

2012

Adaptive Sampling with Mobile Sensor Networks

Shuo Huang
Michigan Technological University

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>



Part of the [Computer Engineering Commons](#)

Copyright 2012 Shuo Huang

Recommended Citation

Huang, Shuo, "Adaptive Sampling with Mobile Sensor Networks", Dissertation, Michigan Technological University, 2012.

<https://digitalcommons.mtu.edu/etds/477>

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>



Part of the [Computer Engineering Commons](#)

ADAPTIVE SAMPLING WITH MOBILE SENSOR NETWORKS

By

Shuo Huang

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Computer Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2012

© 2012 Shuo Huang

This dissertation, "Adaptive Sampling With Mobile Sensor Networks," is hereby approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING.

Department of Electrical and Computer Engineering

Signatures:

Dissertation Advisor _____

Dr. Jindong Tan

Committee Member _____

Dr. Jeffrey Burl

Committee Member _____

Dr. Zhijun Zhao

Committee Member _____

Dr. Chaoli Wang

Department Chair _____

Dr. Daniel R. Fuhrmann

Date _____

Dedication

To everyone who has ever helped me in my life.

Contents

List of Figures	vii
List of Tables	x
Preface	xi
Acknowledgments	xii
Abstract	xiii
1 Introduction	1
1.1 A Big Picture of Adaptive Sampling	1
1.2 Research Challenges in Adaptive Sampling	3
1.2.1 Adaptivity	3
1.2.2 Mobile Sensor Cooperation	4
1.2.3 Onboard Energy Constraints	4
1.3 Adaptive Sampling with Mobile Sensor Networks	5
1.4 Dissertation Overview	7
2 Background: Adaptive Sampling	9
2.1 Adaptive Sampling	9
2.1.1 Mobile Sensor Network Localization	9
2.1.2 Area Coverage	10
2.1.3 Simultaneous Localization and Mapping (SLAM)	11
2.1.4 Target Detection, Localization and Tracking	12
2.1.5 Perimeter Detection and Tracking	13
2.2 Adaptive Sampling and Compressive Sensing	13
2.2.1 Nyquist-Shannon Sampling Theorem and Down-Sampling	14
2.2.2 Compressive Sensing	15
2.2.3 Wavelet Tree Structure	18
2.2.4 Bayesian Compressive Sensing	20

3	Adaptive Sampling with Mobile Sensors^{1 2}	24
3.1	Introduction	24
3.2	Problem Formulation	25
3.3	Adaptive Sampling Algorithm with Mobile Sensors	26
3.3.1	Initial Random Measurements	28
3.3.2	Estimation of the Posterior Mean and Variance	28
3.3.3	Measurement Determination	30
3.3.4	Motion Planning	31
3.3.5	Parameter Update	31
3.3.6	Signal Reconstruction	32
3.4	Experimental Results	33
3.4.1	Average Performance	33
3.4.2	Sampling & Reconstruction of Great Lakes Ice Cover	35
3.4.3	Indoor Mapping, a Practical Application	38
3.4.3.1	Sampling and Reconstruction	39
3.4.3.2	Sparsity Analysis	41
3.4.3.3	Motion Planning	43
3.4.3.4	Map Stitching and Error Correction	43
3.5	Conclusion	46
4	Adaptive Sampling with Mobile Sensor Networks³	47
4.1	Introduction	47
4.2	Problem Formulation	48
4.3	Adaptive Sampling with Mobile Sensor Networks	51
4.3.1	Data Sharing and Sensor Fusion	53
4.3.2	Determining the Most Informative Measurement	56
4.3.3	Network Motion	59
4.3.4	Partial Signal Reconstruction	61
4.4	Experimental Results	62
4.5	Conclusion	65
5	Development of Quadrotor UAV	66
5.1	Quadrotor Kinematics and Dynamics	67
5.1.1	Quadrotor Kinematics	68

¹©2009 IEEE. Portions reprinted with permission, from **Shuo Huang** and Jindong Tan, “*Compressive Mobile Sensing in Robotic Mapping*”, in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3070 - 3075, 2009. See Appendix for a copy of the copyright permission from IEEE.

²©2011 IEEE. Portions reprinted with permission, from **Shuo Huang** and Jindong Tan, “*Adaptive Sampling Using Mobile Robotic Sensors*”, in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1668 - 1673, 2011. See Appendix for a copy of the copyright permission from IEEE.

³©2012 IEEE. Portions reprinted with permission, from **Shuo Huang** and Jindong Tan, “*Adaptive Sampling Using Mobile Sensor Networks*”, in Proceedings of IEEE International Conference on Robotics and Automation, pp. 657 - 662, 2012. See Appendix for a copy of the copyright permission from IEEE.

5.1.2	Quadrotor Dynamics	70
5.2	Quadrotor Control	73
5.2.1	Stabilizing Control	74
5.2.2	Moving Control	75
5.2.3	Speed Control	77
5.3	Hardware Setup	79
5.3.1	Hardware Architecture	79
5.3.2	IMU	81
5.3.3	Laser Scanner and Ultrasonic Sensor	82
5.3.4	Motors and Propeller	83
5.3.5	Onboard Processor	85
5.4	Simulations	85
5.4.1	Set-point Control	87
5.4.2	Tracking Control	90
5.5	Conclusion	93
6	Conclusions	94
	References	95
	Appendix	110
	Appendix A Copyright Permission for Chapter 3 and 4	110

List of Figures

1.1	Diagram of decentralized sensor nodes.	2
1.2	Different sampling patterns applied to a sensing field.	5
	(a) Lake Superior temperature map.	5
	(b) Uniform sampling.	5
	(c) Random sampling.	5
	(d) Adaptive sampling.	5
1.3	Feed-back process on adaptive sampling.	7
2.1	Signal dimension reduced in compressive sensing.	16
2.2	Signal sparse representation in compressive sensing.	16
2.3	Compressive sensing illustration.	17
2.4	Two dimension signal under Haar Wavelet Domain	18
2.5	Concept of Wavelet transform.	19
3.1	Framework of adaptive sampling & reconstruction.	27
3.2	Reconstruction error comparison.	34
	(a) Reconstruction error comparison curves.	34
	(b) Original image.	34
	(c) Reconstruction error 0.104.	34
	(d) Reconstruction error 0.127.	34
3.3	Great Lakes ice cover RGB image.	35
3.4	Reconstruction of gray scale image.	36
3.5	Lake Superior ice cover.	36
3.6	P3-AT robot with LS200 range finder.	38
3.7	Binary map in adaptive sampling experiment.	41
3.8	Laser returns analysis.	42
	(a) Laser range finder returns.	42
	(b) Difference of adjacent laser return pairs.	42
3.9	Two additional laser return examples.	44
	(a) Laser returns analysis example 1. Unit: millimeter.	44
	(b) Laser returns analysis example 2. Unit: millimeter.	44
3.10	Binary map reconstruction.	46
4.1	Informative measurement collection with mobile sensor networks.	49
4.2	Diagram of adaptive sampling algorithm.	50

4.3	A network deployed in a scalar field.	53
4.4	Data sharing diagram.	54
4.5	SIFT example.	56
4.6	Measurement determination diagram.	57
4.7	Motion control diagram.	60
4.8	Partial signal reconstruction diagram.	61
4.9	Scalar field and measurement distribution.	62
	(a) 2-D scalar field.	62
	(b) Measurement of v_1	62
	(c) Measurement of v_2	62
	(d) Measurement of v_3	62
4.10	Reconstruction performance with 8000 measurements.	63
	(a) Sensing field reconstruction with error 0.1029.	63
	(b) Errors for each pixel.	63
4.11	Reconstruction performance comparison.	64
4.12	Path length comparison.	65
5.1	Quadrotor UAV hardware.	67
5.2	Quadrotor + model with roll, pitch, and yaw angles illustrated.	68
5.3	Quadrotor X models with roll, pitch, and yaw angles illustrated.	69
5.4	PD controller with X_{ref} as the control target.	74
5.5	A 4th order controller with X_{ref} as the control target.	77
5.6	A 3rd order controller with X_{ref} as the control target.	78
5.7	UAV architecture.	80
5.8	IMU module.	80
5.9	IMU module orientation displayed on PC screen.	81
	(a) IMU position 1.	81
	(b) IMU position 2.	81
5.10	Laser and ultrasonic sensors.	82
5.11	UAV sitting on force and torque sensors.	83
5.12	UAV motor and propeller with reflective sensor installed.	83
5.13	Motor force and torque <i>v.s.</i> rotation speed.	84
	(a) Motor lifting force measurement.	84
	(b) Motor drag torque measurement.	84
5.14	IMU module.	85
5.15	Set-point control: quadrotor position.	87
5.16	Set-point control: quadrotor speed.	88
5.17	Set-point control: quadrotor angle.	88
5.18	Set-point control: artificial inputs.	89
5.19	Set-point control: motor speed.	89
5.20	Tracking control: tracking trajectory.	90
5.21	Tracking control: quadrotor position.	91
5.22	Tracking control: quadrotor speed.	91

5.23 Tracking control: quadrotor angle.	92
5.24 Tracking control: quadrotor motor speed.	92

List of Tables

5.1 Quadrotor UAV constants and parameters used in simulations 86

Preface

This dissertation presents my research work in pursuing the Ph.D. degree in Computer Engineering at Michigan Technological University. The major contributions of this dissertation is to develop an adaptive sampling method which could adaptively and automatically reconstruct dynamic unknown environments using mobile sensor networks. This dissertation includes previously published articles in Chapter 3 and Chapter 4.

Chapter 3 contains portions of two articles previously published by IEEE. As the first author, I identified the research issue which was to reconstruct an unknown environment using a mobile sensor. With the guidance of my advisor (the second author of these articles), I completed the algorithm design, simulation implementation, and experiment setup. The articles were completed by my advisor and me.

The material in Chapter 4 is adapted from one published IEEE article. As the first author, I identified the research issue which was to reconstruct an unknown environment using mobile sensor networks. With the guidance of my advisor (the second author of the article), I completed the algorithm design, simulation implementation, and experiment setup. The article was completed by my advisor and me.

Acknowledgments

First and foremost, I would like to thank my advisor Dr. Jindong Tan for his guidance and advice of my research and study. His encouragement and support have carried me through this project. It is impossible for me to finish this dissertation without his encouragement, patience, and enthusiasm.

I would also like to thank my committee members, Dr. Zhijun Zhao, Dr. Jeffrey Burl, and Dr. Chaoli Wang for taking the time to review and critique my dissertation and provide me with very valuable feedback. I would also like to thank my great labmates, who have made my life in the Robotics and Embedded Systems Laboratory more colorful and enjoyable. They are Huaming Li, Lufeng Shi, Sheng Hu, Fanyu Kong, Xi Chen, Xinying Zheng, Ya Tian, Xiaolong Liu, Zhenzhou Shao and Jian Lu. I also want to say thanks to my friends at Tech, Qi Gao, Di Huang, and my roommate Xiaodao Chen for their continuous help. Last but not least, I would like to thank my father and mother for their unwavering love, support, and encouragement.

Abstract

Mobile sensor networks have unique advantages compared with wireless sensor networks. The mobility enables mobile sensors to flexibly reconfigure themselves to meet sensing requirements. In this dissertation, an adaptive sampling method for mobile sensor networks is presented. Based on the consideration of sensing resource constraints, computing abilities, and onboard energy limitations, the adaptive sampling method follows a down sampling scheme, which could reduce the total number of measurements, and lower sampling cost. Compressive sensing is a recently developed down sampling method, using a small number of randomly distributed measurements for signal reconstruction. However, original signals cannot be reconstructed using condensed measurements, as addressed by Shannon Sampling Theory. Measurements have to be processed under a sparse domain, and convex optimization methods should be applied to reconstruct original signals. Restricted isometry property would guarantee signals can be recovered with little information loss. While compressive sensing could effectively lower sampling cost, signal reconstruction is still a great research challenge. Compressive sensing always collects random measurements, whose information amount cannot be determined in prior. If each measurement is optimized as the most informative measurement, the reconstruction performance can perform much better.

Based on the above consideration, this dissertation is focusing on an adaptive sampling approach, which could find the most informative measurements in unknown environments and reconstruct original signals. With mobile sensors, measurements are collected sequentially, giving the chance to uniquely optimize each of them. When mobile sensors are about to collect a new measurement from the surrounding environments, existing information is shared among networked sensors so that each sensor would have a global view of the entire environment. Shared information is analyzed under Haar Wavelet domain, under which most nature signals appear sparse, to infer a model of the environments. The most informative measurements can be determined by optimizing model parameters. As a result, all the measurements collected by the mobile sensor network are the most informative measurements given existing information, and a perfect reconstruction would be expected.

To present the adaptive sampling method, a series of research issues will be addressed, including measurement evaluation and collection, mobile network establishment, data fusion, sensor motion, signal reconstruction, etc. Two dimensional scalar field will be reconstructed using the method proposed. Both single mobile sensors and mobile sensor networks will be deployed in the environment, and reconstruction performance of both will be compared. In addition, a particular mobile sensor, a quadrotor UAV is developed, so that the adaptive sampling method can be used in three dimensional scenarios.

Chapter 1

Introduction

The development of technologies in sensing, signal processing, communication, networking, and advanced computing enables mobile sensors and mobile sensor network to accomplish complicated tasks, especially in hazardous environments where human access or assistance is not available. Mobile sensors, including unmanned aerial vehicles, underwater vehicles, and ground vehicles, are capable of various sensing applications of exploration, searching, mapping, target detection and tracking, etc. The common features of mobile sensors in these applications fall into flexible mobility, self navigation, and large coverage. All of these features enable mobile sensors to actively respond to the environment according to sensing demands. In this dissertation, an adaptive sampling method based on mobile sensors is proposed, which can adaptively collect sensing elements of interest and real-time recover the original scene.

1.1 A Big Picture of Adaptive Sampling

Applications of mobile sensing may include mapping, sensing coverage, area exploration, localization, mapping, target detection and tracking, etc. Regardless of the applications, tasks can be simply generalized as information acquisition. The kind of information to acquire and the way to acquire information have been becoming the core questions for mobile sensing. They rely on the mobility of mobile sensor networks, which is the most important ability for mobile sensor nodes to fulfill sensing tasks. For example, the mobility of mobile sensors can help to improve the efficiency and save sensing resources in area exploration, especially for large environments with limited number of sensors. To control the mobility has been a research task for many researchers. The term of controlled mobility

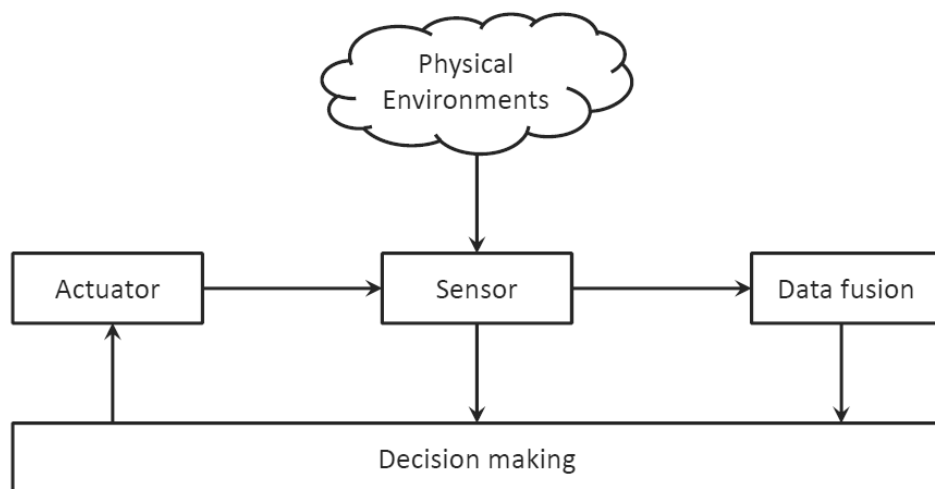


Figure 1.1: Diagram of decentralized sensor nodes.

indicates mobile sensors could perform active sensing. More specifically, mobile sensors adaptively determine their moving trajectories and sensing locations for more reliable and accurate information acquisition from dynamical physical environments.

