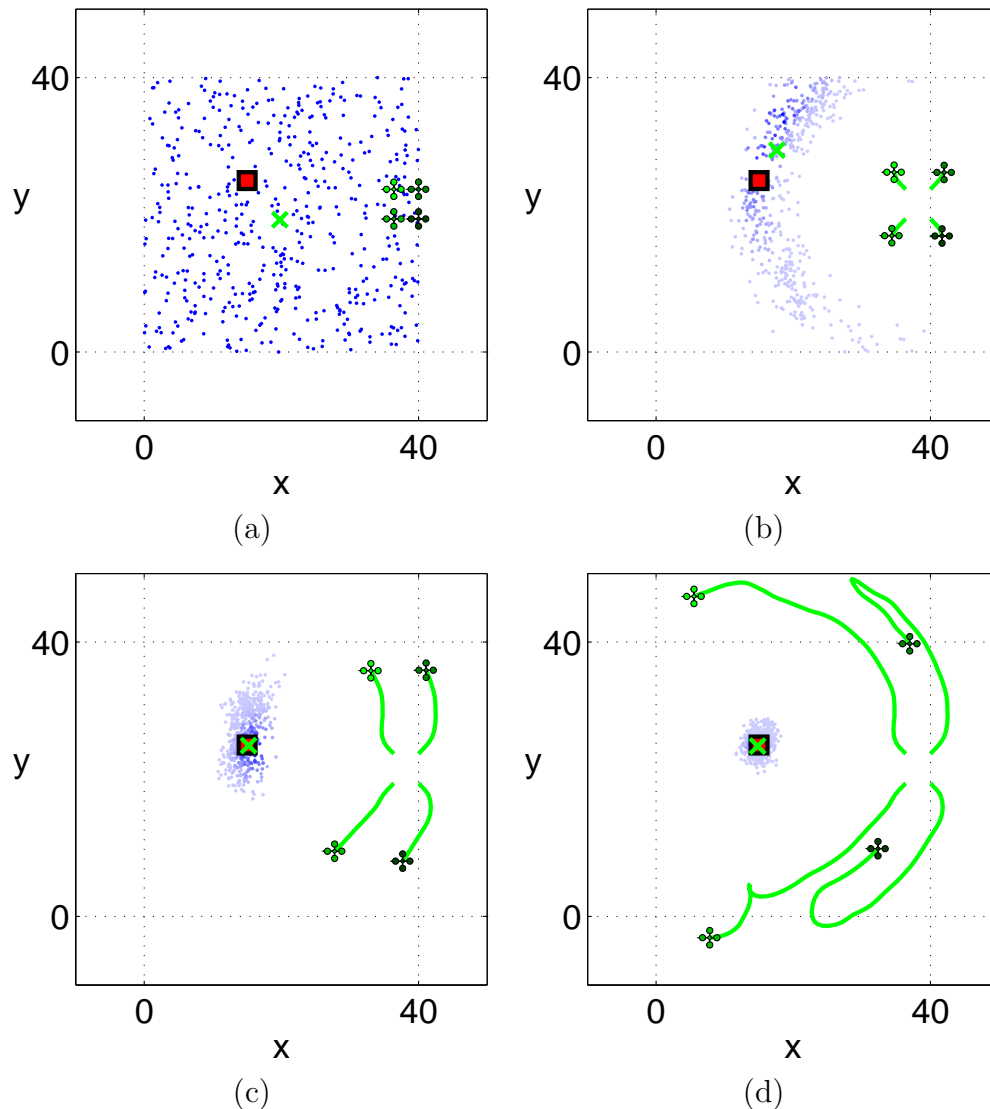


Figure 5.9: Range-only target localization with 4 mobile sensors (quadrotor helicopters) using the particle filter distribution to compute mutual information and the pairwise-node approximation for distributed control. The sensor has additive noise of  $\sigma = 5$  units. The plotted particles, etc., are as described in Fig. 5.5. Plots (a)-(d) show the results for times 0, 5, 15, and 50. The vehicles spread out along the ring of particles, and then fan out at optimal distances to the target, according to the minimum and maximum range of their sensors. Approaching the target would be a sub-optimal solution for these sensors.

target, despite complete prior uncertainty over the search region, as shown in a typical result in Fig. 5.9, using four vehicles with the pairwise-node approximation. Again, the consistent ability of the mobile sensor network to localize the target demonstrates the robustness of the proposed methods. As expected from the predictions above, the vehicles fan out. The mutual information objective function indicates they will gain the most information in this manner. However, unlike bearings-only scenario, there is no advantage to approaching the target. As a result, the vehicles circle the target at a standoff distance.

Upon further inspection, it is observed that the standoff distance the vehicle converges to is a consequence of considering the nonlinear sensor characteristics. The differences between the prediction using the linearized approximation and the more accurate particle filter method are highlighted by the comparison in Fig. 5.10. These differences are due to sensor nonlinearities that are eliminated in the linearization step.

The nonlinearities arise from several sources. First, there is saturation of the range-only sensor; the range sensor cannot measure a range less than zero. A typical range sensor has a finite minimum and maximum range. In the example shown in Fig. 5.10, the sensor is limited to measurements of 0 to 56 units. Additionally, there are effects of the structure of the distribution, which may be ring-like or multimodal, that are captured by the particle filter mutual information objective function, but cannot be captured by the linearized method. Finally, there is the effect of curvature when interpreting range measurements. This cannot be captured using a linearized method, but is automatically included by the particle filter method. Now, consider a third and final sensor, rescue beacons.

### 5.1.3 Magnetic Dipole Sensors (Rescue Beacons)

For this example, consider a sensing modality for which EKF's are prone to failure—sensing the avalanche rescue beacon of a victim buried in snow due to an avalanche. The beacon uses a modulated magnetic dipole with a field that can be measured by beacon receivers. Both the position and orientation of the beacon are unknown,

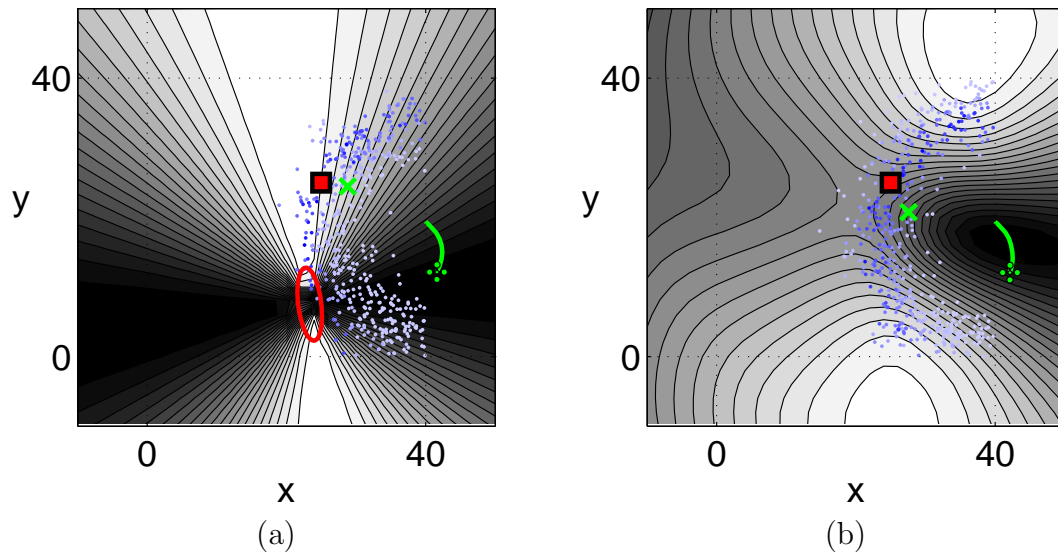


Figure 5.10: Using the particle filter to directly compute available mutual information captures effects not possible with a linear Gaussian approximation, such as the saturation of measurements at the near and far limits of the sensor’s range. The mutual information contours, for moving to any point at the subsequent time step, are shown (a) using a linear Gaussian approximation, with an EKF, versus (b) using a particle filter. The brighter the contour, the more information is available. The  $1 - \sigma$  ellipse of the EKF is shown in (a). In both examples, the vehicle is controlled using the particle filter mutual information, leading to similar trajectories. The plotted particles, etc., are as described in Fig. 5.5.

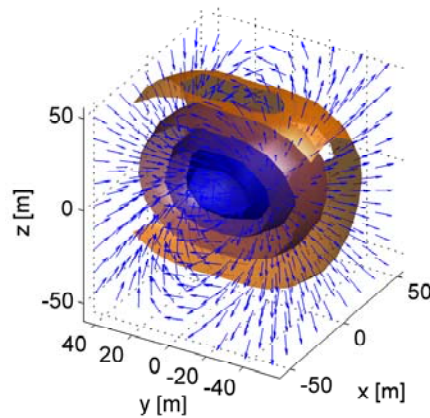


Figure 5.11: Rescue beacon magnetic field, as given in (5.13). A cross-section field strength isosurfaces is shown, with arrows depicting local magnetic field orientation. The transmitter antenna is at the origin, with its antenna parallel to the  $y$ -axis.

adding complexity beyond the sensors of the previous sections. The search is currently performed by individual rescuers using a rehearsed search pattern—a complex activity requiring professional training to be effective [5]. A rapid localization of the victim is essential—in one study, odds of survival were 92% for victims unburied within 15 minutes, but dropped to 30% after 35 minutes [66].

The use of particle filter techniques is demonstrated to estimate the posterior probability distribution and control the vehicles, directly using the particle filter distribution, such that they maximize the rate at which they acquire information about the victim’s location. The expected behavior is derived in limited situations by linearizing the measurement model. Due to the periodic domain, the information in the linearized model is only accurate when the target is well localized. This enables validation of the particle filter implementation under that circumstance, toward the conclusion of the search. The particle filter methods will be shown to handle the automatic acquisition of information during all stages of the search.

### Measurement Model

The rescue beacon system uses measurements of the magnetic dipole emitted by a loop antenna modulated at 457 kHz, a frequency that penetrates snow and water,

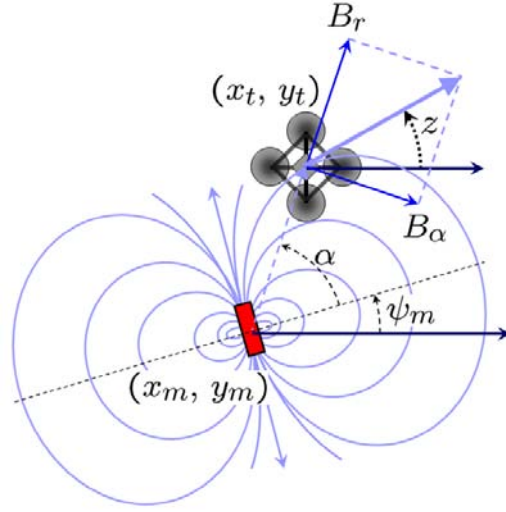


Figure 5.12: Rescue beacon measurement model, in two dimensions, with the transmitter antenna axis lying along the  $xy$ -plane at  $(x_m, y_m)$  with orientation  $\psi_m$ . The measurement  $z$  is the local orientation of the magnetic field vector in the plane of the receiver, with additive noise, given by (5.14). The field has components  $B_r$  and  $B_\alpha$  from (5.13). The mobile sensor, at time  $t$ , is at position  $(x_t, y_t)$ .

and is not reflected by rock [38]. The magnetic field of a modulated electromagnetic source  $B : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  is derived in [48], and shown in Fig. 5.11. Given the modulation frequency and range of rescue beacons, the near-field formula for the magnetic field is appropriate [48, 38]. Measurements are made of the toroidal near-field at spherical coordinates  $(r^{(i)}, \phi^{(i)}, \alpha^{(i)})$  with respect to the antenna, where  $r^{(i)}$  is the range,  $\phi^{(i)}$  is the rotation angle about the axis of the antenna, and  $\alpha^{(i)}$  is the elevation angle from the plane of the antenna loop, in the right-hand sense, as shown in a two dimensional cross-section in Fig. 5.12. The magnetic field is [48],

$$\mathbf{B} = \frac{m}{2\pi (r^{(i)})^3} ((2 \sin \alpha^{(i)})\mathbf{e}_r - (\cos \alpha^{(i)})\mathbf{e}_\alpha) \quad (5.13)$$

where the magnitude of the dipole moment is  $m = I_0 \pi r_a^2 n_w$ ,  $I_0$  is the amplitude of the antenna loop current,  $r_a$  is the radius of the antenna loops,  $n_w$  is the number of windings, and  $\mathbf{e}_r$  and  $\mathbf{e}_\alpha$  are unit vectors of the spherical coordinate frame in the positive  $r$  and negative  $\alpha$  directions, respectively. The measurement  $z$  made by the

sensor is the orientation of the magnetic field line's projection onto the plane in the receiver containing two orthogonal receiver antennae. The angle is computed using the arctan of the ratio of the measurements on each axis. To reduce the effects of nonlinearity in signal processing, the receiver can be actively oriented such that the signal strength on each receiver antenna is equal. The receiver orientation is then the field line measurement.

For purposes of this simulation, consider a search for a rescue beacon in two dimensions, with its axis known to lie in the horizontal plane, with unknown heading angle  $\psi_m$ . The state of the target is  $\theta = (x_m, y_m, \psi_m)^T$ , a three dimensional state. Note that it is simple to include the target's altitude and pitch to solve the true problem using the particle filter framework, but the three degree of freedom model used in this section provides more easily visualized results. The measurement equations can then be written in terms of the magnetic field direction at a receiver,

$$h_a^{(i)}(\mathbf{x}_t^{(i)}, \theta, \eta_t^{(i)}) = \xi^{(i)} - \arctan(2 \cot(\alpha^{(i)})) + \eta_t^{(i)} \quad (5.14)$$

where  $h_a$  is the modeled value of the noisy magnetic field orientation measurement  $z$  as shown in Fig. 5.12,  $\alpha^{(i)} = \xi^{(i)} - \psi_t$ , and  $\eta_t^{(i)} \sim \mathcal{N}(0, \sigma_a^2)$  is the measurement noise.

### Predicted Behavior

To gain insight into the behavior of the optimally controlled system, again consider the case of an accurately localized target, with a probability distribution that can be approximated as Gaussian, with a covariance matrix equal to a scaled identity matrix. In this condition, the optimal placement of the sensors can be analytically solved for, as was done for the other sensors in previous sections. The Jacobian of the sensor model is

$$J_a^{(i)} = \frac{1}{3(\lambda^{(i)})^2 + (r^{(i)})^2} \begin{bmatrix} \frac{-\sin(\xi^{(i)})}{r^{(i)}} \left( 3(\lambda^{(i)})^2 - (r^{(i)})^2 \right) \\ \frac{\cos(\xi^{(i)})}{r^{(i)}} \left( 3(\lambda^{(i)})^2 - (r^{(i)})^2 \right) \\ 2(r^{(i)})^2 \end{bmatrix}^T \quad (5.15)$$



where  $\lambda^{(i)}$ , the lateral distance between the mean of the estimated antenna axis and the measurement point, is

$$\lambda^{(i)} = r^{(i)} \cos \alpha^{(i)} \quad (5.16)$$

The optimal sensing utility function can be found using (5.5) by replacing  $J_b^{(i)}$  with  $J_a^{(i)}$ . High prior uncertainty in orientation yields a utility function similar to a range-only sensor, with the optimal relative angle being along the axis of the sensor. Low prior uncertainty in orientation yields a utility function similar to a bearings-only sensor. Unlike the bearings-only or range-only sensors, few additional generalizations can be drawn from the linearized model, due to its complexity. The mutual information contours for the linearized results were compared to those for the particle filter methods. Although the results match for low uncertainty, unimodal posterior probability distributions, they were found to vary substantially for more typical particle distributions encountered during simulated searches, with multiple modes, and high uncertainty.

Rescue beacon simulations were run with the particle filter mutual information utility function and the pairwise-node approximation to show the empirical behavior of the algorithms, and evaluate how the mutual information utility function evolves as the problem converges. For all rescue beacon search simulations, the search scenario was the same as in Section 5.1.1. In addition to the search target having a position, it also has an estimated orientation. To simplify presentation, the simulated searches were performed using three degrees of freedom (two spatial and one directional), as in (5.14).

As shown in Fig. 5.13, the proposed method quickly localizes the target. At first, the vehicles fan out. They proceed to move to locations that reinforce one another's measurements. The behavior is substantially more complicated than that required for range or bearing sensors. The posterior distribution, visualized by the particles, demonstrates the ability of this method to handle complicated posterior beliefs. It successfully exploits the structure of the probability distribution to reduce uncertainty. The four vehicles cooperate in a distributed, computationally efficient manner.

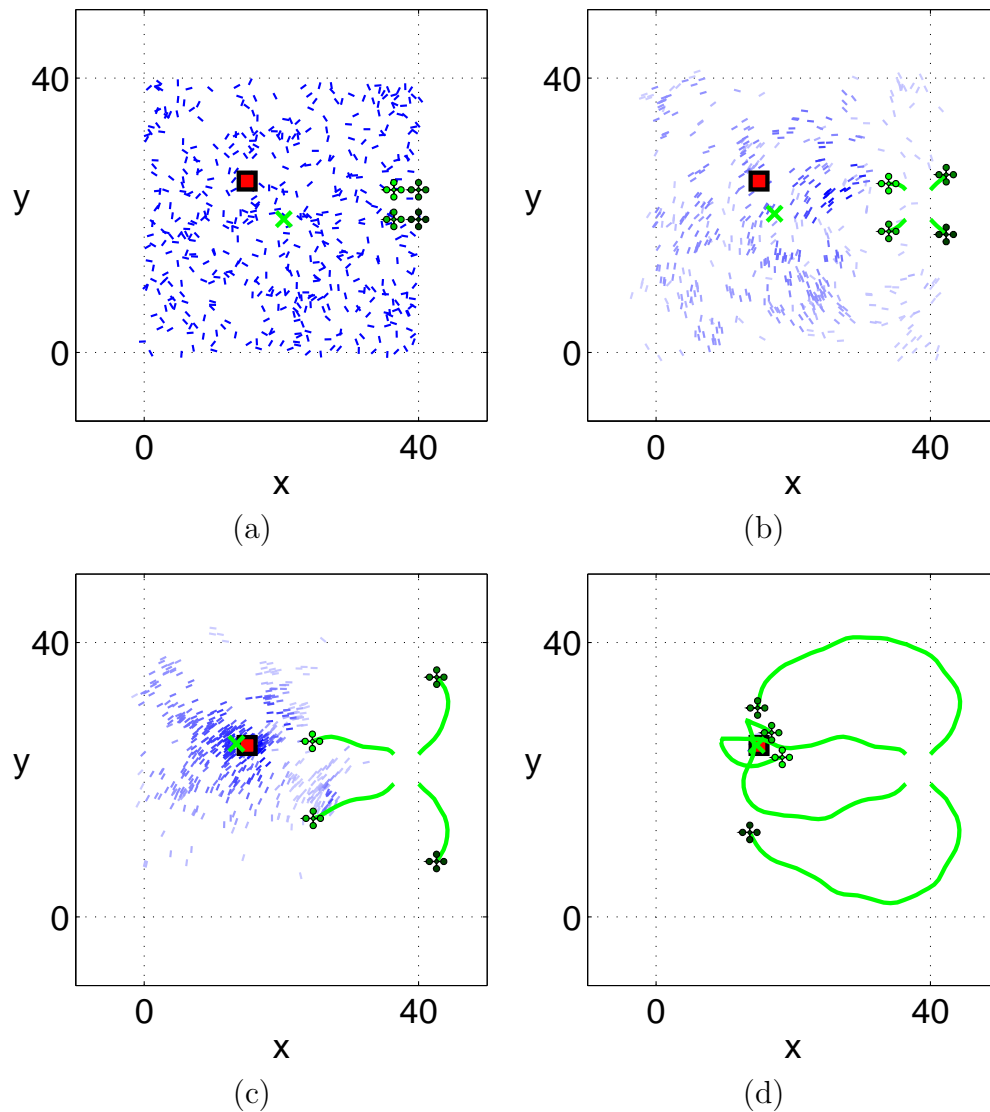


Figure 5.13: Rescue beacon localization with 4 mobile sensors using the particle filter distribution to compute mutual information and the pairwise-node approximation for distributed control, simulating the search for a victim buried in an avalanche. The target's true location is the small square ( $\psi_m = \pi/4$  radians), the MMSE estimate is the X, and the lines behind the vehicles are trajectory histories. The sensors measure the local magnetic field line orientation with an additive noise of  $\sigma = 0.7$  radians. The particles, short lines with darkness proportional to their weights, were initialized from a uniform distribution over the search region. Plots (a)-(d) show times 0, 5, 15, and 50. The vehicles fan out and then approach the expected target location, only to gain more information about its location.

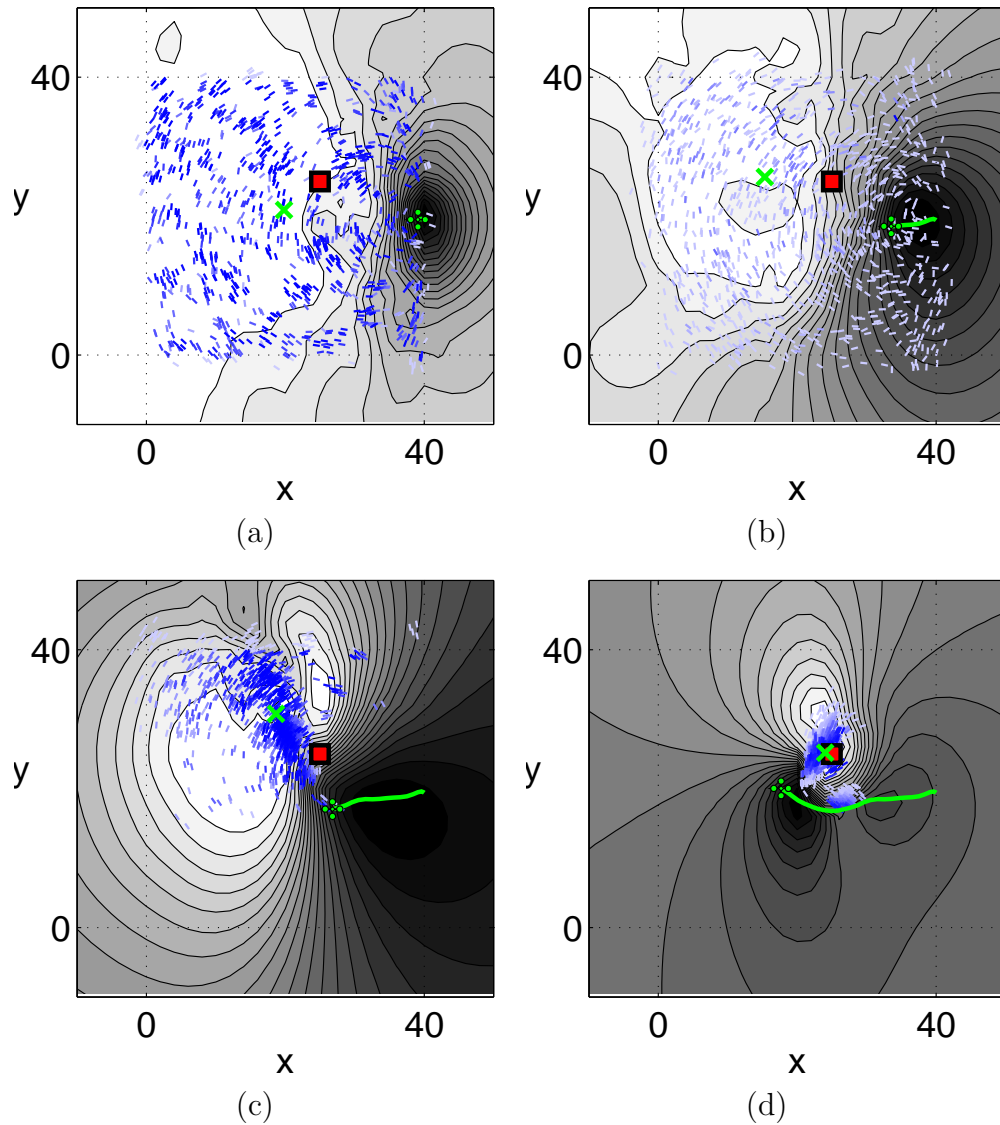


Figure 5.14: The evolution of the available mutual information is shown for a measurements from any point. The mobile sensor is localizing a rescue beacon using the exact mutual information utility function. By using the particle filter to compute available mutual information, nonlinear and non-Gaussian effects are captured, such as the spatially varying orientation in the posterior distribution and multiple modes. The mobile sensor's control input moves the vehicle to the location with maximal mutual information, subject to constraints. The plotted particles, etc., are as described in Fig. 5.13. The measurement has additive noise of  $\sigma = 0.3$  radians. Plots (a)-(d) show times 3, 9, 16, and 26.

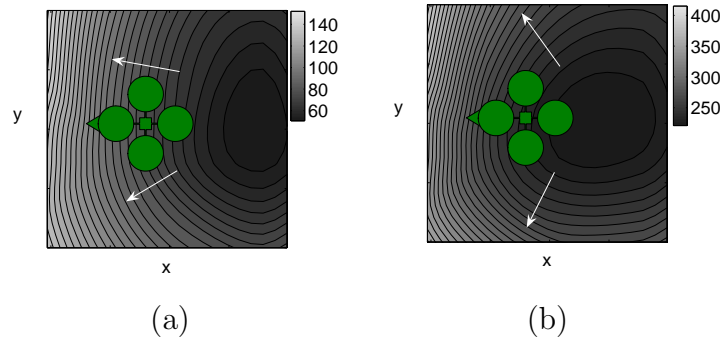


Figure 5.15: Comparison of contours of mutual information that a second vehicle would obtain for making a measurement from that point, using (a) the single-node approximation versus (b) the pairwise-node approximation. The brighter the contour, the more mutual information is available for sensing from that location. Using the single-node approximation, the second vehicle would tend to move along side the first vehicle. Using the pairwise-node approximation, the second vehicle would prefer to fan out. The arrows indicate the preferable directions of travel from potential current positions of the second vehicle. The scenario shown is a rescue beacon search, following actions from time step 2.

To visualize the optimization being performed onboard the vehicles, the contours of the mutual information objective function can be plotted for an observation from any point. The result for one vehicle is shown in Fig. 5.14. The vehicle initially is driven away from where the initial measurements were made, as to not make redundant measurements. The low region for mutual information, in the simulated scenario, follows the direction of the field line that was already measured—maximum information can be gained by initially moving orthogonally to the measured field line. Note that this differs from a common method of trained rescuers, who follow the field line direction to compensate for a lack of geo-referenced measurements. As the search progresses, and the particle set gains more structure, sometimes multimodal, the contours evolve guiding the vehicle to the best available measurements.

The effect of the pairwise-node approximation versus the single-node approximation can also be visualized using a contour plot, as shown in a zoomed in view in Fig. 5.15. The contours depict the mutual information utility function for placement

of a second vehicle, given that a vehicle exists in the location shown. The single-node approximation is not effected by the existing vehicle, whereas the pairwise-node approximation leads to the cooperative behavior of the vehicles fanning out, as appropriate for the depicted rescue beacon search.

Having thoroughly examined the characteristics of the mobile sensor guidance algorithms using information theoretic control, the algorithms are next demonstrated in flight experiments of the STARMAC quadrotor helicopters.

## 5.2 Flight Experiments

This section presents experimental results from the implementation of the proposed information theoretic control laws on STARMAC quadrotor helicopters. They are instrumented with avalanche rescue beacon receivers to search for a rescue beacon “lost” in the field. This section proceeds by presenting the vehicle software and system configuration, with details of how the sensors are used, and then gives the experimental results.

### 5.2.1 Vehicle Software

The autonomous guidance software for STARMAC is *FlyerBrain*, diagrammed in Fig. 5.16. Onboard the aircraft, *FlyerBrain* interfaces with the inner loop vehicle control system running on the Robostix, with the GPS receiver, with the ground station, and with *FlyerBrain* instances on other aircraft. It stores log data onboard the aircraft. *FlyerBrain* processes several data streams: it computes the GPS solution, as described in Section 2.2.2, processes rescue beacon raw measurements, runs the vehicle state EKF, and estimates the rescue beacon location using all measurements it has made or received from other vehicles. *FlyerBrain* also runs the optimization algorithm presented in Chapter 3 to compute reference commands to send to the inner loop control system. The real time control module provides several modes of control, including hover, trajectory tracking, and relaying reference commands from the optimization algorithm or from the ground station.