To make a robust mobile sensing platform, it requires mobile sensors to not only work independently but also collaborate with others. A mobile sensing system could still be working even with failure of certain number of its nodes. This suggests a decentralized sensing mechanism. Figure 1.1 shows such a decentralized diagram for one sensor node in a mobile sensing system. Each mobile node can be summarized as four function blocks: sensor, actuator, data fusion, and decision making.

Sensor: Each mobile node equips with single or multiple sensors, which are the component directly interfacing with the physical environments. Information collected by sensors will be simply processed and sent to data fusion and decision making.

Data fusion: One of the advantages of networked sensors is information propagation and sharing through the entire network. By merging with information collected by other sensor nodes in the network, the local estimates of targets could be updated to increase its accuracy. Fused information is passed to the decision making block also.

Decision making: Based on the knowledge of both the local capabilities and estimates of the target and the transmitted neighborhood information, this part determines the mobile node next motion with considering its sensing, communication, and energy consumption

factors. As a decentralized algorithm, it will only make decisions for the local sensor node given certain constraints from neighbor nodes, like collision avoidance, etc. Then the mobility control instruction is sent back to the actuator.

Actuator: With actuator, mobile platform can move autonomously on receiving of control instructions. The actuator controls sensor node moving parameters like velocity, direction, etc, and leads the mobile node to newly determined sensing locations. These four parts together build the decentralized mobility control. With the feedback control from decision making, mobile sensor network should be able to fulfill adaptive sampling, that is, actively respond to physical environments and adaptively plan motion trajectories to extract the most important information from the environment.

1.2 Research Challenges in Adaptive Sampling

By controlling the mobility of each node in a mobile sensor network, it could navigate itself to acquire information of interest adaptively. The performance of the adaptive sensing is determined by many aspects, including mobile sensor hardware, communication quality, onboard energy, moving strategy, etc. The research challenges of adaptive sampling algorithm in the following aspects.

1.2.1 Adaptivity

In the presence of a mobile sensor network, which is targeted for sampling a physical phenomenon from multiple points simultaneously, the network must be able to adaptively respond to the environments. According to the sampled data, the mobile sensor network changes its topology, looking for the potential samples that can describe the physical phenomenon. The adaptivity of a sensor network can be explained as the process to analyze sampled data, change topology accordingly, and determine new measurements. It involves problems including signal analysis, topology control, motion planning, etc.

Sensors' adaptivity to the environment indicate they are capable to choose the best samples to characterize the unknown environments. The most important portion in the adaptivity is exactly the determination of new measurements given sampled data. Newly determined measurements should be the most informative ones, and can be used to guide sensor motion and rebuild network topology. Thus, the first challenge is to define a metric to evaluate potential measurement values to find the one of the most information. Many methods have

been used to infer new measurements. Hoffmann *et al.* propose mobile sensor network control using mutual information methods for target searching that the mutual information between sensor readings and target state is maximized [1]. The mixed integer linear programming [2] is used to find the vehicle path that maximizes the uncertainty of field estimates to improve the accuracy of the field estimate the most. In this thesis, information entropy is used to estimate the information amount of potential measurements.

1.2.2 Mobile Sensor Cooperation

A mobile sensor network outperforms a single mobile sensor in sensing coverage and efficiency. It requires networked mobile sensors to collaborate with each other. Sensor collaboration may include network topology establishment, information fusion, and mobile network control. Due to the communication bandwidth and time latency, any kind of centralized mechanism seems infeasible, especially when the sensing nodes are extended to large scale. Network should be established locally within a small group of mobile sensors nearby, although the effective communication range could be much larger than that. With sensor linked, sampled data has to be fused together; otherwise the network would be a cluster of individual sensors. Data fusion is the synergistic combination of information from different sources to provide a better understanding of the state of the world. A significant issue in fusing data from multiple sources is determining the distributed measurements that correspond to the same environmental feature. With a fusion center in the mobile sensor network, a global description of the environment can be presented. It is impossible to propagate sensed data through the entire network. In addition, mobile network control is suffering the same problem for centralized methods. A highly centralized cooperative control mechanism cannot quickly enough respond to the environment due to the network propagation and potential loss of particular sensor nodes. Thus, a decentralized mechanism for sensor cooperation is preferred.

1.2.3 Onboard Energy Constraints

Onboard energy is a very critical concern for mobile sensing nodes. An efficient energy consuming plan has to be carried out in order to maximize the sensing ability. Onboard energy is mainly consumed for sensing, data transfer, and sensor motion. Processing with high sampling frequency requires high bandwidth and more sensing energy. Low sampling frequency results in low bandwidth and less energy consumed. However, sensing performance is directly related to the number of measurements collected. Thus, selection of measurements becomes also important on saving onboard energy. In addition, the

communication is not always perfect. High amount of data transmission may cause communication jam. The trade-offs between communication and sensing objectives have been demonstrated in [3], where the impact of imperfect communication on the performance of a decentralized mobile sensor network is characterized. The most straightforward and essential way to save energy and build reliable communication is to reduce the number of measurements, resulting less data transmitted. Fewer measurements would also potentially reduce sensor motion. In a word, more measurements would bring better performance, but also cause more energy consumed at sensing, data transfer, and motion. To effectively utilize the limited on-board energy of mobile sensors, a research task is being post as how to select measurements that can best represent the target and deal with the relationship between measurement sensing and energy consumption.

1.3 Adaptive Sampling with Mobile Sensor Networks

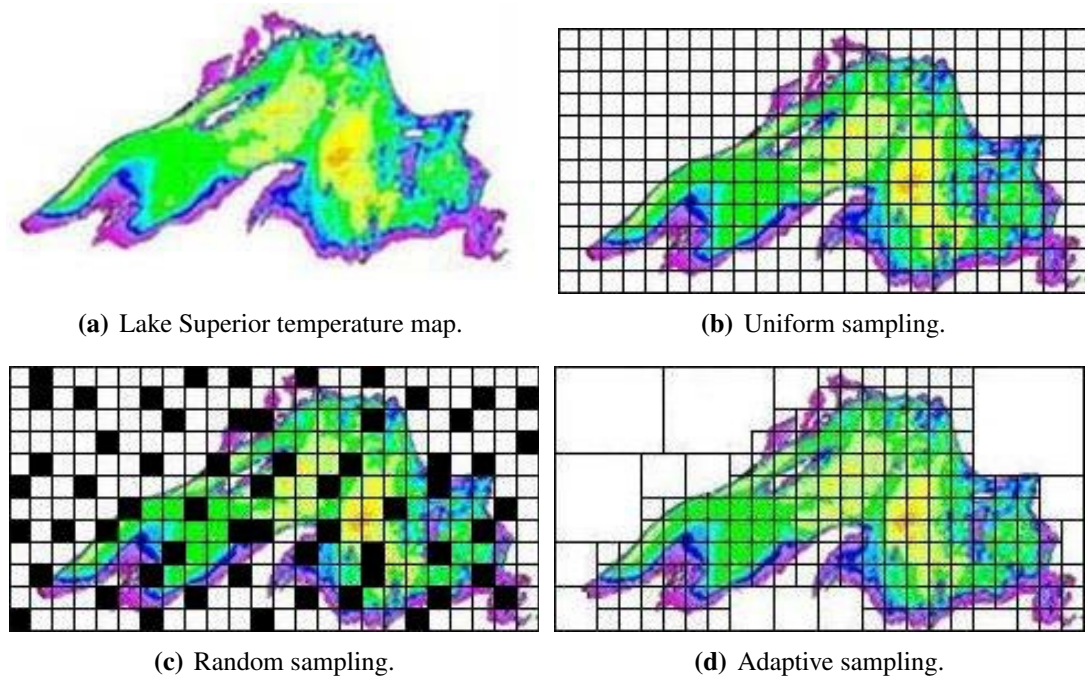


Figure 1.2: Different sampling patterns applied to a desired sensing field.

Information acquisition is the core problem in adaptive mobile sensing. In a mapping application, information is distributed the same as the physical objects. In a target detection and tracking problem, most information concentrates on the target. Adaptive sampling is exactly to search for the information of interest and collect corresponding samples to estimate the unknown environment. The efficiency of mobile sensors falls into the sensing

resource allocation according to the information distribution.

Figure 1.2 illustrates an example of sensing field sampling and reconstruction. To sample and reconstruct the temperature map of Lake Superior in Figure 1.2(a), different sampling patterns are given. The traditional sampling methods suggest uniform sampling according to Shannon sampling theory to recover the sensing field, as shown in Figure 1.2(b). Uniform sampling covers the entire sensing field, and provides accurate reconstruction with high resolution. The sensing resources are allocated uniformly over the sensing field. In other words, each measurement is supposed to have the same weight. However, it would cost too much to cover every element in the scene.

Compressive sensing (CS) [4] proposes a down-sampling method to collect condensed random measurements. Elements in Figure 1.2(c) are partially covered, and ones that are not covered can be estimated given covered ones. The original scalar field can be recovered from the reduced random sampling rate with acceptable reconstruction errors. Therefore, each sample is containing more information than the uniform sampling method. Both of the methods above are of pre-computed sampling patterns that involve fixed and non-adaptive sensing positions. With mobile sensors, it becomes a motion planning problem to cover all the pre-computed way-points. Though the CS approach collects samples of more information, neither uniform sampling nor the CS approach considers the information distribution of the sensing field.

To better capture the information in the sensing field, the concept of adaptive sampling is explained in Figure 1.2(d). Different areas of importance have been paid different attention. In this method, the sensing field is sampled sequentially by mobile sensors. Sensing locations can be chosen to enclose more information after analyzing existing samples, if the information distribution can be roughly estimated. For example, samples are collected according to the information distribution, where the sensing field is first coarsely surveyed and then more measurements are collected from areas that require higher resolutions [5].

Adaptive sampling methods have been discussed in literatures [2, 6, 7, 8, 9, 1]. In these methods, measurements are collected sequentially in nature. Measurement distributions keep changing as sampling proceeds. Old measurements give feedback to determine new measurements, and new measurements are collected to more accurately represent the sensing field. These adaptive sampling methods using mobile sensors explain the efficiency in sensing resource allocation.

In the mapping example above, it is easy to observe that the most informative measurements are located at the boundaries of objects which can be seen physically, while in other practical problems, the most informative measurements may be hidden or cannot be observed under spatial domain. Thus, it is difficult to analyze the sampled data under

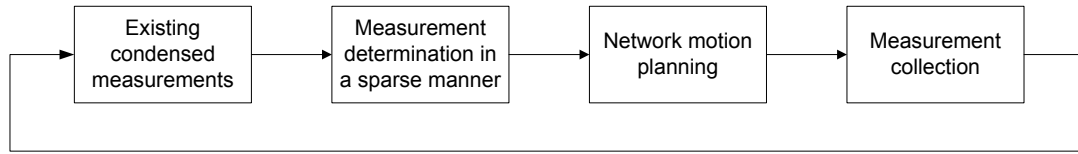


Figure 1.3: Feed-back process on adaptive sampling.

the spatial domain. However, there must be a way to exploit the internal relationship between the most informative measurements and the physical environment. Motivated by the research challenges and the above mapping application, the overall objective of this dissertation is to develop an adaptive sampling method using mobile sensor networks, which integrates cooperative sensing, in-network data processing, topology control, etc. A sparse domain is chosen to sparsely represent the sensing field, under which signals appear highly concentrated, and sparse representation algorithms will be developed accordingly. Figure 1.3 roughly explains the feedback process of adaptive sampling method proposed in this thesis. Condensed sampled data is analyzed under the sparse domain and a new one is determined by projecting back to the spatial domain. After network motion, newly determined measurement is collected and merged with existing measurements to start a new loop. Details about the adaptive sampling method is presented in the remaining of this thesis.

1.4 Dissertation Overview

This thesis is organized in the following manner:

Chapter 2 summarizes related works and background knowledge used in this thesis. Different applications of adaptive sampling are presented to give a global view in different perspectives. The adaptive sampling method addressed in this paper is based on a down sampling method, compressive sensing. Basic concepts of compressive sensing are introduced, as well as its variations, including Bayesian compressive sensing, Wavelet Tree structure, etc.

Chapter 3 presents the adaptive sampling using a single mobile sensor. Given an unknown sensing field, a mobile sensor would adaptively and automatically move and collect samples to recover the sensing field. It deals with the most important research task, how to make mobile sensing adaptive. The determination of the most informative measurements is introduced mathematically. Simulation and experiment results are also attached to support the adaptive sampling method.

Chapter 4 presents the adaptive sampling using mobile sensor networks. Compared with a single mobile sensor, the advantages of mobile sensor networks fall into the information sharing and coverage expanding. With collaboration of networked sensors, target sensing field can be more precisely represented. Besides the most informative measurement determination, network establishment, sensor fusion, and network motion are discussed.

Chapter 5 develops a particular mobile sensor, quadrotor UAV. It could be a crucial member in a mobile sensor network, and suitable for applications like mapping, area exploration, target tracking, etc. Quadrotor dynamics and kinematics are introduced, based on which linear control algorithms are developed. Hardware setup is also introduced, and bench data is generated for simulation use. Simulations include set point moving and stabilizing, and target tracking.

Chapter 6 summarizes the contribution of this work, and discusses the future work.

Chapter 2

Background: Adaptive Sampling

This chapter mainly focuses on the preliminary knowledge for this dissertation, which includes a general introduction of adaptive sampling and its applications, and detailed mathematical background for a particular down sampling method, compressive sensing.

2.1 Adaptive Sampling

The background of this research is the emerging applications and research interests in mobile sensing networks and continuing research efforts in multi-robot systems. With the increasing research interests in wireless sensors networks due to their wide applications [2, 10, 5, 6, 11, 12, 13, 14], researchers are becoming aware of the advantages of using mobile sensing platforms, which can be equipped with more advanced sensors and computational devices. The mobility allows them to actively respond to the environments and reconfigure themselves according to sensing requirements. The adaptive sampling method addressed in this dissertation is a general method to acquire information from environments, which could apply to many different application areas. The rest of this section will summarize related applications.

2.1.1 Mobile Sensor Network Localization

Location awareness is important, as applications such as environment monitoring, target tracking and intrusion detection need to know the locations of mobile node precisely.

Location awareness saves energy for route discovery [15] and enhances security [16]. Existing approaches for network localization include two categories: range based approaches [17] and range free approaches [18]. Range based localization can provide more accurate position estimates compared with range free localization, however, it costs more due to special hardware required.

Generally, localization schemes for mobile sensor networks use a small number of seed nodes, anchor nodes or beacon nodes, whose localizations are known, while larger number of sensor nodes with unknown locations also exist in the network. In range free localization, both seeds and nodes of a mobile sensor network can be either static or moving. The random waypoint mobility model [19] is adopted for both nodes and seeds. In Yi's research work [20], seeds and nodes of a mobile sensor network move randomly with random waypoint model. It assumes that a seed knows not only its position but also the global time of the moment when it obtains its location, which could be transmitted to nearby nodes. Mobile anchor nodes equipped with GPS move in the sensing field and broadcast position information periodically [21]. Sensor nodes compute their locations on receiving broadcasting messages. The scenario that both seed nodes and sensor nodes can move is discussed in Zhang's work [22, 23], where Monte Carlo Localization method is used to achieve high sampling efficiency and localization accuracy.

Range based localization is not that popular used due to the high cost. A range based version of Monte Carlo Localization algorithm is proposed [24], using mobility in the network to increase localization accuracy. A localization scheme for large scale underwater mobile sensor network in harsh aqueous environments has been studied [25], which uses acoustic sensors to provide distance information.

2.1.2 Area Coverage

In some monitoring and exploration applications, the efficiency of sensor networks depends on the coverage of the area. Mobility can improve coverage, especially for unknown or hostile environments like disaster or toxic regions. Mobile sensors are controlled to move to the desired positions to maximize network coverage [26, 27, 28, 29, 30, 31]. Algorithms proposed in these literatures let mobile sensors actively respond to dynamic environments and accordingly recalculate desired sensor locations. Their common objective is to improve coverage after network moving. The difference mainly lies on how the desired sensor positions are computed.

More specifically, an incremental deployment algorithm is proposed [28] to deploy one sensor node each time into an unknown environment, which uses information gathered

from previously deployed nodes to determine its deployment location. A potential field based algorithm is developed [27], where sensor nodes are regarded as virtual particles, and virtual forces in the sensing field would push sensor nodes to avoid collisions. A virtual force based sensor movement strategy is also proposed to enhance network coverage after an initial random placement of sensors [26]. An artificial potential field based algorithm [32], which does not require a prior map of the environment, is also used for area coverage with the constraint that each node should have at least a certain number of neighbors. Voronoi diagrams are used to discover the coverage holes [29], which can move sensors from densely deployed areas to sparsely deployed areas. A peer to peer model based on Delaunay Triangulation and Voronoi diagrams is presented to define the geometrical relationship among mobile sensors [33]. The coverage problem for hybrid networks comprising both static and mobile nodes has been discussed in [34, 35]. A dynamic coverage maintenance scheme, which exploits the limited mobility of the sensor nodes is proposed for coverage loss problem caused by early failure of sensor nodes [36].

Dynamic area coverage with controlled mobility is also studied, which is characterized as area coverage at a specific time instant. Besides trying to improve stationary network configuration, authors of [37] focus on the dynamic aspects of coverage capabilities due to moving networked nodes. Hussein *et al.* formulate the dynamic coverage problem in a mathematically precise problem statement [38], where the coverage goal is to cover a given domain using multiple mobile sensors such that each point can be surveyed.

2.1.3 Simultaneous Localization and Mapping (SLAM)

Autonomous environment exploration, mapping and concurrent localization in the map is a key mission of a mobile sensor network for practical applications in unknown environments. Multi robot SLAM has mostly been addressed in data fusion aspect characterized by two major sources of uncertainty due to the noise in sensing and motion without considering controlled mobility [39]. Extended Kalman filter and particle filter approaches have been successfully implemented for data fusion of multi robot SLAM [40]. Map merging and stitching has been studied in [41].

Active SLAM has been studied for single mobile sensors in [42, 43, 44, 45, 46, 47, 48]. An information-based approach to measure the certainty of a map and vehicle locations has been proposed [47], where a utility function for planning the vehicle trajectory is introduced. A Bayesian optimization method [48] dynamically explores unknown areas by minimizing uncertainty based on current best solution. A method evaluating the quality of actions for a single camera while mapping unknown indoor environments is presented in [49].

Active SLAM for multiple cooperating UAVs on trajectory control is developed [50], where each UAV shares map information over a data fusion network. An integrated approach that combines autonomous exploration with simultaneous localization and mapping is presented in [51]. An approach simultaneously taking into account the moving cost and its utility is presented to coordinate multiple robots [52]. The coordination cannot succeed if sensors do not share a common map. In [53] networked robots communicate with each other actively to verify location hypotheses using a rendezvous strategy. A strategy to select optimal motions of multi robot system equipped with cameras is studied [54] to improve the observation of environment.

2.1.4 Target Detection, Localization and Tracking

Target detection, localization and tracking has recently been paid significant attention due to its importance on environment monitoring, surveillance, etc. Many literatures focus on the motion control of automated robots for target tracking and searching [55, 56, 57, 58, 59]. However, controlled mobility for target detection and tracking poses new challenges, including limited mobility and sensing resources, environmental constraints, and stringent quality of service requirements such as low false alarm rate, high detection probability and bounded detection delay [60]. On the other hand, static sensor network cannot achieve satisfactory sensing performance [61, 62, 63, 64] due to unpredictable spatiotemporal phenomena and dynamic changes of network conditions. Mobile sensors can dynamically reconfigure themselves to improve the robustness and meet sensing demands.

Active target estimation is one of the basic problems for mobile sensor networks. A team of indoor robots equipped with laser ranger finders are coordinately controlled to maximize the information amount to achieve best estimates and optimal resource consumption [65]. Information maximal is also used in [66] to implement coordinated control of sensors. A solution is provided [67] to the optimal trajectory planning problem in target localization for multiple bearing only sensors. Hoffmann *et al.* develop a control architecture based on particle filters to quickly localize a target with bearing only measurements [68]. A gradient searching based decentralized algorithm is presented to control mobile nodes with sonar sensors to estimate the state of a dynamic target [8].

A general theme in probabilistic tracking control is the balance between computational efficiency and the need to model a target estimate distribution in case of target lost. Optimal trajectories are generated for a mobile sensor network to track a moving target using distance only measurements [69]. In [70], optimal trajectories for a team of heterogeneous mobile sensors have been determined to track a moving target using distance and bearing measurements. It may not be favorable for sensor nodes to move, since energy consumed

by moving might be higher than that consumed by sensing and communicating [71]. Zou and Chakrabarty propose to move a sensor node only if a target is detected [72].

The presence or absence of a target is detected using a mobile sensor network [73]. Target tracking is modeled as stochastic event [74] whose arrival and disappearance will be captured by mobile sensors. Authors of [75] investigate the impact of node density on detection performance for stationary target detection with a hybrid sensor network.

2.1.5 Perimeter Detection and Tracking

Perimeter detection and tracking has a wide range of applications: radiation, chemical spills, oil spills, algae blooms, forest fires, etc. Mobile sensor networks could track a substance while dynamically reconfiguring as environment is changing. Gradient based and gradient free approaches are investigated.

The gradient based approaches use more information compared with gradient free approaches. Marthaler *et al.* design a centralized motion algorithm to detect and track algae blooms, where each sensor node measures the concentration gradient [76]. Clark and Fierro detect and track the dynamic perimeter of oil spills [77]. Dantu and Sukhatme use a heterogeneous sensor network to detect and track contour lines of a scalar field [78].

The gradient free approaches depend only on density observations of surroundings. A simple algorithm is proposed for multiple underwater mobile sensors, collecting scalar concentration measurements to monitor an underwater perimeter [79], which has been tested on Caltech's land testbed [80]. Jin and Bertozzi develop a framework for environmental boundary tracking and estimation using hidden Markov model [81]. A fully decentralized algorithm for changing perimeter surveillance is presented [82] with communication range limitations.

2.2 Adaptive Sampling and Compressive Sensing

Motivated by the wide applications and research challenges of adaptive sampling, this dissertation presents an information driven adaptive sampling method with mobile sensor networks. This section will analyze how to reduce sampling cost while actively responding to environments, and introduce the mathematical foundation of this dissertation.

2.2.1 Nyquist-Shannon Sampling Theorem and Down-Sampling

Sampling is the process to translate a signal to a limited series of number. For a continuous signal, the sampling process reduces the signal to a discrete signal. Sampling can be done for functions of one dimension, two dimensions or more, varying in time, space, etc. The goals of sampling include but not limited to transmitting, storing, and reconstructing. All of these would require no information lost, or at least little information lost. It comes out the question how to sample a signal so that all the information can be acquired.

The Nyquist-Shannon sampling theorem [83] has defined the way of sampling. It plays the fundamental role in information theory. The theory suggests to sample a signal at the frequency at least twice of the highest frequency of the sampled signal. By this way, the original signal can be recovered with no information lost. If inadequate samples are collected, it may result aliasing after reconstruction. Aliasing can be prevented or reduced by either increasing the sampling rate to above twice of the signal highest frequency or introducing an anti-aliasing filter to cut off the high frequency parts.

The most straightforward example of the sampling theorem is the formulation for single variable functions. The theorem can be directly applied to time-dependent signals, and extended in a much wider way to formulate functions with more variables. A typical example is grayscale images. Grayscale images are often represented as matrices of real numbers, which can be regarded as two dimension signals. Elements of the matrices represent the relative intensities of pixels whose locations are defined by the intersections of rows and columns. Color images typically consist of three or four grayscale elements, such as RGB, HSV, CMYK, etc. Those color images are exactly functions over 2D sampling domains. If samples or pixels are not adequate, the two dimension images can also suffer from aliasing, which is quite similar to one dimensional signals mentioned above.

The Nyquist-Shannon sampling theorem is saying that signal must be sampled two times faster than the signal bandwidth to avoid aliasing. In other words, if the sampling rates fall lower, more information will lose. In many applications, including video cameras and digital images, the high Nyquist rate results in too many samples, making compression necessary before storing and transmitting. In other applications, such as radars, medical scanners, and high speed analog to digital converters, it would be very expensive to increase the sampling rate to acquire more samples. To maintain acceptable sampling, storing, and transmitting costs as well as acceptable information loss, down-sampling becomes necessary.

By down-sampling, an anti aliasing filter is used to reduce the bandwidth of a desired signal, in order to make sure the Nyquist-Shannon sampling theorem criterion is maintained.

Compared with the Nyquist sampling rate, down-sampling may only pick out samples at interval M times longer, where M is larger than 1. Data rate reduction occurs, thus reducing cost in sampling, transmitting, etc. Down-sampling will introduce information loss with no doubt according to Nyquist-Shannon sampling theorem, but it has to be done to seek the trade-off between cost and information. Down-sampled signal reconstruction can still be improved by introducing powerful signal reconstruction methods.

2.2.2 Compressive Sensing

Compressive sensing, a recently developed down sampling and reconstructing method yielding a sub-Nyquist sampling criterion, uses condensed linear measurements for reconstruction under a sparse domain without losing useful information [4, 84, 85, 86, 87, 88, 89]. Using this down-sampling approach, the signal can be recovered by a sampling rate that is much lower than the requirements from the well-known Shannon sampling theory; hence it reduces the load of sampling and storing. The signal recovery is a convex optimization problem and can be solved by linear algebra [89]. Compressive sensing can be applied to many practical applications [90, 91, 92, 88, 93, 94, 95, 96, 97, 98, 99, 100], most of which suggest random measurements. The Logan-Shepp phantom example in [88] shows that an image can be perfectly reconstructed by samples along radial lines rather than random projections. A promising new paradigm for networked data analysis is described in [100] to reconstruct sparse or compressible networked data in multi-hop networks and wireless sensor networks. Compressive sensing is also used in a mobile cooperative network [95] that is tasked with building a map of the spatial variations of interest with a small number of measurements.

Figure 2.1 shows the sampling pattern of compressive sensing. Given an unknown discrete signal x with the dimension of $N \times 1$, it is required N measurements in traditional sampling methods to fully recover the x . As a down-sampling method, compressive sensing collects fewer samples. It projects the signal from higher dimension (N) to lower dimension (M) with $M < N$, generating the measurement matrix Φ ($N \times M$) and measurement set y ($M \times 1$), subject to $y = \Phi x$. Each measurement is a linear combination of elements of the target signal x with the corresponding row in the measurement matrix Φ as factors.

To recover the signal x , it is exactly to solve x in Figure 2.1. However, it cannot be solved directly, since there are more unknown variables than equations. Therefore, a sparse representation has been introduced as shown in Figure 2.2, where x is represented in a sparse pattern with $x = \Psi^T s$, Ψ is a selected sparse basis, and s ($N \times 1$) is the sparse representation with a small number of significant entries. The insignificant entries are always regarded as zeros in compressive sensing, so the number of unknown entries has

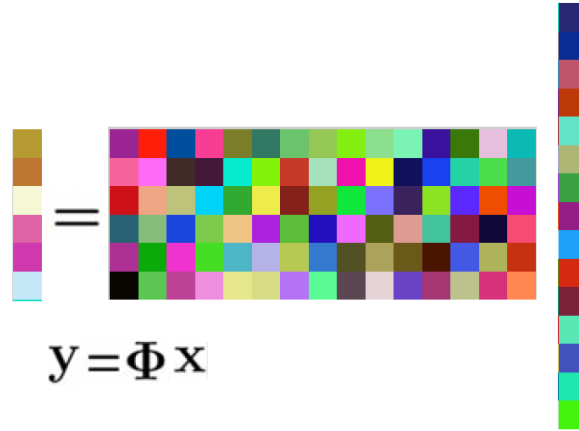


Figure 2.1: Signal dimension reduced in the sampling stage of compressive sensing. Compressive sensing projects signals from higher dimension to lower dimension: $y = \Phi x$.

been reduced. With a sparse s that has only K significant values ($K \ll M$), the system can be solved. Only K unknown variables have to be dealt with rather than N ($K \ll N$), thus reducing the signal complexity.

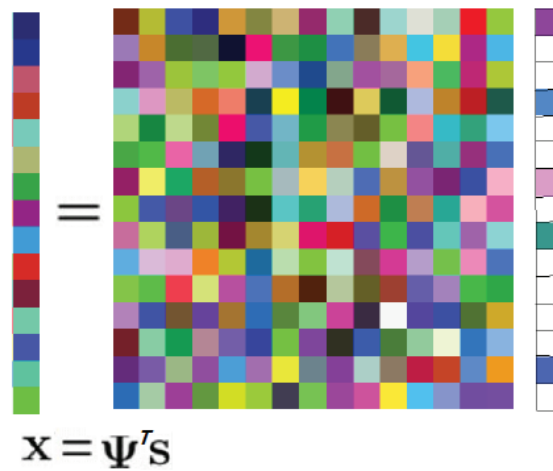


Figure 2.2: Signal sparse representation in compressive sensing: $x = \Psi^T s$.