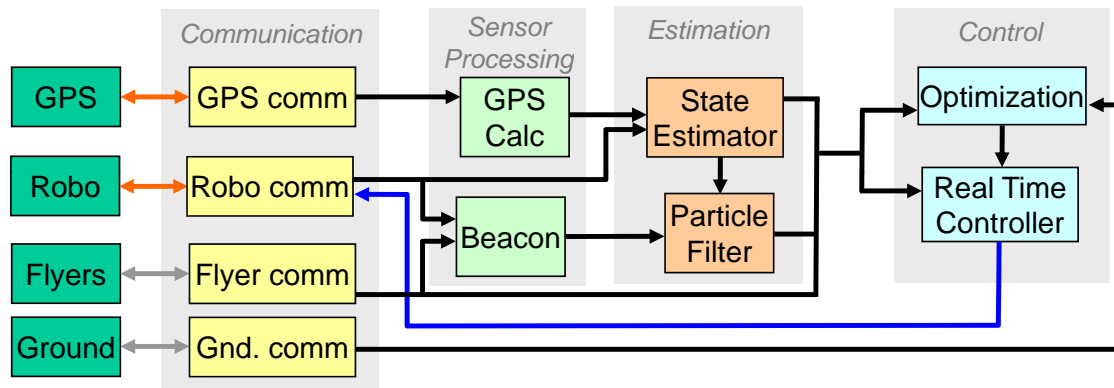


Figure 5.16: Control software for STARMAC, *FlyerBrain*. Each block in a gray area is a module with a dedicated thread. Communication occurs with the GPS receiver, Robostix (“Robo”), other aircraft (“Flyers”), and the ground station. The guidance algorithm is run in the optimization block. When the search algorithm executes, the real time controller switches from hovering to relaying the output of the guidance algorithm.

The software is a C++ program is written to run in Linux on either type of higher level computing platform flown on STARMAC: the Gumstix or the PC104s (described in Section 2.2.3). Most components of *FlyerBrain* can execute in real-time on the Gumstix, though because it lacks a floating point unit, the particle filter and optimization algorithm have only been used on the PC104. *FlyerBrain* uses POSIX threads for managing real-time execution to minimize the time required for context switches. Software modules are interfaced using a data port template with queues using mutexes. Data flow with this paradigm is either many-to-one or one-to-many.

Communication between vehicles and with the ground station uses UDP over an 802.11g network. The software is operated through a remote SSH session. The onboard computers send flight data to a laptop computer ground station, and can receive commands from the ground station, such as joystick controlled attitude and altitude commands or experiment start/pause/stop commands.

The particle filter used a 6 degree of freedom particle filter with 20,000 particles to estimate beacon position and the orientation unit vector. The measurement model was the 6 degree of freedom version [48] of the model used in Section 5.1.3, the near

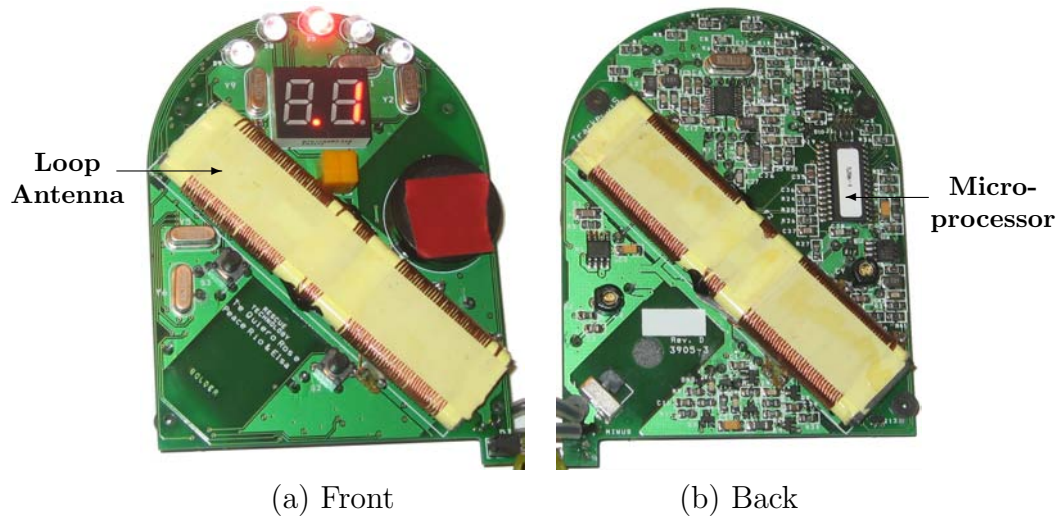


Figure 5.17: Tracker DTS Avalanche Rescue Beacon Transceiver circuitry. While in transmit mode, the back antenna transmits 457 kHz pulses. In receive mode, analog measurements from the diagonal antennae on the front and back are multiplexed into pin 2 of the microprocessor, controlled by digital pins 22 and 23.

field of a magnetic dipole.

$$\mathbf{B} = \frac{m}{4\pi\|\mathbf{r}^{(i)}\|^3} \left( 3 \frac{\mathbf{m} \cdot \mathbf{r}^{(i)}}{\|\mathbf{r}^{(i)}\|^2} \mathbf{r}^{(i)} - \mathbf{m} \right) \quad (5.17)$$

where  $\mathbf{m}$  is the unit orientation vector of the transmitting beacon,  $m$  is again the dipole moment, and  $\mathbf{r}^{(i)}$  is the relative position of the receiver with respect to the transmitter. The measurement model assumes additive Gaussian noise on the measured field vector, projected into the plane of the receiver.

### 5.2.2 System Configuration

This experiment uses off-the-shelf rescue beacons, the Tracker DTS. These are *transceivers*; they have a transmit mode and receive mode. In receive mode, they use two antennae, discussed in Section 5.1.3 to measure the local orientation of the field line of a beacon in transmit mode. The Tracker DTS uses the standard frequency for avalanche rescue beacons, 457 kHz, a frequency with low absorbance by water and

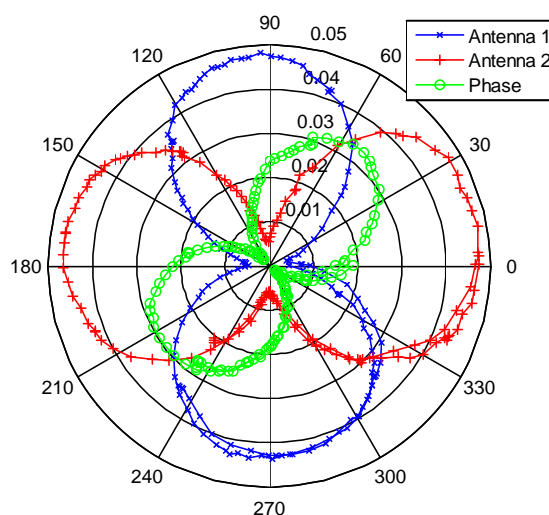


Figure 5.18: Measured received signal strength from a Tracker DTS avalanche rescue beacon. The raw measurements were scaled and exponentiated. The measurements of the two antennae are characteristic of a loop antenna. The third measurement is the amplitude of the difference of the two antenna signals—an indicator of signal phase that disambiguates the field line orientation.

with low reflectance by rock.

One rescue beacon flies on each aircraft. The beacons have no external interface, so they are connected by soldering wires to pins on their microprocessor, shown in Fig. 5.17. By monitoring the state of pins 22 and 23, the state of the digital multiplexer for the analog input can be determined. The analog signal is connected to pin 2. External power is provided using the 5 V supply from the electronics interface board. To convert the power source to one more similar to the batteries being replaced (3 AA batteries in series), a diode with a 0.6 V drop is used to reduce the input voltage.

Three analog signals are measured. Each signal is the log of the received signal strength (RSSI). Two signals are for the two perpendicular antennae. The third measurement is the amplitude of the difference of the two antenna signals—an indicator of signal phase that disambiguates the field line orientation. The raw signals, for both antennae and the phase signal,  $n_1$ ,  $n_2$ , and  $n_p$ , are processed to recover the actual



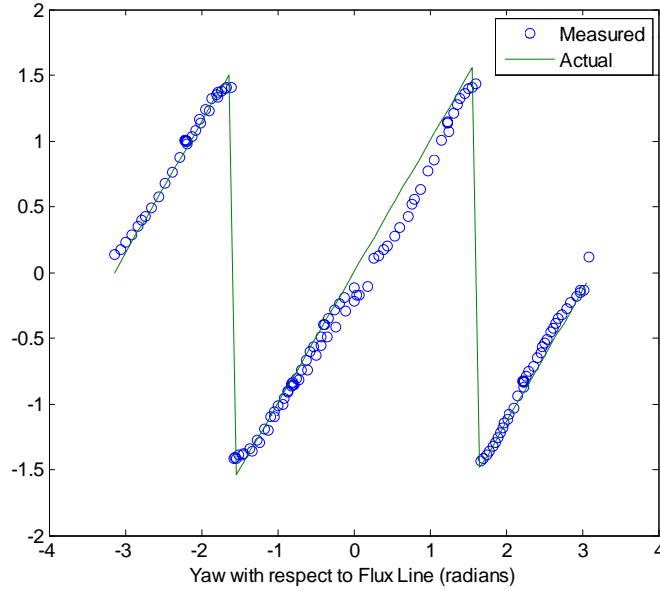


Figure 5.19: Comparison of yaw angle estimate using the IMU to yaw angle estimate from the processed rescue beacon measurements.

signal strength,

$$\begin{aligned}\bar{n}_1 &= \exp\left(\frac{n_1}{s_b}\right) \\ \bar{n}_2 &= \exp\left(\frac{n_2 + o_1}{s_b}\right) \\ \bar{n}_p &= \exp\left(\frac{n_p + o_2}{s_b}\right)\end{aligned}$$

where  $o_1$  and  $o_2$  are calibration offsets for different signals, and  $s_b$  is a scaling constant such that  $e$  is the correct base for the logarithmic signal. The phase is then normalized by dividing by  $\sqrt{\bar{n}_1^2 + \bar{n}_2^2}$  to yield a value that can be tested to disambiguate orientation. The results for rotating the receiver in a full circle are shown in Fig. 5.20. The cutoff point for the phase signal was found by calibration. The  $\arctan2$  function is then used to compute the measured orientation of the field line in the plane of the receiver, with the phase signal result used to change the sign of the



Figure 5.20: A STARMAC quadrotor helicopter with a avalanche rescue beacon transceiver.

arguments accordingly. The result is shown in Fig. 5.19.

The avalanche rescue beacon is mounted to the aircraft on a fiberglass honeycomb plate, shown in Fig. 5.20. The mounting location was found to experience minimal interference due to the electric motors. Field testing this configuration demonstrated that although the motors did not adversely effect the maximum range, when the signal was sufficiently weak, the orientation measurement was meaningless. Further, electrical noise in the surroundings caused spurious measurements when no beacon was transmitting. For instance, an electrical generator 50 m away caused false measurements. To some extent, the meaningless and spurious measurements were eliminated by accepting measurements only above an acceptable signal strength.

### 5.2.3 Flight Tests

Flight experiments were flown with a single quadrotor helicopter and with a pair of quadrotor helicopters. The optimization was executed at 10 Hz to replan considering the effects of control inaccuracy and wind disturbances. Note that this is an information-seeking algorithm, hence frequent replanning does not suffer the same drift issues that a standard model predictive control scheme has for tracking from the current state. The particle filter executed when a measurement was made or received from another vehicle. Computation time was 20-30 ms.

Two optimization solving methods were used. One was an open-source package, IPOPT, an interior point optimization package [99]. Solution time typically matched the performance of Matlab's *fmincon* function. However, it was found that superior results were obtained by using an alternative method, an exhaustive search over the control input space. This method requires fewer iterations, and handles the non-convexity of the objective function. The optimal value was typically better than that found by IPOPT, and execution time was typically 30-70 ms. Note that the control input was constrained to be no more than  $10^\circ$  tilt in any direction. The details of each set of experiments follow.

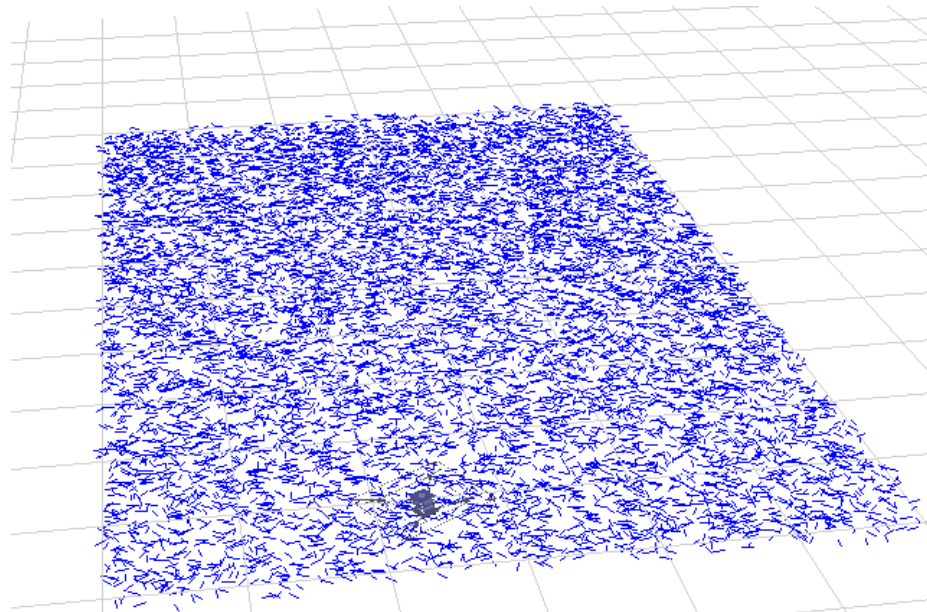
### Single Vehicle Search

The single vehicle flight experiments performed an automatic search for a rescue beacon in a  $6 \times 10$  m field. The particles were initialized uniformly over this field, with a depth of 0.2 m, and a uniform distribution on orientation. Using a single vehicle, the mutual information objective can be directly computed. To contain the experiment within the available space, the objective function included a quadratic penalty for leaving the search area.

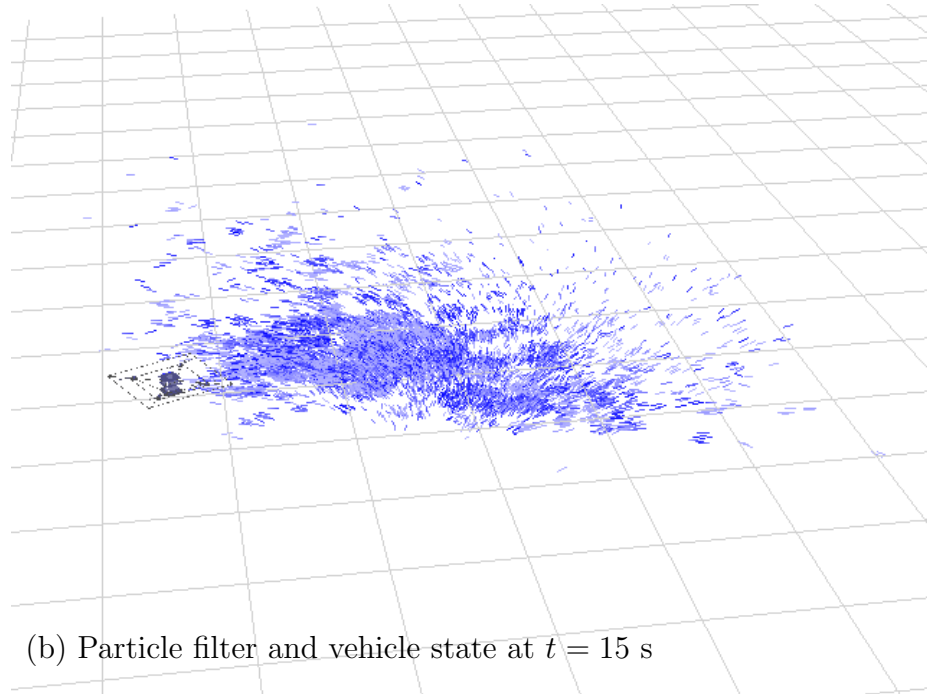
Several flights were flown, with the particle filter consistently localizing the target to within the accuracy of the GPS position estimate. One example is shown in Fig. 5.21, where the beacon is oriented to point North. The behavior of the flight vehicle closely matched that of the simulations. It tended to move orthogonally to the local field line direction, moving toward unexplored space and closer to the true beacon location. At the end it typically hovered over the true beacon location, making only minor motions to further improve the estimate accuracy. Typical search times were on the order of 1 minute to precisely locate the beacon.

### Two Vehicle Search

The two vehicle flight experiments performed an automatic search for a rescue beacon in a  $9 \times 9$  m field, shown in Fig. 5.22. Again, the particles were initialized uniformly over this field, with a depth of 0.2 m, and a uniform distribution on orientation. The



(a) Particle filter and vehicle state at  $t = 0$  s



(b) Particle filter and vehicle state at  $t = 15$  s

Figure 5.21: Flight demonstration of avalanche rescue beacon search localization by a STARMAC quadrotor helicopter in a  $6 \times 10$  m field. This is a visualization of the state of the flight software.

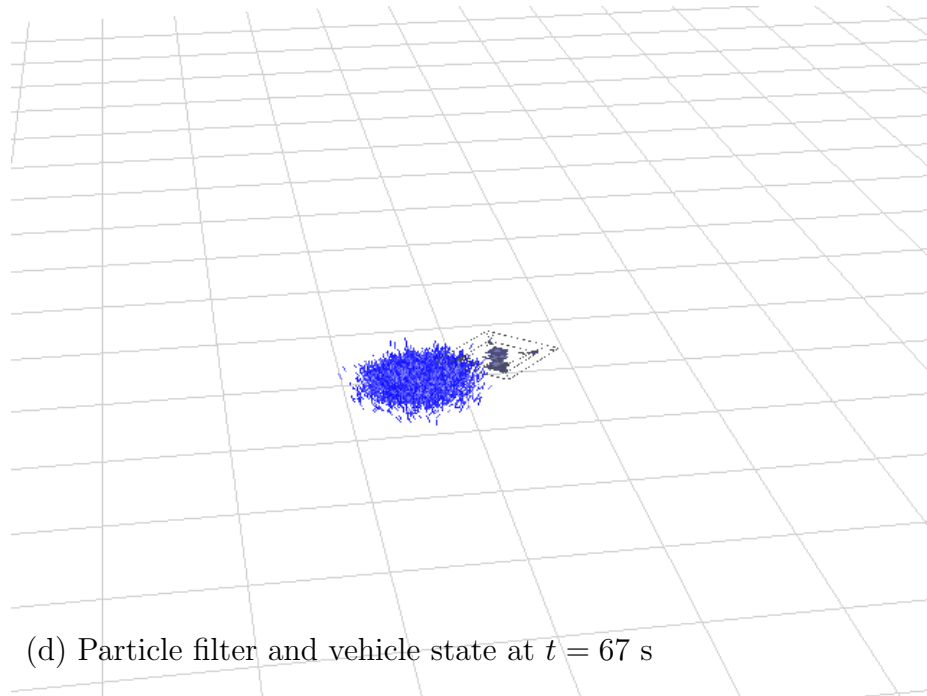
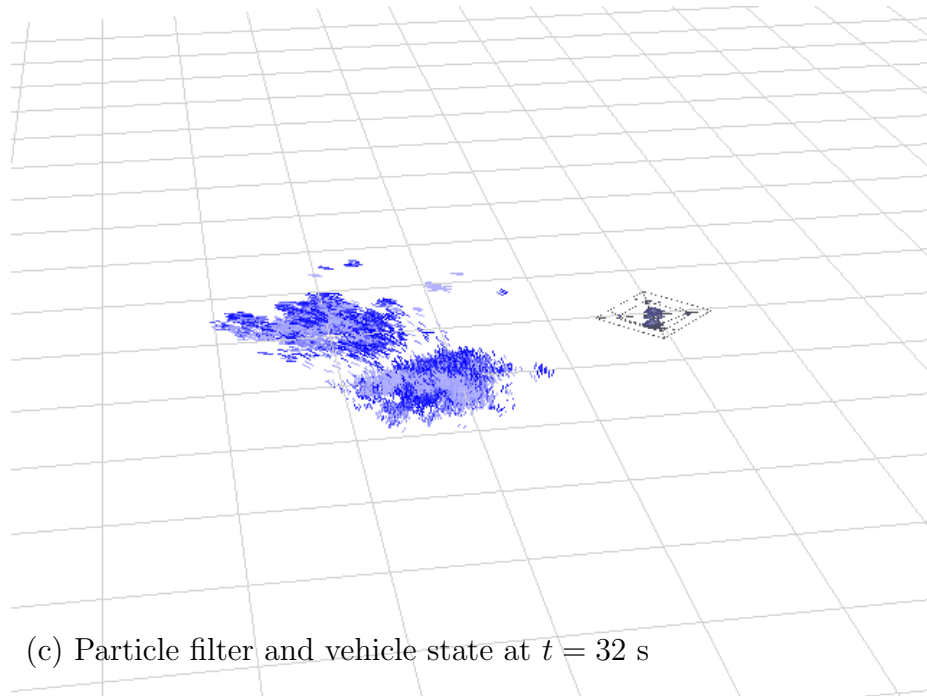


Figure 5.21: (continued) The final MMSE estimate of the beacon's location is 3.09 m East and 2.81 m North from the corner of the search region, matching the true value measured by GPS.



Figure 5.22: Two STARMAC quadrotor helicopters flying with avalanche rescue beacon transceivers.

decoupled optimization was used with the pairwise-node approximation. Note that in this case, the pairwise-node approximation incurs no error—it is exact. To contain the experiment within the available space, the objective function again included a quadratic penalty for leaving the search area, though the objective function was not used when collision avoidance control inputs were required. For the collision avoidance algorithm, a keep-away distance of 2 m was used.

Several flights were flown, with the particle filter consistently localizing the target to within the accuracy of the GPS position estimate. One example is shown in Fig. 5.23, where the beacon is oriented to point North. Again, the behavior of the vehicles closely matched that of the simulations. The vehicles tend to spread out to regions where unique information can be gained. Then, as the particles converge on a single solution, all vehicles tend to move toward that location. In the resulting motion, collision avoidance becomes active, maintaining vehicle separation. Note that a spacing of more than 2 m is maintained due to the effect of the speed of the vehicles on the size of the avoid sets. Typical search times were on the order of 20 seconds to precisely locate the beacon.

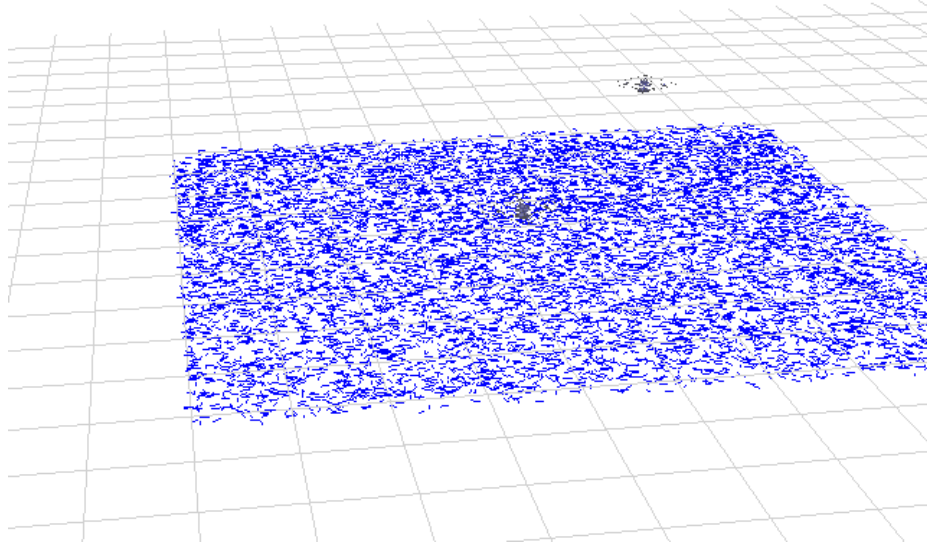
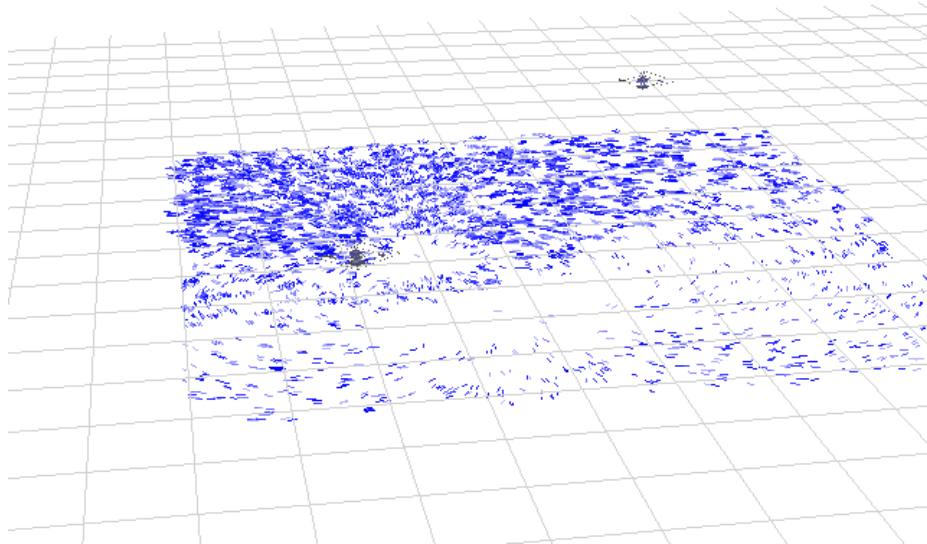
(a) Particle filter and vehicle states at  $t = 0$  s(b) Particle filter and vehicle states at  $t = 5$  s

Figure 5.23: Flight demonstration of avalanche rescue beacon search localization by two STARMAC quadrotor helicopters in a  $9 \times 9$  m field. This is a visualization of the state of the flight software.

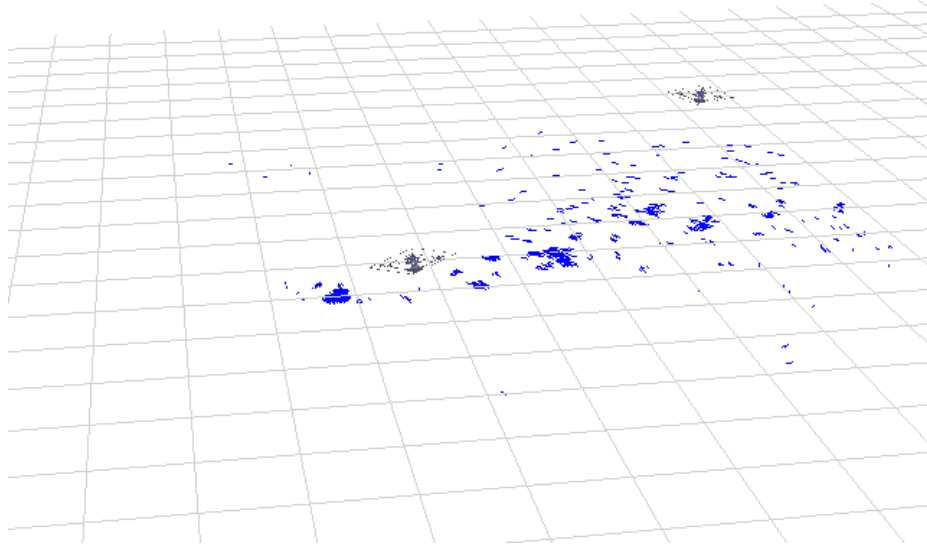
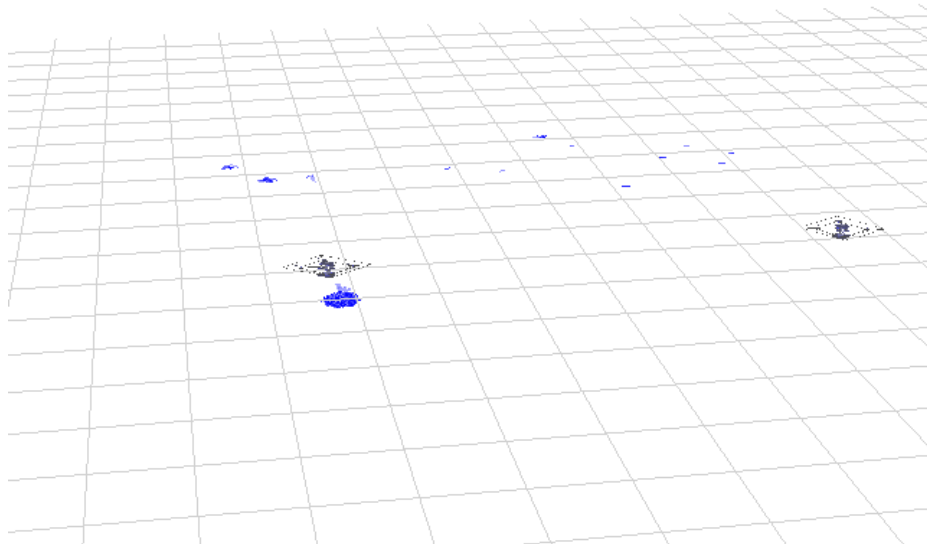
(c) Particle filter and vehicle state at  $t = 8$  s(d) Particle filter and vehicle state at  $t = 16$  s

Figure 5.23: (continued) The additional sensing resources are used effectively to localize the beacon faster than in the single vehicle search experiments.