By combining Figure 2.1 and Figure 2.2, the basic compressive sensing equation can be acquired,

$$y = \Phi x = \Phi \Psi^T s \tag{2.1}$$

which is shown in Figure 2.3. If s in Equation (2.1) can be solved, the target signal x can be reconstructed. In this compressive sensing system, y , Φ , Ψ are known, however, it is

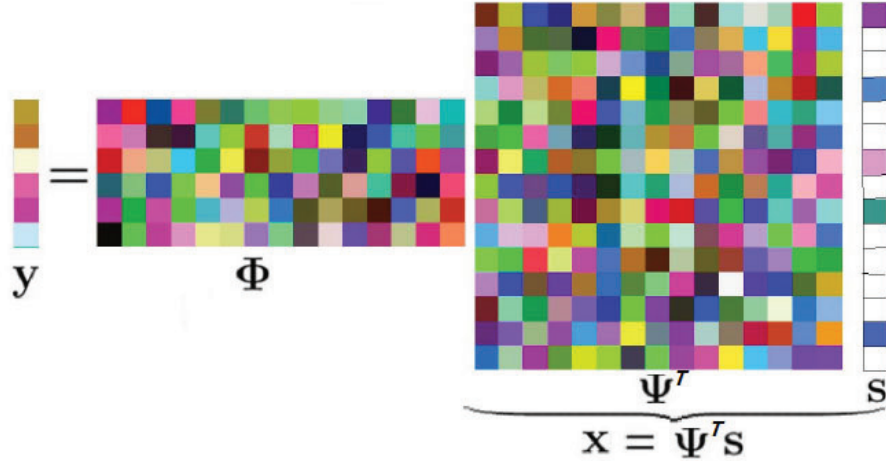


Figure 2.3: Compressive sensing illustration: $y = \Phi x = \Phi \Psi^T s$.

still not that straightforward to solve s , since the locations of the significant entries in s are unknown. A sufficient condition has to be satisfied to make compressive sensing works, Restricted Isometry Property(RIP) [88, 89], which can be simplified as the measurement matrix Φ and sparse basis Ψ should be totally uncorrelated. Different random measurement matrices have been proven effective [101, 102]. It has also been proven [103] that linear random measurement matrix can be used together with many sparse domains for compressive sensing, like Haar Wavelet, Fourier, discrete cosinusoid.

A significant problem in compressive sensing to find out a fast recovery method. To find the solution of compressive sensing is to solve

$$\hat{s} = \arg \min \|s\|_0 \quad s.t. \quad y = \Phi \Psi^T s \quad (2.2)$$

which is an l_0 norm minimization problem. However, this norm problem is NP-complete, but not feasible to solve. An alternative solution has been proposed [89].

$$\hat{s} = \arg \min \|s\|_1 \quad s.t. \quad y = \Phi \Psi^T s \quad (2.3)$$

which is l_1 norm problem that has been proven equivalent to l_0 norm problem for compressive sensing. Based on l_1 norm problem, many reconstruction algorithms have been developed to reduce computational complexity, such as Basis Pursuit [104], Reweighted l_1 norm [105], Orthogonal Matching Pursuit [106], and Tree-Based Orthogonal Matching Pursuit [107].

Besides the theory of compressive sensing, much work also focuses on its applications. Baraniuk *et al.* have developed a new camera architecture with only a single detection element [108]. The novel radar system [103] based on compressive sensing operates

at a potentially low “information rate” rather than the high Nyquist rate. Tian [96] applies compressive sensing to detect the spectrum of cognitive radio by randomly sampling the radio signal in time domain. Other applications include Biosensing [109], Analog-to-Information Conversion [110] and Astronomy [111].

2.2.3 Wavelet Tree Structure

Compressive sensing suggests a signal be represented under a particular sparse domain, given the prior knowledge that the signal appears sparse under this domain. Equation (2.1) describes the sparsely represented signal of one dimension. As compressive sensing is used in image processing [88], researchers find ways to sparsely represent two dimension signals.

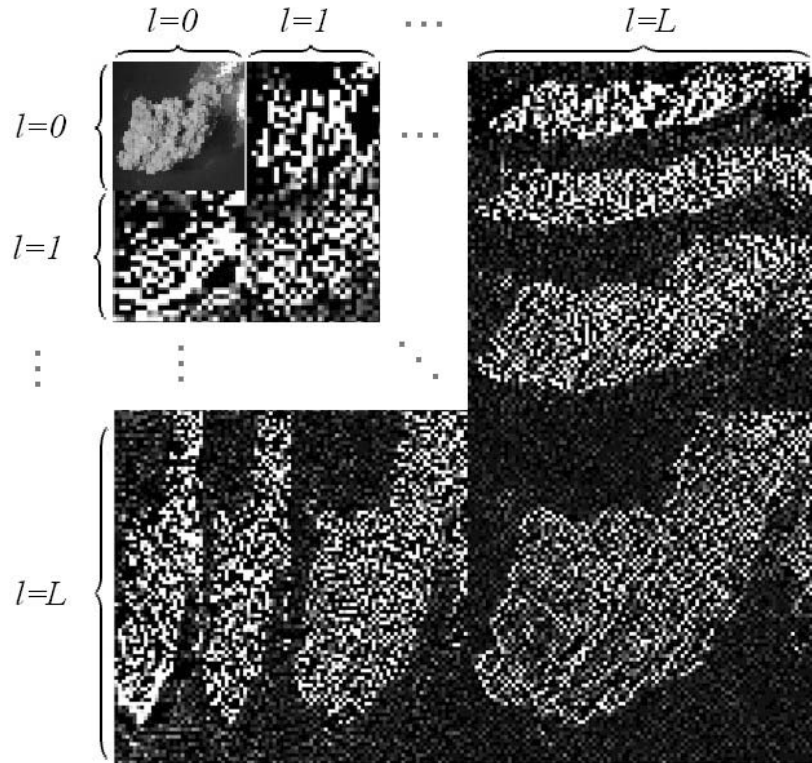


Figure 2.4: Two dimension signal (plume image) under Haar Wavelet Domain.

One popular way to represent two dimension signals is vectorization. Suppose x is a two dimension signal to be sampled and reconstructed. To apply compressive sensing shown in Equation (2.1), target signal x has to be vectorized by stacking its columns

together, generating a long vector. Then the vectorized signal can be sparsely represented easily. However, the vectorization method has its disadvantage. If x represents an image, vectorization may destroy the consistency between adjacent pixels. A particular feature located in a certain area of x may fall into different segments after vectorization, resulting that the sparse representation cannot reflect the particular feature. Therefore, two dimension signals should be directly represented in a two dimension formation sparsely.

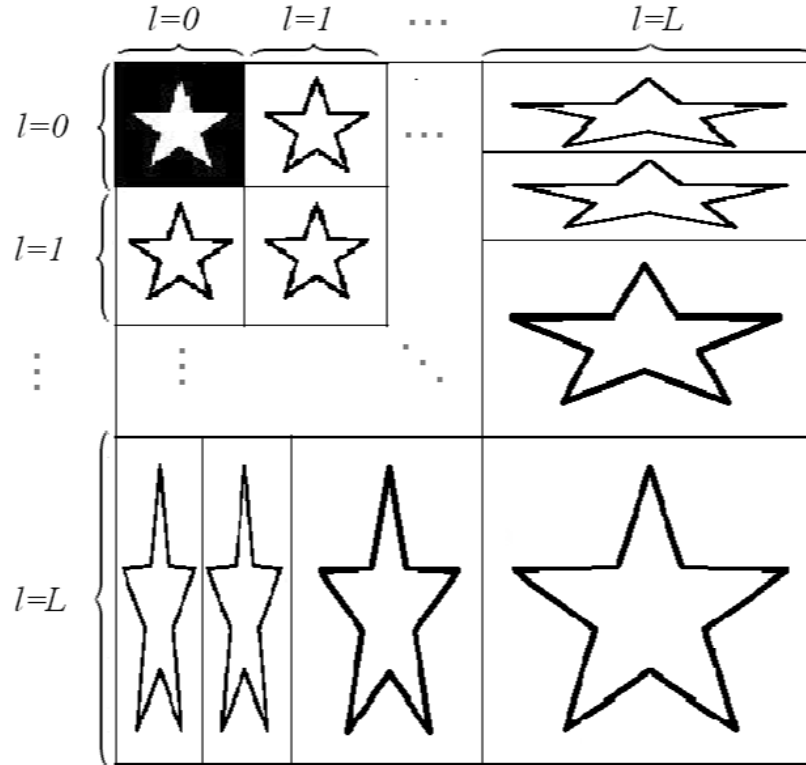


Figure 2.5: Concept of Wavelet transform. A “star” is decomposed under Haar Wavelet domain. It has different scales with indices ranging from 0 to L .

Two dimension Haar Wavelet has been used [112] to sparsely represent a two dimension signal, preserving its features. In the following parts of this dissertation, Ψ is exactly the Haar Wavelet basis. Figure 2.4 shows an example of two dimension sparse representation of a plume image under Haar Wavelet domain. In this example, the plume image is repeated in different scales under Haar Wavelet domain with l as the scale index, ranging from 1 to L . Each feature that can be observed in the original plume image can also be observed in the sparse representation. A more illustrative example is shown in Figure 2.5, where the star is decomposed into different scales in Haar Wavelet domain. The repeated pattern can be easily seen in this example. In these two examples, a significant element in a higher level may result several significant elements in the adjacent lower level correspondingly.

This is the unique feature in Haar Wavelet domain, and is also an important reason why Haar Wavelet is chosen, besides that signals appear sparse under it.

To reconstruct the a two dimension signal, Equation (2.2) has to vary into another form [88]

$$\hat{\mathbf{s}} = \arg \min \|\mathbf{s}\|_{TV} \quad s.t. \quad y = \Phi \Psi^T \mathbf{s} \quad (2.4)$$

where $\|\mathbf{s}\|_{TV}$ indicates the total variation (TV) norm of the two dimension signal s . The TV norm considers the consistency of a two dimension signal in both horizontal and vertical directions, the definition being given

$$\|\mathbf{s}\|_{TV} = \sum_{t_1, t_2} \sqrt{|D_1 s(t_1, t_2)|^2 + |D_2 s(t_1, t_2)|^2} \quad (2.5)$$

where D_1 is the difference in horizontal direction $D_1 s(t_1, t_2) = s(t_1, t_2) - s(t_1 - 1, t_2)$, while D_2 is defined as vertical difference $D_2 s(t_1, t_2) = s(t_1, t_2) - s(t_1, t_2 - 1)$.

2.2.4 Bayesian Compressive Sensing

Compressive sensing introduced above is following a straightforward way consisting of two step, collecting samples and reconstructing the original signals. Random samples are collected [4], and the signal is directly reconstructed without considering the dependencies of entries of the sparse representations. By exploiting the signal structure, Baraniuk *et al.* [113] prove that compressive sensing reconstruction performance can be improved. The following of this chapter will focus on Bayesian compressive sensing, expanding compressive sensing statistically and exploiting wavelet tree structure under Haar Wavelet domain.

Recent research on compressive sensing explores sparsity structures of target signals from the Bayesian perspective. With Bayesian approaches applied into the measuring phase, the Bayesian compressive sensing is developed, where the measurement matrix is generated gradually, so that each measurement is particularly designed to achieve local optimum. The target signal has statistical characteristics which can be used to significantly reduce the number of compressive sensing measurements by the Bayesian inference [114, 113]. With compressive sensing signals modeled as Gaussian random variables, Tipping [115] gives the solution to optimizing the variable parameters. Posterior mean and covariance are estimated through expectation maximization algorithm. Based on posterior mean and covariance, Ji *et al.* [116] propose a Bayesian compressive sensing framework which optimizes the measurement matrix by providing posterior belief of the sparse representation

from the Bayesian perspective. Meng *et al.* [117] apply Bayesian compressive sensing into wireless sensor networks for sparse event detection to reduce the number of wake-up sensors.

By applying Bayesian approaches into the decoding phase, recovery algorithms are enhanced. Signal models and measurement models are considered during recovery, resulting in less reconstruction noise, and faster computation. Markov chain Monte Carlo (MCMC) methods are used to infer new projections by drawing samples. He and Carin [118] demonstrate that substantially fewer projection measurements are sufficient to achieve accurate compressive sensing reconstruction via MCMC methods by analyzing the Wavelet tree structure. Similarly, Tan and Li [119] show that a sparse Bayesian learning algorithm and a block Gibbs sampling algorithm can be used to estimate the transform coefficient vector, which is also called sparse representation. Baron *et al.* [120] perform approximate Bayesian inference using belief propagation decoding to represent the measurement matrix as a graphical model based on available statistical characterization. Babacan *et al.* [121] develop a greedy algorithm for fast reconstruction using Laplace priors to model the sparsity of unknown signals.

From the statistical perspective, signals in the compressive sensing system as shown in Equation (2.1) are modeled under Gaussian distributions. Each entry of the wavelet coefficients/sparse representations s is modeled as a Gaussian random variable

$$s_i \sim N(0, \sigma_i^2), i = 1 \dots N \quad (2.6)$$

where the variance σ_i^2 is written as α_i^{-1} in most literatures, indicating the precision of Gaussian distributions. The sparse representation s contains entries of significance and insignificance. Thus, it can be decomposed as the sum of two items with significant entries and small entries separately, $s = s_m + s_e$, where s_m is established by replacing small entries by zeros, and s_e is established by setting significant entries zero. Both s_m and s_e can be modeled as multivariate Gaussian random variables with significant and small variances using Equation (2.6). However, the second item s_e is always regarded as Gaussian noises, since it can always be ignored for a sparse signal. The system equation is updated

$$y = \Phi\Psi^T s_m + \Phi\Psi^T s_e = \Phi\Psi^T s_m + n_e \quad (2.7)$$

where n_e , the insignificant item, is regarded as a sort of noise. In reality, measurement noise has to be added into the sampling phase. With the consideration of measurement noise, the new equation can be expressed as

$$y = \Phi\Psi^T s_m + n_e + n_w = \Phi\Psi^T s_m + n_0 \quad (2.8)$$

where n_w is the measurement noise, and n_0 is the total noise in the system. The total noise also falls into Gaussian distributions, $n_0 \sim N(0, \alpha_0^{-1})$ with Gaussian precision $\alpha_0^{-1} = \sigma_0^2$.