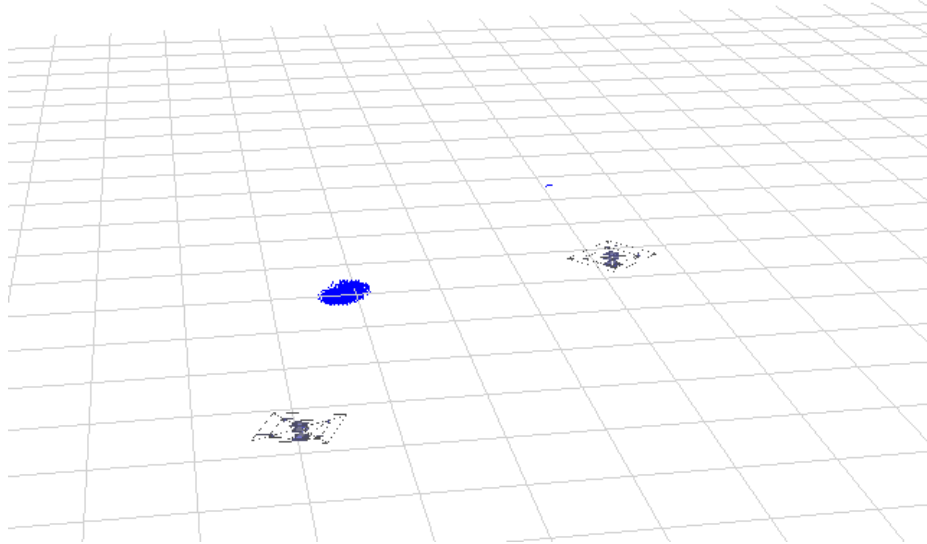
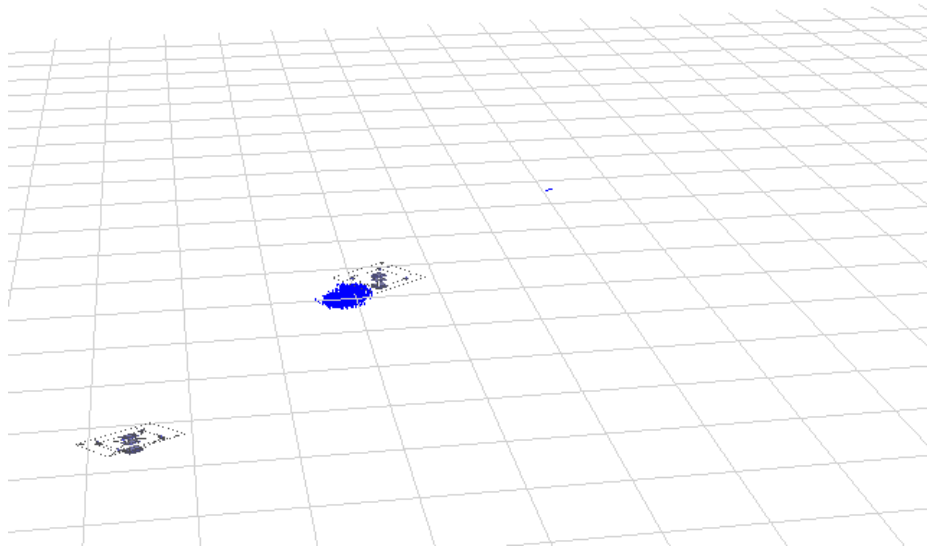
(e) Particle filter and vehicle state at  $t = 20$  s(f) Particle filter and vehicle state at  $t = 21$  s

Figure 5.23: (continued) The final MMSE estimate is 1.80 m East, 3.07 m North from the corner of the region, matching GPS. Collision avoidance causes the vehicle over the beacon to move away. The other vehicle then hovers over the beacon.

### **Flight Test Conclusions**

As expected, with more sensors, localization time decreases. It is interesting to compare the results of these flight tests to the procedures followed by professional rescuers. A rescuer typically travels in the direction of the field line, rather than orthogonal to it. This has two advantages for the rescuer. First, they cannot geo-reference their measurements. They must trust that all field lines eventually converge on the magnetic dipole source. Second, their goal is to rescue the victim once localization is completed. Therefore, the step of moving to the victim right away is indeed sensible. Then, when the rescuer approaches the victim, they perform a more complicated “pinpoint search” to acquire the information that they missed out on while moving on a sub-optimal path. The mobile sensors, however, have the ability to geo-reference their measurements. Therefore, because they already know the local field line orientation, they initially move in the direction which it is most unknown, orthogonal to it, as seen in simulations in Section 5.1.3. The result is that the target is often well localized before it is approached.

# Chapter 6

## Conclusion

### 6.1 Summary

This dissertation has examined challenges in coupled sensing and control systems with the goal of developing algorithms that exploit the relationship between these systems to increase the autonomy of autonomous vehicles. Specifically, autonomous vehicle design, control, and guidance were discussed.

An autonomous robotic testbed, STARMAC, was designed with sufficient sensing and computing resources onboard to enable higher levels of vehicle autonomy. The vehicle control system enabled the use of autonomous guidance algorithms. This testbed inspired the subsequent development of information theoretic control methods, served as a model for simulation, and acted as an experimental platform on which the methods were implemented.

A set of general methods were developed to enable information theoretic distributed control of a mobile sensor network, based on estimation by particle filters, to search for a target. Although particle filters have a higher computational cost than parametric approximation methods, they provide superior descriptiveness of the probability distribution of the search target's state. The techniques presented in this dissertation exploited the structure of these probability distributions of the target state and of the sensor measurements to compute the control inputs leading to future observations that minimize the expected future uncertainty of the target state. Formulae were derived to compute information theoretic quantities using particle filters,

and single-node and pairwise-node approximations were derived to enable scalability in network size. Analytical bounds were found for the error incurred by the approximations, and it was proven that the pairwise-node approximation is a more accurate objective function than the single-node approximation. These techniques open the door to a variety of future applications. They provide methods to decouple information, and to directly use particle filters to quantify and actively seek information.

These general methods were demonstrated in simulated target searches using three different sensing modalities. The results using bearings-only sensors provided comparison to previous work using parametric linearized methods, and demonstrated the performance of the techniques in Monte Carlo experiments. The range-only sensor results demonstrated the ability to handle a sensing scenario that is simple to understand, yet complicated to solve using parametric methods. The results further demonstrated the ability of the proposed algorithms to capture common nonlinear effects, such as saturation and curvature of measurements. Finally, the results of search using avalanche rescue beacons demonstrated the ability of the techniques to handle problems that would pose significant hurdles to previous strategies. Flight experiments with avalanche rescue beacons onboard STARMAC quadrotor helicopters demonstrated the application of the proposed methods.

## 6.2 Future Directions

There is much future work to be done extending the ideas presented here. For instance, there are several applications to consider for automated planning and sensor scheduling. These include tasks such as autonomous guidance for: unexploded ordnance detection, rescue beacon tracking of first responders, RFID tracking in commercial package management, and wildlife monitoring. It would be interesting to investigate automated area surveillance for information collection. Applications include activities such as monitoring environmental conditions, automated structural inspection, surveying disaster areas, and exploring the oceans. Appropriate probabilistic representations and distributed control algorithms must be established.

There are many additional complexities to address for optimization in distributed,

networked systems. For example, longer time horizons and motion in the environment can be considered, with interesting analogies to data compression. Models of information content of messages can be considered for systems with bandwidth constraints. The exchange of information between people and machines can be considered and modeled such that it is optimized. Increasing the optimality and autonomy of mobile sensors networks, using rigorous algorithm design and information theoretic optimization, has the potential to improve the accessibility of information in many areas.

Another direction for research is to re-examine the optimal coding result that motivates the use of Shannon entropy as a metric. The associated derivation, given in Appendix C, assumes that any observation can be made at the subsequent time step. In fact, the observations are limited by dynamic constraints. Applying these constraints to the optimization, even in a discrete game, may yield interesting results.

One additional direction to explore is the application of active sensing to improving performance in other control tasks. One such area is the application of information theory to machine learning problems. Reinforcement learning requires potentially risky and costly experiments. It would be interesting to apply particle filter methods to algorithms that start with a model and refine it automatically, safely, and efficiently, by quantifying uncertainty in a non-parametric way. Another such area is probing the environment to achieve a task. For instance, particle filters have recently been used for robots to estimate door knob positions using contact forces, but accurate determination of the door knob location relies on many random measurements. Another example is a car traversing a four-way stop. In order to determine the ability to go, risk may be reduced by actions such as probing for a turn to cross.

As the technological capabilities of autonomous vehicle components advance, there are ever increasing opportunities to develop algorithms to model their interaction and create methods that exploit these interactions to increase the autonomy of the overall system. This dissertation has addressed several components of the coupling between sensing and control systems, working toward systems that are capable of not only coping with uncertainty, but of acting to reduce uncertainty.

# Appendix A

## Vehicle State Estimation

The state of the quadrotor is estimated using an extended Kalman filter (EKF) to propagate a prediction of the state using inertial measurements and to fuse measurement updates [93]. Inertial measurements are treated as control inputs in the EKF formulation, because they measure the actual effect of the rotors on the dynamics of the vehicle rather than just the predicted effect from the control inputs. It was found that bias estimation is not beneficial for the sensor suite selected. To model the system, define

$$\mathbf{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}, \mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}, \mathbf{p} = \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}, \mathbf{x} = \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \\ \mathbf{p} \end{pmatrix}, \mathbf{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}, \omega = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (\text{A.1})$$

where  $\mathbf{r}$  is the location,  $\mathbf{v}$  is the velocity,  $\mathbf{p}$  is the attitude in Euler angles,  $\mathbf{x}$  is the complete pose,  $\mathbf{a}$  is the measured acceleration in body coordinates, and  $\omega$  is the measured attitude rate in body coordinates.

Define the Euler rotation sequence from the global coordinate frame to the local, vehicle coordinate frame to be 3-2-1, yaw-pitch-roll. The rotation matrix to go from

the global frame to the vehicle's body fixed frame is

$$R_{g,b}(\phi, \theta, \psi) = \begin{pmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \theta \sin \phi \cos \psi - \cos \phi \sin \psi & \sin \theta \sin \phi \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \sin \theta \cos \phi \cos \psi + \sin \phi \sin \psi & \sin \theta \cos \phi \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{pmatrix} \quad (\text{A.2})$$

as found by multiplying the three respective rotation matrices. To transform the vehicle frame to the global frame, apply the transpose of (A.2). Note that the coordinate frame is selected such that when angular velocity measurements are incorporated, the singularity associated with Euler angles occurs when pitched up  $90^\circ$ , which is not planned to occur with the given vehicle. Should this assumption be violated, then it is trivial to change to a different sequence of Euler angles before the singularity is approached too closely.

The matrix to convert from body angular rates to Euler angle rates is

$$R_{\omega, \dot{\mathbf{p}}}(\phi, \theta, \psi) = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{pmatrix} \quad (\text{A.3})$$

as can be found by using rotation matrices to project the Euler angle rates onto the body coordinate frame, setting the sum of the projections equal to the body fixed angular rates, and then solving for the Euler angle rates as a function of the body fixed angular rates.

The motion model is

$$g(\mathbf{x}_{t-1}, \mathbf{u}_t) = \begin{pmatrix} \mathbf{r}_{t-1} + \mathbf{v}_{t-1} \Delta t + \frac{1}{2} (R_{g,b}(\phi_{t-1}, \theta_{t-1}, \psi_{t-1})^T (\mathbf{a}_t + \sigma_a) + \mathbf{g}) \Delta t^2 \\ \mathbf{v}_{t-1} + (R_{g,b}(\phi_{t-1}, \theta_{t-1}, \psi_{t-1})^T (\mathbf{a}_t + \sigma_a) + \mathbf{g}) \Delta t \\ \mathbf{p}_{t-1} + R_{\omega, \dot{\mathbf{p}}}(\phi_{t-1}, \theta_{t-1}, \psi_{t-1}) (\boldsymbol{\omega}_t + \sigma_\omega) \Delta t \end{pmatrix} \quad (\text{A.4})$$

where  $\mathbf{g} = [0 \ 0 \ g]$  is the gravity vector,  $\sigma_a \sim \mathcal{N}([0 \ 0 \ 0]^T, \Sigma_a)$  is the accelerometer noise, and  $\sigma_\omega \sim \mathcal{N}([0 \ 0 \ 0]^T, \Sigma_\omega)$  is the rate gyro noise. This model can be separated into

deterministic and stochastic components,

$$g(\mathbf{x}_{t-1}, \mathbf{u}_t) = \begin{pmatrix} \mathbf{r}_{t-1} + \mathbf{v}_{t-1}\Delta t + \frac{1}{2}(R_{g,b}(\phi_{t-1}, \theta_{t-1}, \psi_{t-1})^T \mathbf{a}_t + \mathbf{g})\Delta t^2 \\ \mathbf{v}_{t-1} + (R_{g,b}(\phi_{t-1}, \theta_{t-1}, \psi_{t-1})^T \mathbf{a}_t + \mathbf{g})\Delta t \\ \mathbf{p}_{t-1} + R_{\omega,\dot{\mathbf{p}}}(\phi_{t-1}, \theta_{t-1}, \psi_{t-1})\omega_t\Delta t \\ \frac{1}{2}\sigma_a\Delta t^2 \\ R_{g,b}(\phi_{t-1}, \theta_{t-1}, \psi_{t-1})^T \sigma_a\Delta t \\ R_{\omega,\dot{\mathbf{p}}}(\phi_{t-1}, \theta_{t-1}, \psi_{t-1})\sigma_\omega\Delta t \end{pmatrix} \quad (\text{A.5})$$

where the first term is deterministic and the second term is stochastic. The stochastic term can be linearized about the current state estimate to obtain the Jacobian of the motion model.

To perform state estimation, this model is recursively used to propagate the state prediction in an EKF as given in [93]. Asynchronously, measurement updates are applied using measurements from the GPS receiver while outside and a color-blob-tracking camera while inside. While inside, the altitude is measured using an ultrasonic ranger. The ultrasonic ranger also uses independent outlier detection and Kalman filtering.



# Appendix B

## Quadrotor Aerodynamics

Two main effects are presented here that have each been experimentally observed on the STARMAC platform as affecting the flight dynamics of the vehicles within the flight envelope of interest. Aerodynamic drag, a reaction force proportional to speed squared, will not be discussed because it is already well known. At moderate speeds, both experimental results and the literature[59] show that the effect of drag on rotorcraft is smaller than the following more dominant effects.