Thus, the compressive sensing measurements y can be regarded as a multivariate Gaussian distribution

$$y \sim N(\Phi\Psi^T s_m, \alpha_0^{-1}I) \quad (2.9)$$

With s under the Haar Wavelet domain, the Wavelet-tree structure [112] can be studied based on its Gaussian distribution. Figure 2.5 illustrates the structure of the wavelet tree of a “star” under the Haar Wavelet domain. This figure represents the sparse signal s . The signal is repeated at different scales with the scale indexed $l = 0 \dots L$. Each coefficient in upper scale (smaller scale index number) has four children coefficients in the adjacent lower scale (larger scale index number). Whether the coefficient in lower scale is significant or insignificant depends quite a lot on whether its parent is significant or not. Two states can be defined to illustrate the significant or insignificant entries in the sparse representation. Given a compressive sensing model, the entries of the sparse representation s are modeled in two states, “high” and “low”, corresponding to the significant and small values separately. It is believed that one coefficient is likely to be significant, if it has a parent of significance, and *vice versa*. Therefore, in the sparse representation, no entry is independent except for the root level $l = 0$. The dependency between entries is exactly the latent feature to be explored.

To illustrate the “likely to be significant or small” relationship, the variance is considered in two aspects, zero(or nearly zero) and non-zero(or significant), corresponding to “low” and “high”. The transition-probability matrix P at scale l is defined as $P(1, 1) = 1 - \pi_l^0$, $P(1, 2) = \pi_l^0$, $P(2, 1) = 1 - \pi_l^1$, $P(2, 2) = \pi_l^1$. The $P(i, j)$ indicates the probability of a child in state j given its parent in state i . Thus, the variance of each wavelet coefficient locates in a mixture distribution

$$\sigma_i^2 = (1 - \pi_i)\varepsilon\tau_i^2 + \pi_i\tau_i^2 \quad (2.10)$$

$$\pi_i = \begin{cases} \pi_i^0 & \text{if } s_{pa(i)} = 0 \\ \pi_i^1 & \text{if } s_{pa(i)} \neq 0 \\ \pi_i^r & \text{if } \text{root scale} \end{cases} \quad (2.11)$$

where ε is a very small value, and τ is the original variance. The small value ε could be zero, but does not has to be, since the precision is used $\alpha_i = \sigma_i^{-2}$. Each Wavelet coefficient maintains two mixture parameters π_i except that in the root scale ($l = 0$), which indicates two possible different states of the corresponding parent. $p_{pa(i)}$ is defined as the parent

coefficient of coefficient s_i . For $S_{pa(i)} = 0$, it means $\sigma_i^2 = \varepsilon\tau_i^2$, corresponding to the first item in Equation (2.10). For the coefficients at root scale, there is no parents existing, and they are considered as significant values all the time.

Chapter 3

Adaptive Sampling with Mobile Sensors¹

2

This chapter presents an adaptive sparse sampling approach using mobile sensors. An information driven algorithm is established based on compressive sensing. While compressive sensing collects totally random measurements from unknown fields, the proposed algorithm optimizes each individual measurement by maximizing the information it could include. More specifically, a measurement is uniquely designed by analyzing all the existing information under a sparse domain like Haar Wavelet domain. When a mobile sensor relocates to collect that particular measurement, a motion algorithm is designed by leveraging sensor moving cost and information value contained. Since all the samples have been optimized, a much better reconstruction performance can be expected.

3.1 Introduction

Wireless sensor networks have been developed for decades of years as a low cost option for tasks such as tracking, exploration, monitoring, etc. Nowadays, mobile robotic sensors are playing a more and more important role due to their locomotion capability [6], [5]. Unlike

¹©2009 IEEE. Portions reprinted with permission, from **Shuo Huang** and Jindong Tan, “*Compressive Mobile Sensing in Robotic Mapping*”, in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3070 - 3075, 2009. See Appendix for a copy of the copyright permission from IEEE.

²©2011 IEEE. Portions reprinted with permission, from **Shuo Huang** and Jindong Tan, “*Adaptive Sampling Using Mobile Robotic Sensors*”, in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1668 - 1673, 2011. See Appendix for a copy of the copyright permission from IEEE.

static sensors, which are deployed in advance with limited sensing capability, mobile sensors can move adaptively to specified areas of interests to acquire desired information under a controlling mechanism. Mobile robotic sensors are competent for quite a few fields, including environmental monitoring [122], surveillance [123], robotic mapping [124], etc.

As mobile sensors becomes more popular, related algorithms have been developed and implemented. Mobile sensors are more flexible compared with static sensors, thus consuming more energy during moving and sampling. Some adaptive sampling methods have been used in mobile robotic platforms for the energy optimization and sampling efficiency purpose. However, this area can be dig even deeper. In some cases, mobile sensors collect excessive measurements, causing a waste of sensing resources. Typically, Singh *et al.* [5] have proposed a sampling algorithm to sample and reconstruct a spatial map using mobile sensors, where a coarse survey is executed before adaptive refinement, and a portion of the coarse survey may be regarded as excessive measurements containing less useful information. Therefore, a more efficient sampling method is desired.

Traditional sampling methods involving invariant sampling strategies suggest uniform sampling according to Shannon sampling theory. As down-sampling methods are developed, the sampling cost has been significantly reduced. Partial measurements are enough to represent target sensing fields with acceptable errors. Particularly, compressive sensing [4] guarantees signals can be recovered from incomplete measurements with little information loss [85, 86]. Compressive sensing methods collect random measurements, which have been proved efficient and high-performance in digital signal and image processing. But the reconstruction can be still improved. Researchers working on Bayesian compressive sensing have made efforts to further reduce the number of measurements [114, 113]. Given a set of incomplete measurements, Bayesian compressive sensing infers the original signal distribution, and determine the most informative future measurements to increase the sensing field fidelity.

In this chapter, based on compressive sensing, an adaptive sampling algorithm will be addressed. The proposed algorithm collects condensed measurements using a mobile sensor, which could save a lot of sensing resources. Each measurement will be uniquely optimized in order to further improve the sampling efficiency.

3.2 Problem Formulation

To explore and recovery an unknown sensing field, both sensor networks and mobile robotic sensors have been used to sample and reconstruct. Sensor networks based on a pre-configured sensor distribution provide static monitoring. High-resolution

reconstruction can be achieved. However, it is usually impossible or very costly to deploy sensors according to a pre-computed pattern to cover a large sensing field. By adopting mobile sensors, measurements can be collected in a more flexible way. The mobility allows them to reconfigure themselves according to sensing requirements.

While mobile sensors feature adaptivity and flexibility, they always collect excessive measurements in applications. Additionally, mobile robotic sensors are always constrained by limited on-board energy. These suggest that measurements which can represent the sensing field to the most should be collected. Those measurements are the most informative ones out of the entire measurement set. If these most informative data can be determined in prior, measurements collection can be easily done by flexible mobile sensors. Thus, the only problem left is how to evaluate, determine, and collect the most informative measurements. This chapter will give a solution seeking the most informative measurements for mobile sensors.

3.3 Adaptive Sampling Algorithm with Mobile Sensors

Compressive sensing collects random measurements without considering impact between measurements. The proposed algorithm always collects the most informative measurements given all the existing measurements. Measurements are determined and collected sequentially in nature by mobile sensors to increase sensing field fidelity. A recursive procedure is proposed to collect measurements, analyze existing measurements, and determine new measurements. Wavelet structure and statistical properties of target sensing field signals will be inferred from existing measurements. New measurements are determined by exploiting these statistical properties sequentially. Mobile sensors provide the flexibility to sequentially sample a target sensing field, but also bring unique challenges of direct application of Bayesian compressive sensing. Motion planning is designed to strike the balance between the information amount of each measurement and moving cost from current sensing position to the next one. Overall, the adaptive sampling method alternates between analyzing existing measurements and collecting new measurements.

Figure 3.1 shows the adaptive sampling and reconstruction framework. This framework includes six major parts, *initial measurements*, *mean and variance estimate*, *measurement determination*, *motion planning*, *parameter update*, and *signal reconstruction*. In *initial measurements*, a small portion of random measurements are collected by a mobile sensor at the very beginning. Compressive sensing model in Equation (2.1) is established with Φ as the initial random measurement matrix. An iterative process is used to determine and collect measurements. *mean and variance estimate* approximates the posterior mean and variance of the wavelet coefficients under Haar Wavelet domain. The π parameter

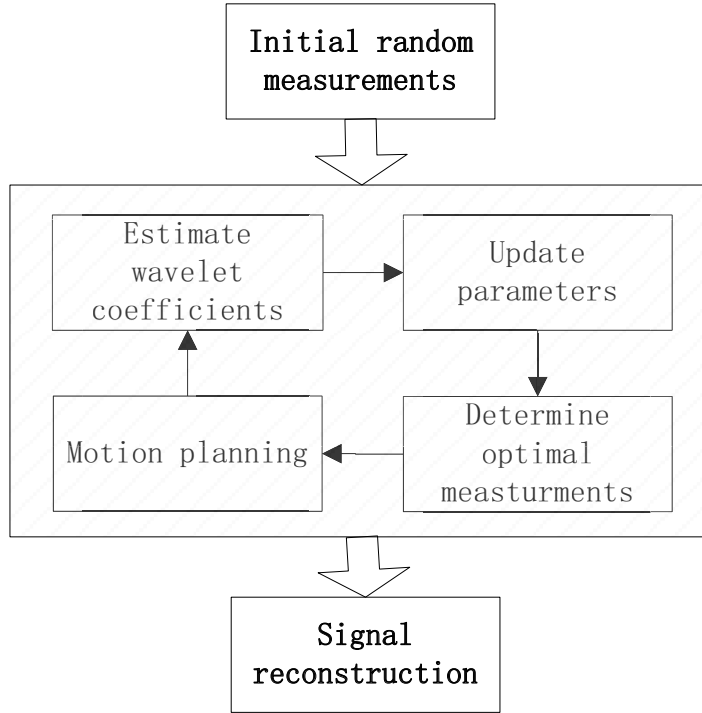


Figure 3.1: Framework of adaptive sampling & reconstruction.

introduced in Equation (2.11) is used to estimate posterior mean and variance, and will be updated in the following process. Once the posterior mean and variance are obtained, *Measurement determination* chooses optimal measurements to maximize the posterior variance. The measurement of the maximal posterior variance can diminish the sensing field uncertainty to the most extent. However, the moving cost has to be considered. *Motion planning* makes a trade-off between measurement importance and sensor motion, and confirm the position of the next measurement from a few candidate ones. The new measurement is collected after the motion is confirmed and accomplished. *Parameter update* generates more accurate parameters given all the existing measurements. Accurate π would result accurate measurement inference. Each of the four-step iterations ends up with a new row added into the measurement matrix Φ and a new measurement added into the measurements set y . In the next iteration, all the computation is based on the newly generated Φ and y . The sampling iteration terminates, once enough measurements are collected. Signal is then reconstructed.

3.3.1 Initial Random Measurements

Nothing can be inferred or optimized without initial information. Before the mobile system runs the adaptive sampling algorithm, an initial measurement set has to be collected. Based on the compressive sensing theory, this portion should be random. This initial set of measurements can be used to recover the entire scene, however, the reconstruction performance could be very coarse. The following four parts help to improve the reconstruction.

3.3.2 Estimation of the Posterior Mean and Variance

In the compressive sensing model, condensed measurements are collected, leaving the unknown sensing field filled with uncertainty. The uncertainty indicates a state that has limited knowledge to exactly describe an existing state. In this case, it is a lack of measurements to represent the target sensing field x . With the knowledge of a measurement set y , we have the uncertainty of x at certain level. A new measurement can result in describing the sensing field more accurately, meaning that it can reduce the uncertainty to some extent. The most informative measurement exactly indicates the measurement that can diminish the uncertainty to the most extent. Given a set existing measurements, this section shows how to estimate the sparse representation s , and Section 3.3.3 shows how to determine the most informative measurement accordingly.

To estimate the signal x in the statistical model, it is important to infer the mean and variance. In the compressive sensing system, the sparse representation s (wavelet coefficients) is subject to multivariate Gaussian distributions. Given a measurement set y , the posterior distribution of s is

$$p(s|y, \alpha, \alpha_0) = \frac{p(y|s, \alpha, \alpha_0)p(s|\alpha, \alpha_0)}{p(y|\alpha, \alpha_0)} = \frac{p(y|s, \alpha_0)p(s|\alpha)}{p(y|\alpha, \alpha_0)} \quad (3.1)$$

where $\alpha = \{\alpha_1, \alpha_2 \dots \alpha_N\}$, and α_i corresponds to the Gaussian precision of each Wavelet coefficient. $p(y|s, \alpha_0)$ and $p(s|\alpha)$ are Gaussian distributions from Equation (2.9) and Equation (2.6). $p(y|\alpha, \alpha_0)$ is a likelihood function for α and α_0 , being given

$$\begin{aligned} \mathcal{L}(\alpha, \alpha_0) &= p(y|\alpha, \alpha_0) = \int p(y|s, \alpha_0)p(s|\alpha)ds \\ &= (2\pi)^{-\frac{N}{2}} \left| \frac{I}{\alpha_0} + \Phi A^{-1} \Phi^T \right| \exp\left\{ \frac{I}{\alpha_0} + \Phi A^{-1} \Phi^T \right\} \end{aligned} \quad (3.2)$$

Since the three distributions in Equation (3.1) $p(y|s, \alpha_0)$, $p(s|\alpha)$, and $p(y|\alpha, \alpha_0)$ are known, the posterior distribution of s can be calculated. Given a set of measurements y , s is subject to conditional Gaussian distributions with posterior mean and variance that is $s|y, \alpha, \alpha_0 \sim N(\mu, \Sigma)$, yielding

$$\mu = \alpha_0 \Sigma \Phi^T y \quad (3.3)$$

$$\Sigma = (\alpha_0 (\Phi \Psi^T)^T \Phi \Psi + A)^{-1} \quad (3.4)$$

where $A = \text{diag}(\alpha_1, \alpha_2 \dots \alpha_N)$.

With the posterior mean and variance calculated, it is worth noting that the parameters in Equation (3.2), which is a likelihood function, can be optimized in order to infer new precisions given the posterior mean and variance. To optimize parameters in Equation (3.2) is to maximize $\mathcal{L}(\alpha, \alpha_0)$, so that the best α and α_0 can be found for the distribution. This is known as the type-II maximum likelihood, and can be implemented through differentiation, being given in [115]. The new parameters yield:

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2} \quad (3.5)$$

$$1/\alpha_0^{new} = \frac{\|y - \Phi \mu\|_2^2}{N - \sum \gamma_i} = \frac{\|y - \Phi \mu\|_2^2}{\sum \pi_i} \quad (3.6)$$

where $\gamma_i = 1 - \alpha_i \Sigma_{ii}$. An iterative algorithm can be executed by alternating between Equation (3.3) (3.4) and Equation (3.5) (3.6). The convergence is very fast, and α_i becomes large for zero or small wavelet coefficients.

As stated in Section 2.2.3, signals have special associations between adjacent scales in Haar Wavelet representation, suggesting that the parameter α can be further optimized. From Equation (2.10), it can be seen that the wavelet structure can be imposed to the variance. Thus, the same structure can be imposed to Equation (3.5), resulting

$$1/\alpha_i^{new} = (1 - \pi_i) \varepsilon \frac{\mu_i^2}{\gamma_i} + \pi_i \frac{\mu_i^2}{\gamma_i} \quad (3.7)$$

The iterative algorithm has been changed to alternating between Equation (3.3) (3.4) and Equation (3.7) (3.6). The posterior mean and variance can be approximated with a few iterations. For the consideration of computation time, 5 iterations have been used, and α_0 and α_i are initially set 0. Thus, given any set of measurements, the posterior mean and variance of the original signal can be estimated.

3.3.3 Measurement Determination

With the estimation of posterior mean and variance, new measurements are to be determined to achieve local information maximum. In the compressive sensing model, the basic procedure to determine a new measurement is to add a new row to the measurement matrix Φ , such that the resulting measurement would have the maximal information amount. The maximal information amount indicates diminishing the uncertainty to the most. In the following description, the derivation of information amount will be shown.

Suppose as a time instant, k measurements have been collected with the given measurement matrix $\Phi = [\phi_1^T, \phi_2^T \cdots \phi_k^T]^T$. The aim in this step is to determine a new row ϕ_{k+1} and add it into the measurement matrix Φ . The latest mean and covariance of s are computed given new α and α_0

$$\mu^{new} = \alpha_0^{new} \Sigma \Phi^T y \quad (3.8)$$

$$\Sigma^{new} = (\alpha_0^{new} (\Phi \Psi^T)^T \Phi \Psi + A^{new})^{-1} \quad (3.9)$$

where $A^{new} = \text{diag}(\alpha_1^{new}, \alpha_2^{new} \dots \alpha_N^{new})$.

The potential new measurement is supposed to be y_{k+1} . It is Gaussian distributed, and its mean and variance can be calculated according to Equation 2.9.

$$y_{k+1} \sim (\phi_{k+1} \Psi^T \mu^{new}, \text{Var}(y_{k+1})) \quad (3.10)$$

where $\text{Var}(y_{k+1}) = \phi_{k+1} \Psi^T \Sigma^{new} (\phi_{k+1} \Psi^T)^T$. With mean and variance, the information amount of y_{k+1} can be quantified by its differential entropy

$$H(y_{k+1}) = - \int p(y_{k+1}) \log p(y_{k+1}) dy = \ln(\sqrt{2\pi e \text{Var}(y_{k+1})}) \quad (3.11)$$

The larger the $H(y_{k+1})$ is, the more information it contains. The most informative measurement projection ϕ_{k+1} is found by maximizing this quantity

$$\hat{\phi}_{k+1} = \arg \max_{\phi_{k+1}} H(y_{k+1}) \quad (3.12)$$

which is the locally optimized measurement. There might be millions of new row forms. It is impossible to go through all of them and determine an optimal one. The candidate new measurements are chosen from a pre-designed library of a small number of elements, subject to the particular sensing pattern of mobile robotic sensors used.

The differential entropy maximization problem of a Gaussian variable can be simplified as to maximize its variance. In such case, it is exactly to maximize $Var(y_{k+1})$. It implies that the information amount of a potential measurement can be explained in terms of its variance. This also explains the relationship between the uncertainty and the most informative measurement.

3.3.4 Motion Planning

Measurements have to be determined not only subject to the maximal posterior variance but also the moving cost from current position to the newly determined measurement position. Based on the description in Section 3.3.3, the most informative measurement is determined as the local optimum. However, it would be wasteful if the mobile sensor go back and forth to collect measurements. To plan the motion of a mobile sensor, the moving cost should be considered. Not the most informative measurement is collected, but the one which strikes the balance between information content and moving cost. Thus, Equation (3.12) should be changed by adding a moving cost item.

$$\hat{\phi}_{k+1} = \underset{\phi_{k+1}}{arg \max} (H(y_{k+1}) - \omega C(\phi_k, \phi_{k+1})) \quad (3.13)$$

where the first item remains the same as Equation (3.12), and the second item represents the moving cost. $C(\cdot, \cdot)$ is a function indicating the moving cost between the two measurement positions, and $C(\phi_k, \phi_{k+1,i})$ represents the moving cost from current sensing position to the next one. ω is a weighted factor that is used to achieve a proper ratio for these two items. In plane areas, the moving cost is usually proportional to the moving distance. However, in the real applications, many other factors have to be considered, including mobile sensor turning, obstacles, collisions, etc. In the simulation, we fairly assume the moving cost is proportional to the moving distance.

3.3.5 Parameter Update

The paradigm shown in Figure 3.1 contains four major parts for each loop with parameter update at last. The parameter π_i indicates whether the wavelet coefficient is significant or not. The aim to update π is to approximate the distribution of s more accurately. It includes π_i for each wavelet coefficient. The concept of conjugate prior is used to estimate variables more accurately with known samples [125]. *Beta* distribution is the conjugate prior of a binomial distribution given some existing samples.

The update of parameter π_i uses all the α_i generated in prior loops. Supposing at the k_{th} loop, there are $k - 1$ sets of α_i that can be used to to estimation π_i , denoting as $\alpha_i^{new(j)}$, $j = 1 \dots k - 1$. The π parameters are updated by

$$p(\pi_i^0 | -) = B(e_0^{i0} + \sum_j 1(\alpha_i^{new(j)} \ll \text{Inf}, \alpha_{pa(i)}^{new(j)} = \text{Inf}), f_0^{i0} + \sum_j 1(\alpha_i^{new(j)} = \text{Inf}, \alpha_{pa(i)}^{new(j)} = \text{Inf})) \quad (3.14)$$

$$p(\pi_i^1 | -) = B(e_0^{i1} + \sum_j 1(\alpha_i^{new(j)} \ll \text{Inf}, \alpha_{pa(i)}^{new(j)} \ll \text{Inf}), f_0^{i1} + \sum_j 1(\alpha_i^{new(j)} \ll \text{Inf}, \alpha_{pa(i)}^{new(j)} = \text{Inf})) \quad (3.15)$$

$$p(\pi_i^r | -) = B(e_0^{ir} + \sum_j 1(\alpha_i^{new(j)} \ll \text{Inf}), f_0^{ir} + \sum_j 1(\alpha_i^{new(j)} = \text{Inf})) \quad (3.16)$$

where $1(x)$ denotes an indicator function. $1(x) = 1$ for true statement x , and 0 otherwise. For example, $1(\alpha_i^{new(j)} \ll \text{Inf}, \alpha_{pa(i)}^{new(j)} = \text{Inf})$ in Equation (3.14) indicates the Wavelet coefficient s_i is significant ($\alpha_i^{new(j)} \ll \text{Inf}$), and its parent $s_{pa(i)}$ is insignificant ($\alpha_{pa(i)}^{new(j)} = \text{Inf}$). e and f are prior hyperparameters for *Beta* distributions. The parameter initialization and setup have been discussed in [118].

3.3.6 Signal Reconstruction

Iterations containing the above four steps carry on until enough measurements are collected. Signals are reconstructed given measurements including initial random measurements and adaptive measurements. All the existing measurements consist the

measurement set Y for the compressive sensing model. Reconstruction is executed based on the method addressed in [118].

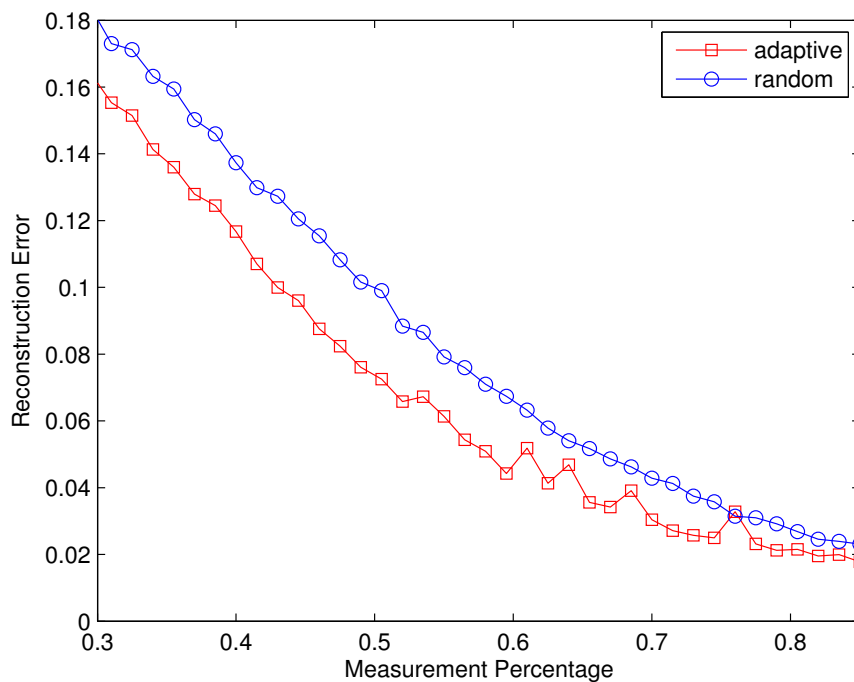
3.4 Experimental Results

In this section, both simulation and experiment are presented using the sampling method addressed in this chapter. In the first simulation, average reconstruction performance is evaluated by comparing reconstructed signals with original signals. In the second simulation, a particular simulation is given, Great Lakes ice cover reconstruction. Moving cost is evaluated by comparing adaptive measurements and random measurements, and the experiment is compared with another adaptive sampling method. All the computations in this section were performed using Matlab run on a server with two Intel Xeon 5130 CPUs working at 2G and 8G DDR2 memory. In the experiment, a real application is carried on, which reconstructs a 2-D map.

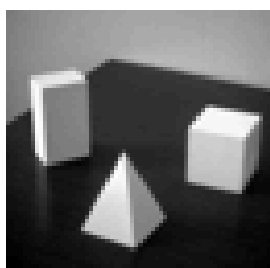
3.4.1 Average Performance

To evaluate the average performance, we reconstruct a gray scale image in Figure 3.2(b). The image consists of 32 by 32 pixels, ranging from 0 to 255. Measurements are collected in both adaptive way and random way. The experiments are repeated many times for the average performance. All measurements are collected in the spatial domain and assumed $3 \times N(0, 1)$ measurement noise. Suppose the original image in Figure 3.2(b) is x , each measurement can be represented as $y_i = \phi_i x = \phi_i \Psi^T s$, where s is the 2-D wavelet transform of the original image. We assume a mobile sensor is equipped with a single-pixel imaging sensor [97]. It covers a small area of 5 by 5 pixels, generating a measurement as a linear combination of these 5 by 5 pixels each time. In each row of the measurement matrix ϕ_i , there exists only $5 \times 5 = 25$ non-zero entries. The coefficients of the combinations are drawn from a standard Gaussian distribution with 0 mean and 1 variance. To evaluate the reconstruction performance, reconstruction error is introduced as $\|x_{rec} - x\|_2 / \|x\|_2$, where x and x_{rec} represent original and reconstructed signals separately.

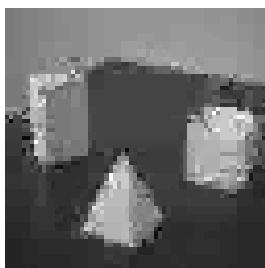
The original image and reconstructions of different ratios of adaptive measurements are shown in Figure 3.2. The blue curve indicates the reconstruction error of the random measurements, where the total percentage is shown in the figure. As discussed in 3.3, a small number of initial random measurements is collected to perform the adaptive sampling method. The red curve in Figure 3.2(a) is generated by choosing 250 (about 24%) random



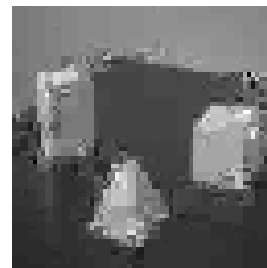
(a) Reconstruction error comparison curves.



(b) Original image.



(c) Reconstruction error 0.104.



(d) Reconstruction error 0.127.

Figure 3.2: Reconstruction error comparison.

measurements at the very beginning, and adaptive measurements as the rest. Each point on the curves is generated by averaging 50 repeats.

Figure 3.2(c) shows the reconstruction result of 200 adaptive measurements collected one by one and 250 random measurements as a basic portion. Figure 3.2(d) shows the reconstruction result of 450 random measurements. The reconstruction errors are 0.104 and 0.127 separately. As discussed in Section 3.3, reconstructions with adaptive measurements outperform, and the reconstruction can be obviously improved when

measurement percentage is relatively small. As measurement percentage increases, the error becomes very tiny, and the impact of adaptive measurements is not that significant.

3.4.2 Sampling & Reconstruction of Great Lakes Ice Cover

In the second experiment, a potential application is discussed, that is, to sample and reconstruct the ice cover of Great Lakes. The ice cover of Great Lakes has great impacts on many aspects of life, including fishing industry, commercial shipping, potential flooding, etc. As an important indicator of regional climatic conditions, research work on ice cover of the world's largest freshwater surface has been paid a huge amount of effort. Most current research work on ice cover monitoring is based on satellite radars with high resolution. However, to explore more details of the ice cover over Great Lakes, it has to be sensed in short range, and some specific areas may be required special emphasis.

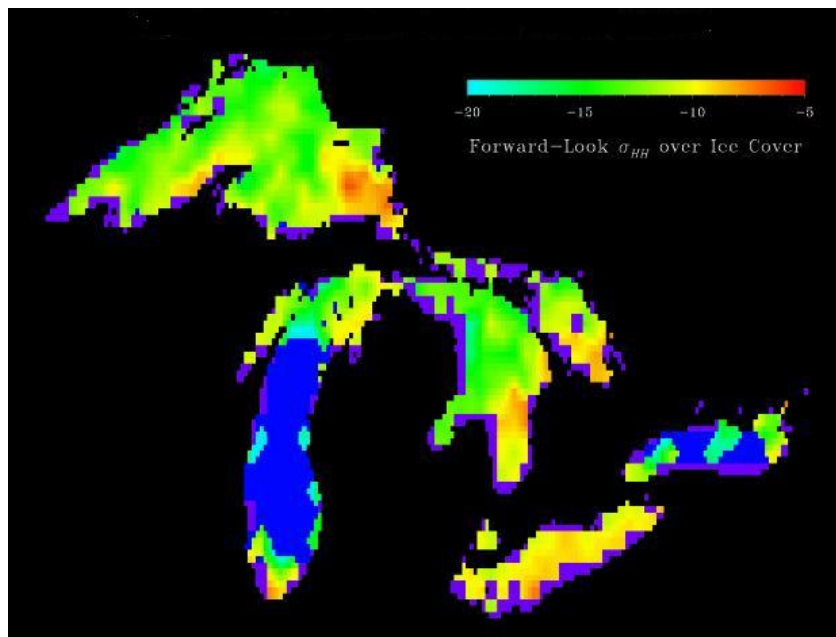


Figure 3.3: Great Lakes ice cover RGB image.

In this part of simulation, we fairly assume to monitor the Great Lakes ice cover using



Figure 3.4: Reconstruction of gray scale image.