The first effect is that the total thrust varies not only with the power input but also with the free stream velocity and the angle of attack with respect to the free stream. In forward flight, climb and rapid descent, an analytic formulation of the impact on thrust produced is presented based on momentum theory. This is further complicated by a flight regime known as the vortex ring state in which there is no analytical solution for the resulting thrust, and experimental data shows that thrust is highly variable and unsteady.

The second effect results from differing inflow velocities experienced by the advancing and retreating blades. This leads to a phenomenon known as “blade flapping” that induces roll and pitch moments on the rotor hub as well as a deflection of the thrust vector. Expressions for the deflection angle and resulting moments are derived for the hingeless, fixed pitch blades used on the STARMAC II platform.

## B.1 Total Thrust

As a rotorcraft undergoes translational motion, or changes its angle of attack, the induced power, the power transferred to the free stream, changes. To derive the effect of free stream velocity on induced power from conservation of momentum, the induced velocity  $v_i$  of the free stream by the rotors of an ideal vehicle can be found by solving[59]

$$v_i = \frac{v_h^2}{\sqrt{(v_\infty \cos \alpha)^2 + (v_i - v_\infty \sin \alpha)^2}} \quad (\text{B.1})$$

for  $v_i$ , where  $\alpha$  is the angle of attack of the rotor plane with respect to the free stream, with the convention that positive values correspond to pitching up (as with airfoils). The physical (non-imaginary) solution to this equation is accurate over a wide range of flight conditions as shown by experimental results in the literature[50], especially at small angles of attack. At large angles of attack, the rotor can enter the vortex ring state, at which point the equation no longer holds, as will be described below. Nonetheless, it provides an accurate result for much of the flight envelope, including portions of the flight envelope for which momentum theory is not applicable. Using the expression for  $v_i$ , or a numerical solution, the ideal thrust  $T$  for power input  $P$  can be computed, using

$$T = \frac{P}{v_i - v_\infty \sin \alpha} \quad (\text{B.2})$$

where the denominator is the air speed across the rotors.

The value of the ratio of thrust to hover thrust,  $T/T_h$ , is plotted for the  $v_h$  of STARMAC II in Fig. B.2. At low speeds the angle of attack has vanishingly little effect on  $T/T_h$ . However, as speed increases  $T/T_h$  becomes increasingly sensitive to the angle of attack, varying by a substantial fraction of the aircraft's capabilities. Similar to an airplane, pitching up increases the lift force. The angle of attack for which  $T = T_h$  increases with forward speed. For level flight, the power required to retain altitude decreases with the forward speed. However, to maintain speed in level flight, the vehicle must pitch down more as speed increases to cancel drag, leading to a need for more thrust to maintain altitude. There is an optimum speed for any rotorcraft, greater than zero, at which power to stay aloft is minimized (a reduction

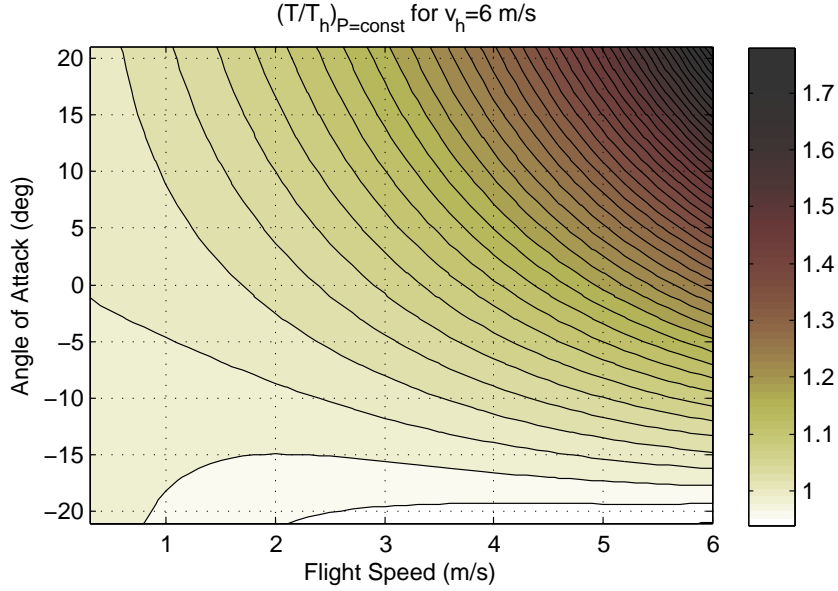


Figure B.1: Thrust dependence on angle of attack and vehicle speed.

from power needs in hover of up to 30% or more)[59]. This speed varies with aircraft configuration.

In the extreme regions of angle of attack, where flight is close to vertical, rotorcraft have three operational modes for the vehicle's climb velocity  $v_c$ , two of which are solutions to Eq. (B.1) (where  $\cos \alpha = 0$ ), and one of which is a recirculation effect that invalidates the assumptions for conservation of momentum[59]. Note that these three modes encompass vertical ascent or descent, and are therefore often encountered. The three modes are defined as follows:

1. Normal working state:  $0 \leq \frac{v_c}{v_h}$
2. Vortex ring state (VRS):  $-2 \leq \frac{v_c}{v_h} < 0$
3. Windmill brake state:  $\frac{v_c}{v_h} < -2$

In normal working state air flows down through the rotor, and in windmill brake state air flows up through the rotor due to rapid descent. In both cases, conservation of momentum can be used to derive the induced velocity. For the normal working

state, encompassing hover and ascent conditions, the induced velocity is

$$v_i = -\frac{v_c}{2} + \sqrt{\left(\frac{v_c}{2}\right)^2 + v_h^2} \quad (\text{B.3})$$

For the windmill braking state, the induced velocity is,

$$v_i = -\frac{v_c}{2} - \sqrt{\left(\frac{v_c}{2}\right)^2 - v_h^2} \quad (\text{B.4})$$

In the vortex ring state, air recirculates through the blades in a periodic and somewhat random fashion. As a result, the induced velocity varies greatly, particularly over the domain  $-1.4 \geq v_c/v_h \geq -0.4$ , causing rapid random variation in the thrust[50]. An empirical model [59] of induced velocity in vortex ring state is

$$v_i = v_h \left( \kappa + k_1 \left(\frac{v_c}{v_h}\right) + k_2 \left(\frac{v_c}{v_h}\right)^2 + k_3 \left(\frac{v_c}{v_h}\right)^3 + k_4 \left(\frac{v_c}{v_h}\right)^4 \right) \quad (\text{B.5})$$

where  $k_1 = -1.125$ ,  $k_2 = -1.372$ ,  $k_3 = -1.718$ ,  $k_4 = -0.655$ , and  $\kappa$  is the measured induced power factor in hover, with a typical value of around 1.15[59]. This model compares with the mean of experimental results in the literature, though it fails to capture the periodic nature of vortex entrapment.

To model the dynamics during climb, the power is the thrust times the speed it is applied at, hence

$$T = \frac{P}{v_c + v_i} \quad (\text{B.6})$$

ignoring profile power (drag) losses. Note that  $Tv_c$  is the power consumed by the climbing motion, whereas  $Tv_i$  is the induced power transferred into the air. It is typically desirable to avoid the vortex ring state, which can be done by maintaining a substantial forward speed while descending[72].

The thrust achieved for a given input power, ignoring profile power losses, can be computed as a function of climb velocity by substituting Eqs. (B.3), (B.4), and (B.5) into Eq. (B.6). For the flight conditions experienced by STARMAC II, the ratio of the thrust to hover thrust at a constant power input is shown in Fig. B.2, for both

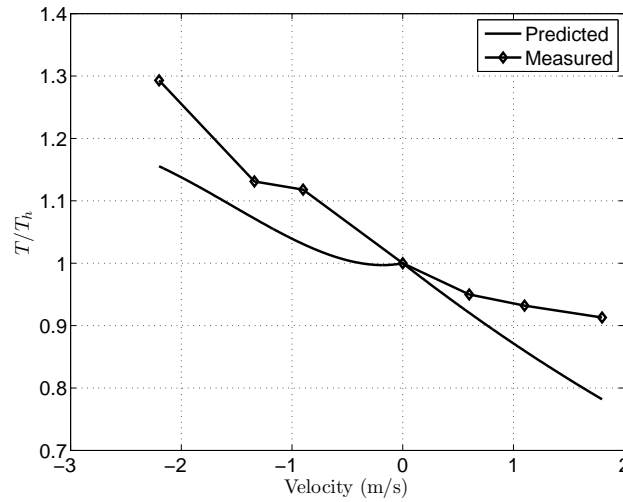


Figure B.2: Predicted ideal thrust and measured climb thrust with vertical velocity.

the theoretical curve, using the solution to the above equations, and experimental data from the thrust test stand using a vertical wind disturbance. As is visible in Fig. B.2, there is a clear loss of thrust associated with climbing, reducing linearly with climb velocity. The vortex ring state has an essentially negligible impact on thrust relative to hover thrust, although in practice this recirculating flow regime is much more variable and hence undesirable. Finally, a significant negative climb velocity results in an increase in resulting thrust. In thrust test stand experiments, the loss of thrust with an applied climb velocity was clearly noted. The descent velocity experiments were less conclusive. The descent speeds available with the test apparatus were less than those required for full vortex ring state, though vibration was observed, indicating that unsteady flow did occur.

The effect of angle of attack on the total thrust generated by STARMAC II was consistently observed in flight tests, as shown in Fig. B.3a, where an earlier generation PID control law is used to control altitude. When the vehicle undergoes a quick roll motion increasing the angle of attack, the thrust increases rapidly, acting as a disturbance on the control system. This reaction is predicted by the equations of motion presented in this section, as shown in a full vehicle simulation in Fig. B.3b. In Section 2.4, a control law is presented that is more capable of rejecting these

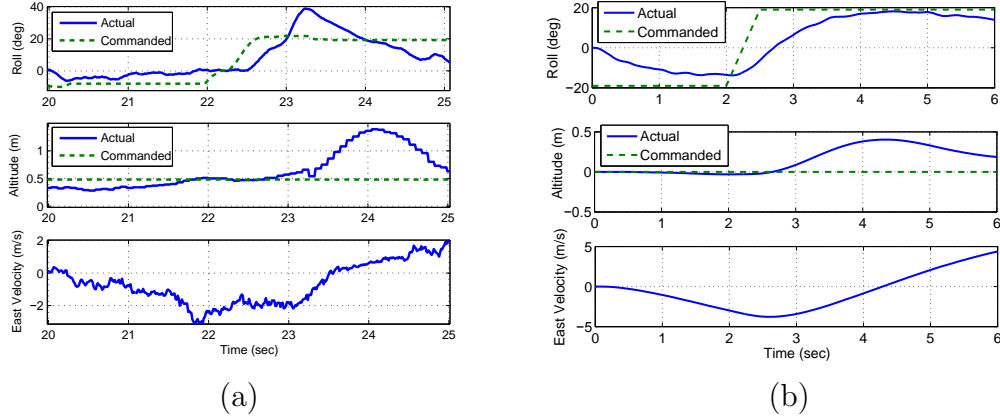


Figure B.3: Effect of angle of attack on altitude control, (a) in a flight test using an earlier generation PID controller, and (b) in simulation using the induced power equations.

disturbances, with feedback control of total thrust.

## B.2 Blade Flapping

The second aerodynamic effect to have a significant impact on the dynamics of quadrotor helicopters is blade flapping. In translational flight, the advancing blade of a rotor has a higher velocity relative to the air, while the retreating blade has a lower velocity, resulting in a variation in lift that causes the rotor blades to flap up and down[81]. This flapping of the blades tilts the rotor plane back from the direction of motion, resulting in a variety of effects on the dynamics of the vehicle, and in particular affecting attitude control performance[79]. For this section, the effects on an individual rotor will be considered, so for readability, the rotor index, subscript  $j$ , is omitted.

Coning, the upward flexure of the rotor blades from the lift force on each blade, combined with unbalanced forcing of the blades, results in a tilt of the rotor plane, the details of which are developed in the literature[81]. This tilt generates moments, but because of the counter-rotating pairs of quadrotor rotors, the lateral effects cancel when yaw rates are low. For stiff 2-bladed rotors, the moments due to the coning

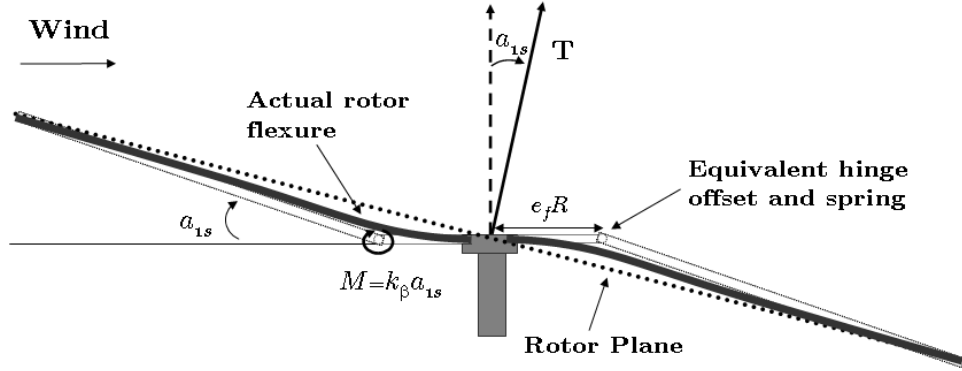


Figure B.4: Blade flapping with stiff rotor blades modeled as hinged blades with offset and spring.

angles are symmetric about the rotor hub and also cancel.

The flap angle  $\beta$  of a rotor blade is typically defined in the helicopter literature as the total deflection of a rotor blade away from horizontal in body coordinates at any point in the rotation, and is calculated as

$$\beta = a_{0s} - a_{1s} \cos \Psi + b_{1s} \sin \Psi \quad (\text{B.7})$$

where  $a_{0s}$  is the blade deflection due to coning,  $a_{1s}$  and  $b_{1s}$  are the longitudinal and lateral blade deflection angles, respectively, due to flapping. The azimuth angle of the blade  $\Psi$  is defined to be zero in the direction opposite horizontal velocity of the rotor. The longitudinal deflection gives the amplitude of the rotor tilt when  $\Psi = (0, \pi)$  rad, and the lateral deflection is the amplitude when  $\Psi = (\pi/2, 3\pi/2)$  rad.