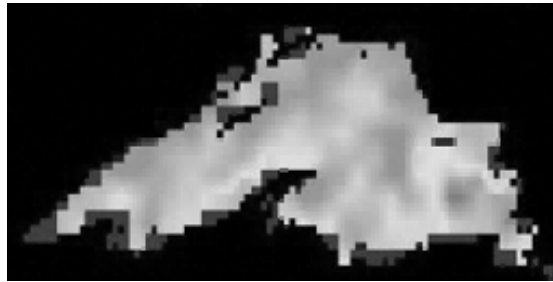


Figure 3.5: Lake Superior ice cover.

a mobile robotic sensor (UAV). Restricted by our experimental equipments, we only did simulations based on a NOAA image. Figure 3.3 shows the ice cover scenario of Great Lakes from NOAA web site ¹. In this simulation, it is supposed that a UAV can fly over the Great Lakes area and reconstruct the ice cover. To simplify the situation, the ice cover image in gray scale is used. The overall image is in the size of 576 by 448. In this scenario, the UAV explores the Great Lakes block by block with the size of 32 by 32. In each block, 200 random measurements are initially generated, and 200 more adaptive measurements sequentially determined and collected given existing measurements. Once enough measurements are collected for each block, a reconstruction algorithm is run to recover the ice cover image of current block. Then the UAV moves to next block and repeats the process.

Figure 3.4 shows the reconstructed ice cover based on the algorithm addressed in this paper. In this gray scale figure, the pixel value ranges from 0 to 255. The reconstruction error is 0.2204 under the assumption that there is no measurement noise. The moving cost is evaluated in this experiment. It is calculated under two conditions, totally random measurements, and optimal measurements with motion planning. Moving cost is calculated for each 32 by 32 block on average. Random measurements have the moving cost of (2.65×10^3) , while optimal measurements with motion planning have significantly smaller moving cost (0.80×10^3) .

At last of this simulation, our adaptive sampling method is compared with the two step adaptive sampling method addressed in [5]. In this part, we suppose to reconstruct the ice cover of Lake Superior as shown in Figure 3.5. In this scenario, gray scale image with the size of 128 by 256 is generated, whose pixel value ranges from 0 to 255. Gaussian noise is added into the measuring process to approximate real measuring situations. Each measurement is added $5 \times N(0, 1)$ noise for both methods. To implement adaptive sampling method in [5], 50% measurements are collected in total, with 25% for the coarse survey and 25% for the refinement. For the adaptive sampling method in this paper, the same amount of total measurements are collected, that is, 25% random measurements and 25% adaptive measurements. After reconstruction, the method addressed in this paper performs a little better with the reconstruction error 0.2758, while the other one has error of 0.3541.

¹<http://coastwatch.glerl.noaa.gov/overview/cw-overview.html#FIG1.5>

3.4.3 Indoor Mapping, a Practical Application

In this section, a real application has been carried out, where a mobile sensor is used for indoor mapping. A PIONEER 3-AT ² robot with an LMS200 ³ range finder is used to explore the 8th floor hallway of EERC Building at Michigan Technological University and generate a binary map, where doors for all rooms are closed. The mobile robotic sensor setup is shown in Figure 3.6. In this application, an embedded computer is used instead of the onboard computer on P3-AT to provide more computation capability, where MATLAB is running. Programs are started by remote control through wireless network, and the mobile sensor can automatically and adaptively move and generate the map.

This application carries on based on the adaptive sampling algorithm, and follows the steps as addressed. Due to the feature of mapping and mobile robots, experimental steps are slightly modified in this practical scenario, as shown in Algorithm 1. Mobile sensor will sample and reconstruct the unknown area block by block, and localize itself in the environment at the same time.

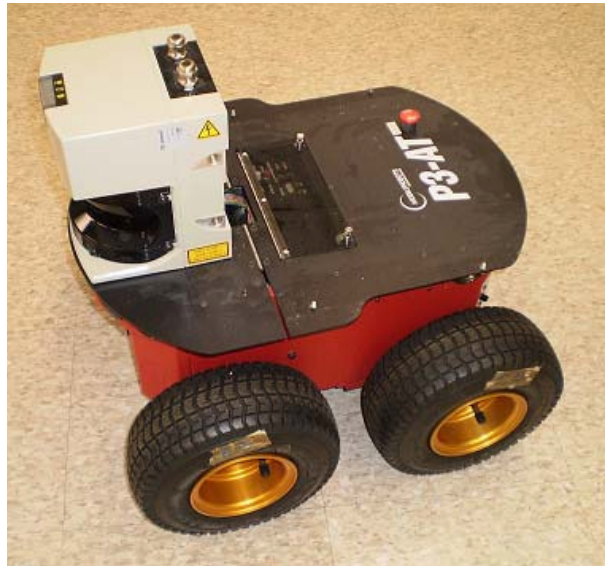


Figure 3.6: P3-AT robot with LS200 range finder.

²<http://www.mobilerobots.com/researchrobots/p3at.aspx>

³<https://www.mysick.com/eCat.aspx?go=FinderSearch&Cat=Row&At=Fa&Cult=English&FamilyID=344&Category=Produktfinder&Selections=50696,34243>

Algorithm 1 Algorithm for mobile compressive sensing

A. Sampling and reconstructing signals

B. Sparsity analysis

C. Adaptive movement

D. Image Stitching

E. Repeat A, B, ,C and D until it is done

3.4.3.1 Sampling and Reconstruction

An indoor environment is modeled as a 2D map with certain grids, $\mathbf{m} = \{m_{\alpha,\beta}\}$, where (α, β) is the coordinates. $m_{\alpha,\beta} = 1$ indicates the grid is occupied, while $m_{\alpha,\beta} = 0$ means free. For example, the hallway is free and the wall is occupied. We first adopt the normal approach of compressive sensing to sample and reconstruct the 2-D map, and then the adaptive sampling method.

According to the compressive sensing theory, the map signal is sampled in spatial domain, and recovered under Haar Wavelet domain, under which most natural signal is sparse. A random projection is involved to sample the signal in compressive sensing, and a random measurement matrix is generated as follows:

- a). Reshape \mathbf{m} to build a 1D vector of length N
- b). Collect M random measurements from \mathbf{m} , denoted as \mathbf{s}
- c). Build an $M \times N$ measurement matrix, Φ based on the measurement projections

Each of the measurements is a linear combination of the all elements in the sensing range with only one coefficient equal to one and others zero. The measurement matrix Φ is established by randomly sampling \mathbf{m} under the spatial domain. Once the measurements are collected, TV norm approach as shown in Equation (2.5) can be applied for recovery under Haar Wavelet domain.

The aim to express the original signal in another domain is to make it sparse so that it can be reconstructed. However, we have some interesting observations against this domain

transform. First, signal is sampled under a sparse domain. What we reconstruct is a binary map containing only 1 and 0 values. Each time the laser scanner is used to collect samples, there exists only one clear curve separating 1 value and 0 value parts of the map apart. Second, Haar Wavelet transition damages the sparsity actually. In spatial domain, the map contains only two values 1 and 0, and they have clear boundary. However, under Haar Wavelet domain, values of signal representation would vary in a relative big range, leading that when we consider the sparsity of signal, much information would be neglected. Moreover, when TV-norm method is applied to recover the map, which is to fill gaps of signals, the domain we are concerned is the spatial domain rather than the wavelet domain. Besides, measurements from LSM200 laser scanner can never be random due to the nature of laser scanner readings.

2-D TV-norm reflects the consistency of a figure. If a sparse basis is selected to transform the sensing data m into another domain for reconstruction, the 2-D TV-norm would account for consistency of the signal under that sparse domain rather than the original signal under spatial domain. Therefore, to take the best advantage of TV-norm, the signal should be reconstructed directly by minimizing the 2-D TV-norm of m in the spatial domain. Note that the mobile sensor used to rebuild the spatial map is laser scanner, which is a quite special sensor that senses the environment by shooting a bunch of laser rays from the sensor with equal angle interval. So random samples are not feasible to be covered. The measurement matrix is designed as:

- a). Reshape \mathbf{m} to build a 1D vector of length N
- b). Collect N measurements along rays emitted from the range finder
- c). Generate an $M \times N$ measurement matrix, Φ according to measurement projections.

In the later method, sampling and reconstructing are different from those in the normal compressive sensing method. Only one domain is selected in order to take better advantage of TV-norm method; a particular sampling pattern is used rather than random projection to suit to a particular sensor. In this application, a more efficient sampling and reconstructing process is used. One reconstruction is considered as one block or one piece. Once done, the mobile robot will move to somewhere else and do the same thing to reconstruct another piece.

3.4.3.2 Sparsity Analysis

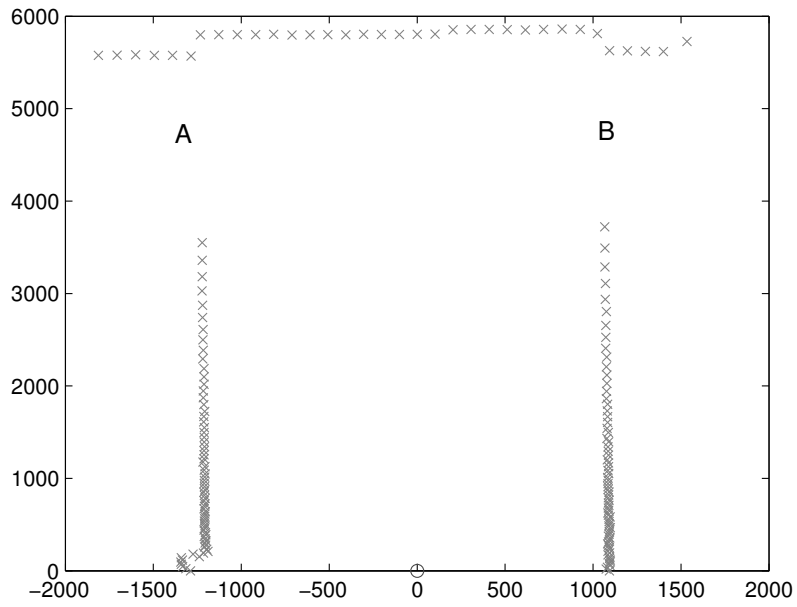
After one piece of map is reconstructed, more samples are about to be collected to expand the exploring and mapping. Existing sampled data will be analyzed to decide next sampling locations. As discussed above, signal can be analyzed under another domain, where differential entropy is maximized for new candidate measurements. This process usually requires a lot of computation. For this indoor mapping application, the situation becomes much easier.



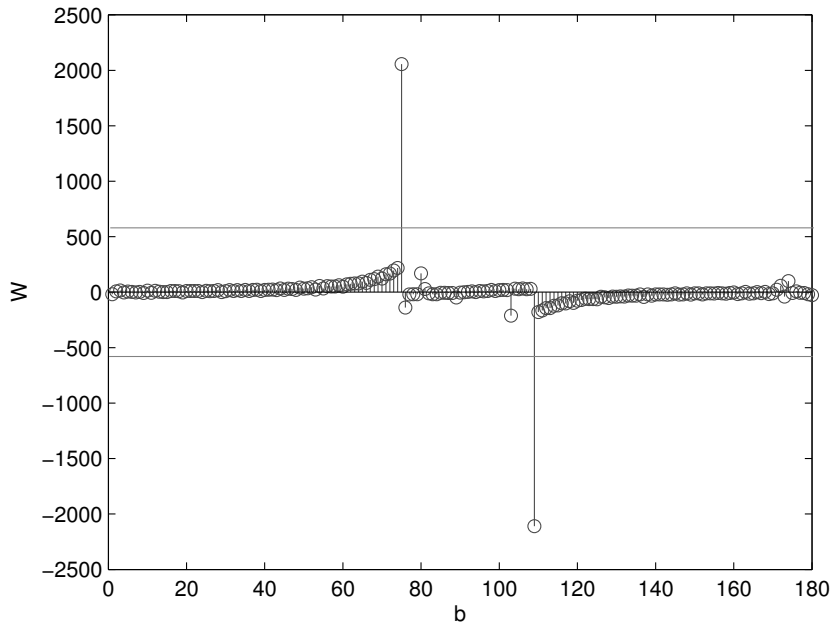
Figure 3.7: Binary map in adaptive sampling experiment.

For an indoor environment, most areas are smooth like walls, and some areas contain doors or at corner are relatively complicated. So for a 2-D mapping problem, we denote these relatively complicated areas as the sparsity of the 2-D signal, which require more attention to be paid. However, the sparsity would be different from the point of view of a mobile sensor. Some areas, usually door or corner areas, would be blocked from mobile sensor at different sensing positions. So the sparsity is redefined as blocked areas from mobile sensors.

In this application analysis is very straightforward. Laser scanner shoots and receives hundreds of lasers with one degree interval. Consistent returns indicate smooth areas, while sudden jumps show the areas that deserve more samples. Figure 3.7 shows a piece of binary map at one corner of the indoor environment, which can be covered by one laser scanning. 181 laser readings are shown in Figure 3.8(a), and Figure 3.8(b) shows the the sparsity clearly. For this scenario, the mobile sensor should move to position A and B to scan the environment. Another two analysis examples are shown in Figure 3.9. In this two examples, it clearly shows where the mobile sensor locates when sampling, where to



(a) Laser range finder returns.



(b) Difference of adjacent laser return pairs.

Figure 3.8: Totally 181 laser readings (red x) from the laser scanner (located at blue o) with 1 degree interval are received. A and B in the figure denote the areas requiring more samples. Unit for both the axes is millimeter.

collect additional samples, and which direction the mobile sensor should face.

After the analysis the 2-D problem has been simplified to a 1-D edge detection problem, in which only the map contour is considered, when analyzing the sparsity. The sparsity exactly reflects areas we want to apply the one step importance sampling. After new samples are collected, the 2-D spatial map is updated, and then a new round sparsity analysis and importance sampling would be applied.

3.4.3.3 Motion Planning

The sensing range of laser scanner used is 8 meters. It returns 8000 when out of its sensing range. Denote $E_i = \{e_j^i, j = 1, 2, \dots\}$ as the set of these areas out of the sensing range in the whole recovered map at sensing position p_i . Denote $L_i^k = \{l_k^i, k = 1, 2, \dots\}$ as the sparsity defined in last section for position p_i . These two set are kept updated to E_{i+1} and L_{i+1} as long as the 2-D map are updated.