The equation for deflection angle of a flapping rotor with hinged blades is[79]

$$a_{1s} = \frac{1}{1 + \frac{\mu_{lon}^2}{2}} \frac{4}{3} \left( \frac{C_T}{\sigma} \frac{2}{3} \frac{\mu_{lon} \gamma}{a_0} + \mu_{lon} \right) \quad (\text{B.8})$$

where  $a_0$  is the slope of the lift curve per radian (typically about 6.0 for conventional airfoils at low Mach numbers[81]). The longitudinal rotor advance ratio,  $\mu_{lon} = \frac{v_{lon}}{v_t}$ , is the ratio of longitudinal speed to blade tip speed,  $v_t = \Omega R$ . The nondimensional Lock number,  $\gamma = \rho a_0 c R^4 / I_b$ , is the ratio of aerodynamic to centrifugal forces, where

$I_b$  is the moment of inertia of the blade about the hinge,  $c$  is the chord of the blade, and  $R$  is the rotor radius. Finally,  $\sigma = A_b/A$  is the solidity ratio of the rotor, where  $A_b$  is the total area of the rotor blades. The non-dimensionalized torque coefficient is  $C_T$ .

The flapping properties of a stiff, fixed-pitch rotor blade, as used on STARMAC II, can be analyzed by modeling the blade as being hinged at an effective offset  $e_f$  from the center of rotation (expressed as a percentage of the rotor radius) and a torsional spring with stiffness  $k_\beta$  Nm/rad at the hinge[72]. This approximates the first bending mode of the blade and is sufficient for the small deflection angles of interest. Both  $e_f$  and  $k_\beta$  can be determined by measuring the natural frequency  $\omega_n$  of blade vibration[81, 72].

With these parameters, the equilibrium flapping angles can be determined by solving[72]

$$\begin{bmatrix} \lambda_\beta^2 & 0 & 0 & 0 \\ \frac{\gamma}{6}\mu_{lon} & (1 - \lambda_\beta^2) & -\frac{\gamma}{8} & 0 \\ 0 & \frac{\gamma}{8} & (1 - \lambda_\beta^2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{0s} \\ a_{1s} \\ b_{1s} \\ \frac{C_T}{\sigma a_0} \end{bmatrix} = \begin{bmatrix} \frac{\gamma}{8} & -\frac{\gamma}{6} \\ 0 & 0 \\ \frac{\gamma}{3}\mu_{lon} & 0 \\ \frac{1}{3} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \Theta_{avg} \\ \mu_{ver} + \lambda_i \end{bmatrix} \quad (\text{B.9})$$

where the induced inflow ratio  $\lambda_i = \frac{v_i}{v_t}$ . The vertical advance ratio is  $\mu_{ver} = \frac{v_{ver}}{v_t}$ . The average pitch angle of the blade is  $\Theta_{avg}$ .  $\lambda_\beta$  is the ratio of the flapping frequency  $\omega_\beta$  to the angular rate  $\Omega$  of the rotor, and can be calculated for use in Eq. (B.9) as

$$\lambda_\beta = \sqrt{\left(1 + \frac{3}{2}e_f\right) + \frac{k_\beta}{I_b\Omega^2}} \quad (\text{B.10})$$

Blade flapping causes both longitudinal and lateral thrust forces and moments. For quadrotor helicopters, however, the moments generated by lateral deflections cancel when yaw rates are low relative to the speed, and generation of unbalanced moments is due entirely to the longitudinal deflection,  $a_{1s}$ . The backward tilt of the



rotor plane generates longitudinal thrust  $T_{lon}$  (see Fig. B.4),

$$T_{b,lon} = T \sin a_{1s} \quad (\text{B.11})$$

If the center of gravity (c.g.) is not vertically aligned with the rotor plane, this longitudinal force will generate a moment about the c.g.,  $M_{b,lon} = T_{b,lon} z_{cg}$ , where  $z_{cg}$  is the vertical distance from the rotor plane to the c.g. of the vehicle. For stiff rotors, as are used in most current quadrotor helicopters, the tilt of the blades also generates a moment at the rotor hub

$$M_{b,s} = k_{\beta} a_{1s} \quad (\text{B.12})$$

where  $k_{\beta}$  is the stiffness of the rotor blade in  $Nm/rad$ . Finally,  $M_{bf} = M_{b,lon} + M_{b,s}$  is the total longitudinal moment created by blade flapping.

To validate the models developed for blade flapping on the STARMAC platform, the thrust test stand was used. The lateral force due to the deflection of the thrust vector by flapping was measured for a single rotor by blowing air at fixed velocities across a spinning rotor attached to the test stand. This data was filtered and used to calculate the average deflection angle as a function of incident wind velocity and compared to the model predictions. Equation (B.8) predicts a roughly linear relationship between velocity and deflection angle in the shown operating regime. In practice, this equation over-predicts the flapping seen by rotors with non-hinged blades where the stiffness of the blades must be taken into consideration, as in the model presented in Eq. (B.9). For the flapping equations,  $\omega_n$  was also measured using the test stand, giving an effective hinge offset of 25% for the Wattage  $10 \times 4.5$  rotors. The value for  $k_{\beta}$  was measured to be  $0.23Nm/rad$ .

The results, plotted in Fig. B.5, clearly indicate that the non-hinged blade flapping model accurately captures the effect of free stream velocity on the angle of the rotor plane. Note that turbulence caused oscillations in the blade deflection during experiments, so the measurements presented are an average deflection over a period of 20 s.

The effect of blade flapping was consistently observed in flight tests as shown in Fig. B.6a, where an aircraft controlled by a simple PD control law is disturbed by

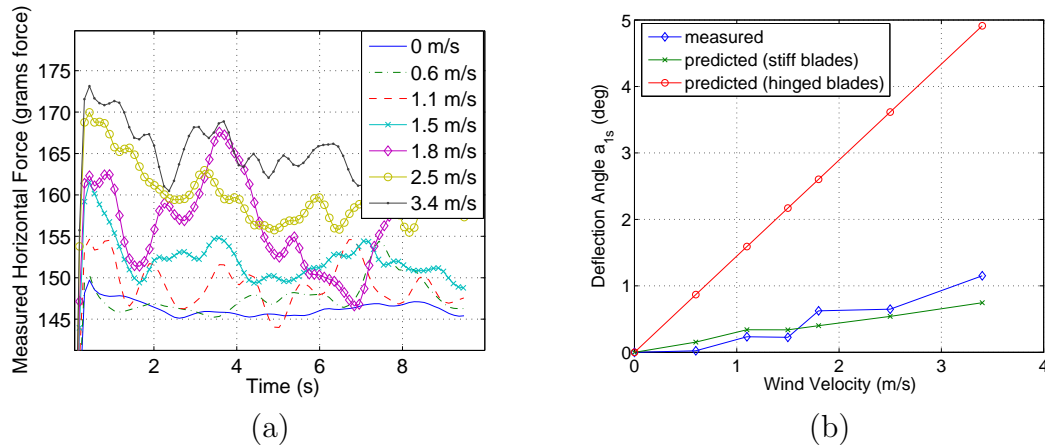


Figure B.5: (a) Horizontal force measurements at different wind speeds to calculate the flapping angles. (b) The measured deflection angle compared with predicted values for hinged, freely flapping blades and for stiff, unflapping blades.

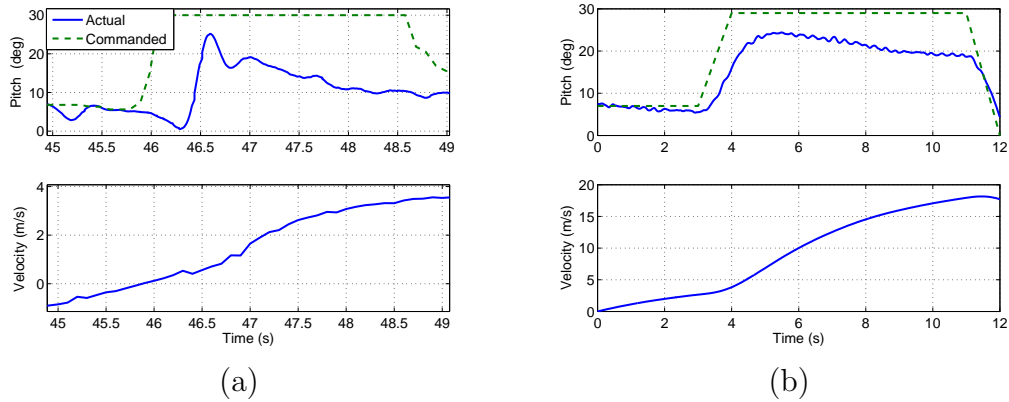


Figure B.6: Effect of blade flapping on pitch control, (a) in a flight test using an earlier generation PD controller and (b) in simulation using the blade flapping equations.

the moment due to blade flapping, an effect that increases with speed. The equations of motion in this section predict this effect on the control system, as shown in a full simulation of the dynamics in Fig. B.6b. In Section 2.4, a control law is presented that is more capable of rejecting these disturbances, actively rejecting discrepancies between desired and actual torques.

# Appendix C

## Information Theory

This appendix provides extended details and derivations for topics in information theory for codewords. Codewords are sets of symbols that are mapped to a particular meaning. These symbols can be bits, letters, or even shapes, for instance.

In the first section, the Kraft inequality derivation is given, which is used in the subsequent section, the derivation of optimal codes. The derivation of optimal codes is the basis for Shannon entropy, and the reason that Shannon entropy minimization is the goal for information seeking control.

### C.1 Kraft Inequality

This inequality provides a constraint for minimizing the expected code word length [17]. For a discrete alphabet of size  $D$ , and codewords of length  $l_1, \dots, l_m$ , the codewords can be represented as the leaves in a  $D$ -ary tree, with the  $D$  branches at each intermediate node representing the value of the symbol at that level in the codeword. Let  $l_{max}$  be the length of the longest codeword in the tree. A codeword  $i$  with length  $l_i$  (at level  $l_i$  in the tree), would have  $D^{l_{max}-l_i}$  descendants at level  $l_{max}$ , disjoint from would-be sets of descendants from any other codeword. The total number of nodes at level  $l_{max}$  must be less than or equal to  $D^{l_{max}}$ . So, the sum of the descendants that

each codeword would have at level  $l_{max}$  satisfies the inequality

$$\sum_i D^{l_{max}-l_i} \leq D^{l_{max}} \quad (\text{C.1})$$

or more simply,

$$\sum_i D^{-l_i} \leq 1 \quad (\text{C.2})$$

which is the Kraft Inequality.

## C.2 Optimal Codes

An optimal code is traditionally defined as one that minimizes the expected length of the codeword. The codeword length is constrained to satisfy the Kraft inequality from Section C.1. It is interesting to consider adding additional constraints, such as maximum codeword length. This lends itself to solving the problem of trading off expected search time with worst case search time. That is left as an open problem. The optimal code problem is then to minimize the expected codeword length,  $L = \sum_i p_i l_i$ , where  $p_i$  is the likelihood of codeword  $i$ , as first derived in [89] and presented in [17].

$$\begin{aligned} & \underset{l_i \in \mathbb{Z}_+}{\text{minimize}} && \sum_i p_i l_i \\ & \text{subject to} && \sum_i D^{-l_i} \leq 1 \end{aligned} \quad (\text{C.3})$$

Using Lagrange multipliers,

$$J = \sum_i p_i l_i + \lambda \left( \sum_i D^{-l_i} - 1 \right) \quad (\text{C.4})$$

Then,

$$\frac{\partial J}{\partial l_i} = p_i - \lambda D^{-l_i} \log_e D \quad (\text{C.5})$$

Setting the derivative to zero to solve for the optimal code lengths,  $l_i^*$ , we obtain

$$p_i = \lambda D^{-l_i^*} \log_e D \quad (\text{C.6})$$

Let us allow  $l_i \in \mathbb{R}_+$  so that the inequality holds with equality. Substituting Equation (C.6) into the constraint, we have

$$\sum_i D^{-l_i^*} = \sum_i \frac{p_i}{\lambda \log_e D} = \frac{1}{\lambda \log_e D} \sum_i p_i = \frac{1}{\lambda \log_e D} = 1 \quad (\text{C.7})$$

Thus,

$$\lambda = \frac{1}{\log_e D} \quad (\text{C.8})$$

Substituting Equation (C.8) into Equation (C.6), one sees that

$$p_i = D^{-l_i^*} \quad (\text{C.9})$$

yielding optimal codeword lengths

$$l_i^* = -\log_D p_i \quad (\text{C.10})$$

and an optimal expected codeword length of

$$L^* = \sum_i p_i l_i^* = -\sum_i p_i \log_D p_i = H_D(X) \quad (\text{C.11})$$

Thus, *the smaller the expected log likelihood of a set of codewords, the smaller the number of questions must be asked, in expectation, to determine a codeword's identity.*

# Bibliography

- [1] Advanced Digital Logic. ADL855 PC104+, 2006. Available from: <http://www.adlogic-pc104.com/products/cpu/pc104/datasheets/MSM855.pdf>.
- [2] E. Altug, J. P. Ostrowski, and C. J. Taylor. Quadrotor control using dual camera visual feedback. In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 3, pages 4294–4299, Taipei, September 2003.
- [3] S. B. Anderson. Historical overview of V/STOL aircraft technology. NASA Technical Memorandum 81280, Ames Research Center, Moffett Field, CA, March 1981.
- [4] K. B. Ariyur and M. Krstić. *Real-Time Optimization by Extremum-Seeking Control*. John Wiley & Sons, Hoboken, NJ, 2003.
- [5] D. Atkins. Companion rescue and avalanche transceivers: The U.S. experience. *AAAP Avalanche Review*, 17(9), 1998.
- [6] BackCountry Access. Tracker dts digital avalanche beacon, 2008. Available from: [http://www.backcountryaccess.com/english/products/tracker\\_dts.php](http://www.backcountryaccess.com/english/products/tracker_dts.php).
- [7] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD thesis, Dept. Elect. Eng., Linköpings Univ., Linköping, Sweden, 1999.
- [8] S. Bouabdallah, P. Murrieri, and R. Siegwart. Towards autonomous indoor micro VTOL. *Autonomous Robots*, 18(2):171–183, March 2005.

- [9] S. Bouabdallah and R. Siegwart. Full control of a quadrotor. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 153–158, San Diego, CA, October 2007.
- [10] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte. Decentralized Bayesian negotiation for cooperative search. In *IEEE/RSJ Int. Conf. on Intelligent Robotics and Syst.*, pages 2681–2686, Sendai, Japan, September 2004.
- [11] M. Buehler, K. Iagnemma, and S. Singh, editors. *The 2005 DARPA Grand Challenge: The Great Robot Race*. Springer, Berlin, 2006.
- [12] P. Castillo, A. Dzul, and R. Lozano. Real-time stabilization and tracking of a four-rotor mini rotorcraft. *IEEE Trans. Contr. Syst. Technol.*, 12(4):510–516, 2004.
- [13] Castle Creations. Phoenix 25 Brushless Motor Control, 2008. Available from: <http://www.castlecreations.com/products/phoenix-25.html>.
- [14] D. E. Chang, S. C. Shadden, J. E. Marsden, and R. Olfati-Saber. Collision avoidance for multiple agent systems. In *Proc. 42nd IEEE Conf. Decision and Control*, pages 539–543, Maui, HI, December 2003.
- [15] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray. On a decentralized active sensing strategy using mobile sensor platforms in a network. In *Proc. 43rd IEEE Conf. Decision and Control*, pages 1914–1919, December 2004.
- [16] W. W. Cochran and R. D. Lord, Jr. A radio-tracking system for wild animals. *J. Wildlife Management*, 27(1):9–24, January 1963.
- [17] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
- [18] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Signal Processing*, 50(3):736–746, March 2002.



- [19] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In G. Sukhatme, S. Schaal, W. Burgard, and D. Fox, editors, *Proceedings of the Robotics Science and Systems Conference*, Philadelphia, PA, 2006.
- [20] DARPA. DARPA Grand Challenge Rulebook, 2004. Available from: <http://www.darpa.mil/grandchallenge05/>.
- [21] D. V. Dimarogonas and K. J. Kyriakopoulos. Decentralized stabilization and collision avoidance of multiple air vehicles with limited sensing capabilities. In *Proc. AACC Amer. Control Conf.*, pages 4667–4672, Portland, OR, June 2005.
- [22] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, New York, NY, 2001.
- [23] DraganFly-Innovations. DraganFlyer IV, 2006. Available from: <http://www.rctoys.com>.
- [24] J. Escareño, S. Salazar-Cruz, and R. Lozano. Embedded control of a four-rotor uav. In *Proc. AACC Amer. Control Conf.*, pages 3936–3941, Minneapolis, MN, June 2006.
- [25] A. A. Feldbaum. Dual control theory I-IV. *Automation Remote Control*, 21, 22:874–880, 1033–1039, 1–12, 109–121, 1960-61.
- [26] X. Feng, K. A. Loparo, and Y. Fang. Optimal state estimation for stochastic systems: An information theoretic approach. *IEEE Trans. Automat. Contr.*, 42(6):771–785, June 1997.
- [27] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, Upper Saddle River, NJ, 4th edition, 2002.
- [28] E. Frazzoli, M. Dahleh, and E. Feron. Trajectory tracking control design for autonomous helicopters using a backstepping algorithm. In *Proc. AACC Amer. Control Conf.*, pages 4102–4107, 2000.

- [29] E. Frew. *Observer Trajectory Generation for Target-Motion Estimation using Monocular Vision*. PhD thesis, Dept. Aero. Astro., Stanford Univ., Stanford, CA, 2003.
- [30] E. W. Frew and S. M. Rock. Trajectory generation for constant velocity target motion estimation using monocular vision. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3479–3484, Taipei, Taiwan, September 2003.
- [31] V. Gavrilets, I. Martinos, B. Mettler, and E. Feron. Flight test and simulation results for an autonomous aerobatic helicopter. In *Proceedings of the 21st Digital Avionics Systems Conference*, pages 8C3-1–8C3-6, 2002. DOI: 10.1109/DASC.2002.1052943.
- [32] A. Gelb, editor. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.
- [33] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proc. F*, 140(2):107–113, April 1993.
- [34] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte. Information-theoretic coordinated control of multiple sensor platforms. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1521–1526, Taipei, Taiwan, September 2003.
- [35] N. Guenard, T. Hamel, and V. Moreau. Dynamic modeling and intuitive control strategy for an x4-flyer. In *Proceedings of the International Conference on Control and Automation*, pages 141–146, Budapest, Hungary, June 2005.
- [36] Gumstix Embedded Computers. Robostix and Verdex Boards, 2008. Available from: <http://www.gumstix.com/>.
- [37] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation and tracking. In *IEEE Trans. Signal Processing*, volume 50, pages 425–437, February 2002.

- [38] J. Hereford and B. Edgerly. 457 khz electromagnetism and the future of avalanche transceivers. In *Proc. Int. Snow Science Workshop*, Big Sky, MT, 2000.
- [39] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.
- [40] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin. The Stanford testbed of autonomous rotorcraft for multi agent control (STARMAC). In *Proc. 23rd Digital Avionics Systems Conf.*, Salt Lake City, UT, November 2004.
- [41] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, Hilton Head, SC, August 2007.
- [42] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In *Proc. AACC Amer. Control Conf.*, New York, NY, July 2007.
- [43] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin. Mutual information methods with particle filters for mobile sensor network control. In *Proc. 45th IEEE Conf. Decision and Control*, pages 1019–1024, San Diego, CA, December 2006.
- [44] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin. Quadrotor helicopter trajectory tracking control. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, Honolulu, HI, August 2008.
- [45] Hokuyo. URG-04LX Laser Range Finder, 2008. Available from: <http://www.hokuyo-aut.jp/02sensor/07scanner/urg.html>.
- [46] J. Hu, M. Prandini, and S. Sastry. Optimal coordinated motions of multiple agents moving on a plane. *SIAM J. Contr. and Optimization*, 42(2):637–668, 2003.

- [47] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. Submitted to *IEEE Int. Conf. on Robotics and Automation*, May 2009.
- [48] N. Ida. *Engineering Electromagnetics*. Springer, New York, NY, 2 edition, 2004.
- [49] G. Inalhan, D. M. Stipanovic, and C. J. Tomlin. Decentralized optimization, with application to multiple aircraft coordination. In *Proc. 41st IEEE Conf. Decision and Control*, volume 1, pages 1147–1155, Las Vegas, December 2002.
- [50] W. Johnson. *Helicopter Theory*, pages 126–131. Princeton University Press, Princeton, NJ, 1980.
- [51] Jürg Schweizer and Georges Krüsi. Avalanche rescue beacon testing. In *Proc. Int. Snow Science Workshop*, Penticton, British Columbia, Canada, October 2002.
- [52] F. Kendoul, D. Lara, I. Fantoni, and R. Lozano. Nonlinear control for systems with bounded inputs: Real-time embedded control applied to uavs. In *Proc. 45th IEEE Conf. Decision and Control*, pages 5888–5893, San Diego, CA, December 2006.
- [53] J. Kim and S. Rock. Stochastic feedback controller design considering the dual effect. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, Keystone, CO, August 2006.
- [54] T. J. Koo and S. Sastry. Output tracking control design of a helicopter model based on approximate linearization. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 4, pages 3635–3640, Tampa Bay, Florida, USA, December 1998.
- [55] C. Kreucher, K. Kastella, and A. O. Hero III. Information based sensor management for multitarget tracking. In *Proc. SPIE*, volume 5204, pages 480–489, Bellingham, WA, August 2003.

- [56] I. Kroo, F. Prinz, M. Shantz, P. Kunz, G. Fay, S. Cheng, T. Fabian, and C. Partridge. The mesicopter: A miniature rotorcraft concept, Phase II interim report. Technical report, Stanford University, 2000.
- [57] Y. Kuwata and J. P. How. Robust cooperative decentralized trajectory optimization using receding horizon milp. In *Proc. AACC Amer. Control Conf.*, pages 522–527, New York, NY, July 2007.
- [58] J. Langelaan and S. Rock. Navigation of small UAVs operating in forests. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, pages 468–473, Providence, Rhode Island, August 2004.
- [59] J. G. Leishman. *Principles of Helicopter Aerodynamics*, pages 36–71. Cambridge University Press, New York, NY, 2000.
- [60] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. David. Collective motion, sensor networks, and ocean sampling. *Proc. IEEE*, 95(1):48–74, January 2007.
- [61] D. G. Luenberger. *Linear and Nonlinear Programming*. Kluwer Academic, Boston, MA, 2 edition, 2003.
- [62] R. Mahony and T. Hamel. Robust trajectory tracking for a scale model autonomous helicopter. *International Journal of Robust and Nonlinear Control*, 14(12):1035–1059, 2004. DOI: 10.1002/rnc.931.
- [63] S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.
- [64] G. Mathews, H. Durrant-Whyte, and M. Prokopenko. Asynchronous gradient-based optimisation for team decision making. In *Proc. 46th IEEE Conf. Decision and Control*, pages 3145–3150, New Orleans, LA, December 2007.
- [65] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 141-3 of *Mathematics in Science and Engineering*. Academic Press, New York, NY, 1982.

- [66] F. Michahelles and B. Schiele. Better rescue through sensors. In *Proc. 1st Int. Workshop Ubiquitous Computing for Cognitive Aids (at UbiComp 2002)*, September 2002.
- [67] Microdrones GmbH. MD4-200 quadrotor helicopter, 2008. Available from: [http://www.microdrones.com/news\\_waypoint\\_navigation.html](http://www.microdrones.com/news_waypoint_navigation.html).
- [68] Microstrain. 3DM-GX1 IMU, 2008. Available from: <http://www.microstrain.com/3dm-gx1.aspx>.
- [69] P. Misra and P. Enge. *Global Positioning System – Signals, Measurements, and Performance*. Ganga-Jamuna Press, Lincoln, Massachusetts, 2001.
- [70] Model Motors. Axi 2208 Brushless Motors, 2008. Available from: <http://www.modelmotors.cz/index.php?page=60&kategorie=2208>.
- [71] P. Moin. *Fundamentals of Engineering Numerical Analysis*. Cambridge University Press, New York, NY, 2001.
- [72] S. Newman. *The Foundations of Helicopter Flight*, pages 107–116. Halsted Press, New York, NY, 1994.
- [73] B. Q. Nguyen, Y.-L. Chuang, D. Tung, C. Hsieh, Z. Jin, L. Shi, D. Marthaler, A. Bertozzi, and R. M. Murray. Virtual attractive-repulsive potentials for cooperative control of second order dynamic vehicles on the caltech mvwt. In *Proc. AACC Amer. Control Conf.*, pages 1084–1089, Portland, OR, June 2005.
- [74] E. B. Nice. Design of a four rotor hovering vehicle. Master’s thesis, Cornell University, 2004.
- [75] Novatel. Superstar II GPS receiver, 2008. Available from: <http://www.novatel.com/products/superstar.htm>.
- [76] P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Trans. Automat. Contr.*, 49(8):1292–1302, 2004.

- [77] S. Park et al. Ric (robust internal-loop compensator) based flight control of a quad-rotor type uav. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, Edmonton, Alberta, August 2005.
- [78] L. Perea, J. How, L. Breger, and P. Elosegui. Nonlinearity in sensor fusion: Divergence issues in EKF, modified truncated SOF, and UKF. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, Hilton Head, SC, August 2007.
- [79] P. Pounds, R. Mahony, and P. Corke. Modelling and control of a quad-rotor robot. In *Proceedings of the Australasian Conference on Robotics and Automation*, Auckland, New Zealand, 2006.
- [80] P. Pounds, R. Mahony, J. Gresham, P. Corke, and J. Roberts. Towards dynamically-favourable quad-rotor aerial robots. In *Proceedings of the Australasian Conference on Robotics and Automation*, Canberra, Australia, 2004.
- [81] R. W. Prouty. *Helicopter Performance, Stability, and Control*, pages 143–146, 476–477. Krieger Publishing Company, Malabar, FL, 1990.
- [82] R. Raffard, C. J. Tomlin, and S. P. Boyd. Distributed optimization for cooperative agents: Application to formation flight. In *Proc. 43rd IEEE Conf. Decision and Control*, pages 2453–2459, December 2004.
- [83] A. U. Raghunathan, V. Gopal, D. Subramanian, L. T. Biegler, and T. Samad. Dynamic optimization strategies for three-dimensional conflict resolution of multiple aircraft. *AIAA J. Guidance, Contr., and Dynamics*, 27(4):586–594, 2004.
- [84] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston, MA, 2004.
- [85] H. Romero, R. Benosman, and R. Lozano. Stabilization and location of a four rotor helicopter applying vision. In *Proc. AACC Amer. Control Conf.*, pages 3930–3935, Minneapolis, MN, June 2006.

- [86] A. Ryan. Information-theoretic tracking control based on particle filter estimate. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, Honolulu, HI, August 2008.
- [87] T. Schouwenaars, J. How, and E. Feron. Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, Providence, RI, August 2004.
- [88] Senscomp. Mini-AE Sonic Ranger, 2008. Available from: <http://www.senscomp.com/minis.htm>.
- [89] C. E. Shannon. A mathematical theory of communication. *The Bell Syst. Tech. J.*, 27:379–423, 623–656, July, October 1948.
- [90] D. H. Shim, H. J. Kim, and S. Sastry. Decentralized nonlinear model predictive control of multiple flying robots. In *Proc. 42nd IEEE Conf. Decision and Control*, pages 3621–3626, Maui, HI, December 2003.
- [91] B. Shucker, T. Murphey, and J. K. Bennett. Switching rules for decentralized control with simple control laws. In *Proc. AACC Amer. Control Conf.*, pages 1485–1492, New York, NY, July 2007.
- [92] D. Stavens, G. Hoffmann, and S. Thrun. Online speed adaptation using supervised learning for high-speed, off-road autonomous driving. In *In the Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, January 2007.
- [93] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.
- [94] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski,



- B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Winning the DARPA grand challenge. *J. Field Robotics*, 23:661–692, 2006.
- [95] Thunder Power. Extreme Series Lithium Polymer Battery Packs, 2008. Available from: <http://www.thunderpower-batteries.com/Li-PolyBatteries.htm>.
- [96] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Trans. Automat. Contr.*, 43(4):509–521, 1998.
- [97] M. Valenti, B. Bethke, G. Fiore, J. P. How, and E. Feron. Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery. In *Proc. AIAA Guidance, Navigation, and Control Conf.*, Keystone, CO, August 2006.
- [98] Videre Design. STH-MDCS 2 Stereo Vision Head, 2008. Available from: <http://www.videredesign.com/sthmdcs2.htm>.
- [99] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.
- [100] S. L. Waslander, G. M. Hoffmann, J. S. Jang, and C. J. Tomlin. Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. In *IEEE/RSJ Int. Conf. on Intelligent Robotics and Syst.*, pages 468–473, Edmonton, Alberta, August 2005.
- [101] Wattage. 10×4.5 Tractor and Pusher Prop, 2008. Available from: <http://www.globalhobby.com/public/gallery/131384.asp>.
- [102] J. L. Williams. *Information Theoretic Sensor Management*. PhD thesis, Elect. Eng., Mass. Inst. Tech., Cambridge, MA, 2007.
- [103] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Mag.*, pages 61–72, March 2002.