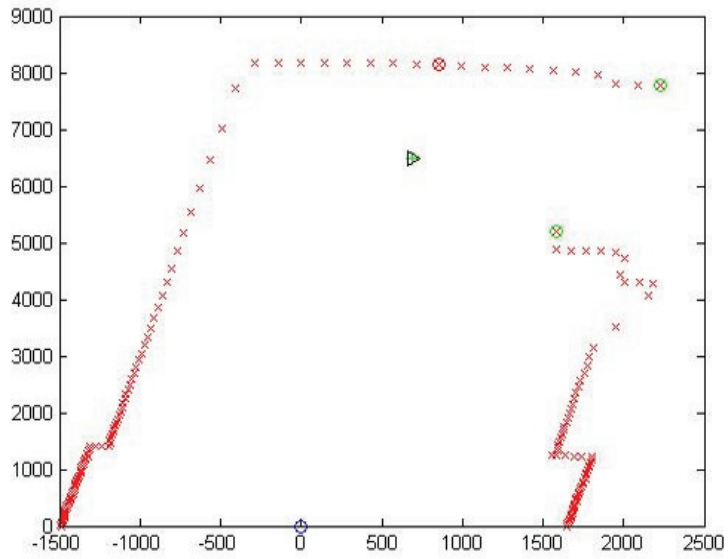
For an indoor hallway environment, we propose a straight line moving strategy, trying to avoid unnecessary movement, and the mobile sensor makes turns at proper positions to cover some sparsity areas. Our basic moving algorithm for mobile sensor is to move to cover all the important areas L_i and then move to the out-of-range areas E_i for more exploration.

One problem should be addressed here, that is, a loop route is detected, which may damage the tree structure. Compared with the existing map, it can be detected whether the mobile moves to somewhere it has been. If so, the current branch would stop, and the mobile sensor returns to the branch root. The moving strategy would go as shown in Algorithm 2.

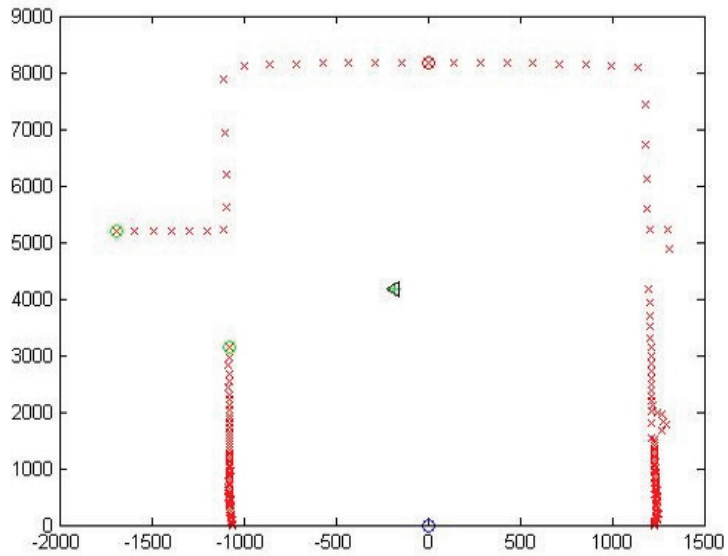
3.4.3.4 Map Stitching and Error Correction

As the mobile sensor moves, pieces of reconstructed maps are generated, however, motion errors are involved. In this section, a stitching methods is used to correct motion errors, so that pieces of maps can be merged together.

Encoders of a mobile robot can provide some motion information including the distance



(a) Laser returns analysis example 1. Unit: millimeter.



(b) Laser returns analysis example 2. Unit: millimeter.

Figure 3.9: Two additional laser return examples. Blue circle indicates mobile sensor sensing position; red x marks laser returns; green circles show the area requiring more measurements; triangle implies the next sensing position and direction.

Algorithm 2 Moving strategy

```
i = current
while  $E_i \cup L_i \neq \emptyset$  do
  if Loop Detected then
    End Branch ; Update  $E_i$  &  $L_i$ 
    Continue
  end if
  if Current Branch is Done then
    Return to Branch Root
    Continue
  end if
  if  $E_i \neq \emptyset$  then
    Go Straight to  $e_i^j$  and Cover Elements in  $L_i$ 
    Update  $E_i$  &  $L_i$ 
    Generate New Branch
  else
    Go Straight and Cover Elements in  $L_i$ 
    Update  $E_i$  &  $L_i$ 
    Generate New Branch
  end if
end while
```

the robot has covered, the angle it has turned. However, encoders does not always provide accurate information due to uneven wheels, slippery floors, or some enforced movement. If maps are stitched according to the position information returned from robot, a blent or even massing map would show. So in this section, we stitch pieces of maps together based on the encoder feedback, and correct motion errors using a correlation-based method [126].

When we have a new piece of map reconstructed, a proper position should be found in the exiting map. By comparing the overlap part between a new piece of reconstructed map and the existing map, maps can be merged together. Define Ω as the whole existing map, and $\omega(\alpha, \beta, \theta)$ as a new piece of map, where α , β , and θ are the coordinates and orientation parameters from robot, telling where this piece should be located in the whole map Ω . Since these position information is not accurate, correction should be added to the location information of $\omega(\alpha, \beta, \theta)$ to correct the motion error. Let $\omega(\alpha + \Delta\alpha, \beta + \Delta\beta, \theta + \Delta\theta)$ be the corrected new piece, and S be the overlap part between $\omega(\alpha + \Delta\alpha, \beta + \Delta\beta, \theta + \Delta\theta)$ and Ω . The corrected error can be obtained by

$$(\Delta a, \Delta b, \Delta \theta) = \arg \min_i \sum_i (\Omega_i - \omega(a + \Delta a, b + \Delta b, \theta + \Delta \theta)_i)^2, i \in S \quad (3.17)$$

where i indicated every grid of the overlap part.

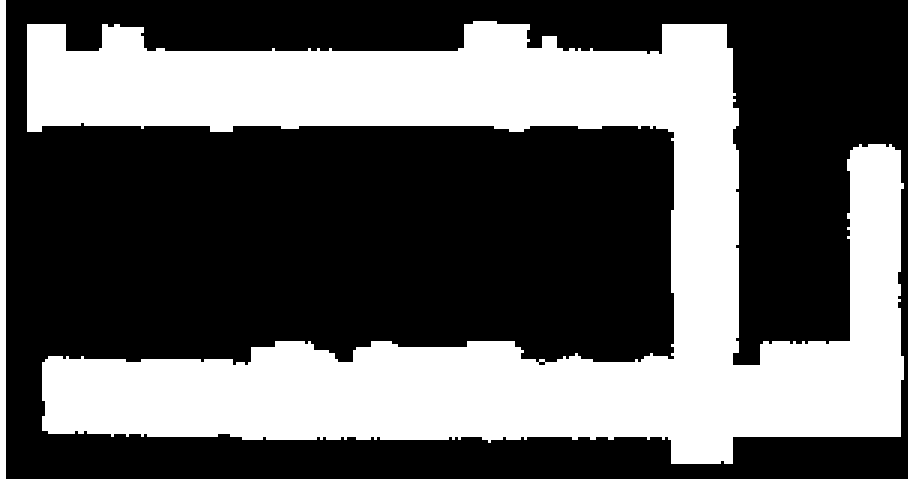


Figure 3.10: Binary map reconstruction.

In sum, compressive mobile sensing performs as a sequential process to move adaptively, sample, reconstruct, and stitch spatial maps. Once all the maps are stitched together, the 2-D mapping for the indoor environment is done. A 2-D map reconstructed is shown in Figure 3.10, where 27% samples have been used.

3.5 Conclusion

In this chapter, an adaptive approach is proposed to sample and reconstruct a given sensing field. This research work bases on a down sampling method compressive sensing, and explores the statistical model, where the Wavelet structure is exploited. This algorithm is designed for mobile sensors, and is categorized as a general method that can be applied in many cases. With a small portion of measurements collected, the sampling cost has been reduced. By maximizing the information amount of each measurement, the efficiency of the algorithm is guaranteed. In the simulation, it has been justified better performance compared with normal compressive sensing method and other adaptive sampling methods that collect excessive measurements. In the mapping application, the proposed method has been proved feasible in the real world.

Chapter 4

Adaptive Sampling with Mobile Sensor Networks¹

Based on the adaptive sparse sampling algorithm addressed previously, this chapter extends it to a much more complicated situation, mobile sensor networks. Adaptive sampling methods for mobile sensor networks will be proposed, as well as the corresponding real-time reconstruction for a scalar field of interest. The most informative measurements will be determined and collected by analyzing all the information throughout the entire mobile sensor network. Different from a single mobile sensor, collaboration between networked sensors is playing a critical role. Besides measurement collection and signal reconstruction, this chapter also presents information propagation, sensor fusion, network motion, etc. At last simulation results will show the efficacy and efficiency of this approach, where various scalar field are recovered. Number of measurements, path lengths, and reconstruction errors are assessed.

4.1 Introduction

With the development of various underwater, ground and aerial robots for sensing applications, mobile sensor networks are becoming more and more important in sampling and recovering a certain sensing field. For example, underwater vehicles are investigated [2] to adaptively determine and collect useful measurements by optimizing an objective

¹©2012 IEEE. Portions reprinted with permission, from **Shuo Huang** and Jindong Tan, “*Adaptive Sampling Using Mobile Sensor Networks*” , in Proceedings of IEEE International Conference on Robotics and Automation, pp. 657 - 662, 2012. See Appendix for a copy of the copyright permission from IEEE.

function. A mobile sensor network [6] is deployed on a lake to reconstruct the temperature field of the lake surface. Vision-based UAVs collaborate to extract and match environmental features for forest fire detection [127]. In all of these applications, it is critical to efficiently planing the motion of the network according to the feature of the sensing field. For instance, the most useful measurements are collected in [2, 6], which are considered to diminish the sensing field uncertainty to the most. Particular features have been captured [127], which can represent the target sensing field. These applications suggest to determine and collect measurements that can best represent the target sensing field. Those measurements are considered as the most informative measurements. The way to maximize network utility efficiency is to actively determine measurements of high information amount, i.e., only the most informative data in the sensing field is attractive. It leads to a specific topic in sensing resource allocation using mobile sensor networks, measurement evaluation, determination and collection.

To explore an unknown field using a mobile sensor network, many methods could be used. Either uniform sampling or random sampling introduced by compressive sensing could be fine. However, with a mobile sensor network, it becomes a motion planning problem to cover all the pre-computed waypoints along the sensors path. The unique feature of a mobile sensor network is its flexibility to reconfigure the network topology in response to the sensing field. Sampling density and locations can be adjusted according to sensing demands. Different areas of importance have been paid different attention For example, rapid changing areas deserve high sampling density. However, the information of the sensing filed is not known in prior, and therefore the sampling pattern can not be pre-determined. As discussed in literatures [2, 6, 7, 128, 9], measurements should be collected sequentially, and measurement distribution keeps changing as sampling proceeds. Old measurements give feedback to determine new measurements, and new measurements are collected to more accurately represent sensing fields. Networked mobile sensors collaborate to determine new measurements by gathering and analyzing all the available information through the network. The way for a mobile sensor network to actively respond to the environment is to establish a feedback system, which can determine the most informative measurements. This chapter will be discussing such an adaptive sampling algorithm for mobile sensor networks, where networked mobile sensor deployed over sensing field will collaborate and seek for the most informative measurements to represent the target sensing field with a sparse representation.

4.2 Problem Formulation

Suppose a target sensing field to be reconstructed is X . By down sampling X by mobile sensors, a set of condensed measurements can be acquired, denoted as Y . The projection

is built as $Y = \Phi X$, mapping target signal X from higher dimension to lower dimension. With signal dimension reduced, it requires each measurement contains more information. In this chapter, a mobile sensor network of n_v sensors, denoting as $v_i, i = 1..n_v$, will collaborate to collect the most informative measurements and recover an unknown sensing field. These sensors are maintaining a connected graph which is defined as G . As the networked sensors move, the graph will change dynamically. An important assumption is made that communication links between mobile sensors are always reliable, and mobile sensors are connected.

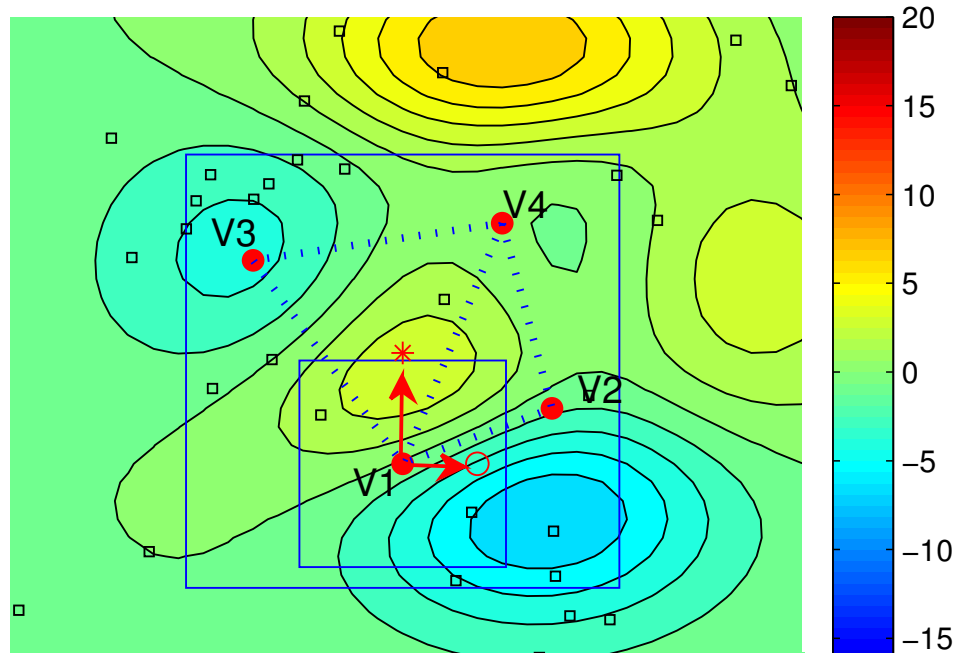


Figure 4.1: An illustrative example of informative measurement collection with mobile sensor networks. Networked mobile sensors deployed over a scalar field. \bullet : networked mobile sensors. \square : existing measurements. \cdots : network connections. \circ : potential measurement determined given information from v_1 . $*$: potential measurement determined given shared information from the mobile sensor network.

In an mobile sensor network, sensors work parallel without timing synchronization when collecting measurements. A sensor collects a new measurement when it can tell where the most informative one is located. Suppose at time instant t , the sensor v_i at position p_i^t is about to determine and collect a new measurement. The specific goal at time instant t is to determine and collect a measurement \hat{y}_i^{t+1} . This measurement should be the most informative one given all the available information, including sensed data by

itself and shared data from other networked sensors. Figure 4.1 intuitively explains the informative measurement determination with network collaboration, where $v_i, i = 1 \dots 4$ is denoting networked sensors, and the sensor v_1 is determining a new measurement. The sensor collaboration can affect the efficiency of the network. The difference can be seen between the measurement determined with and without network collaboration. In this scenario, each sensor has limited awareness of the environment. Suppose v_1 is aware of the measurements within the small box. Without exchanging information with other sensors, a potential measurement may be determined at \bigcirc , because it is the rapid changing area, which deserves more measurements. This measurement is regarded the most informative measurement. However, it differs if sensors collaborate to share information, as v_1 can be aware of what the mobile sensor network can observe, the large box area. With information exchanged, v_1 may determine the potential measurement at $*$, whose impact to sensing field appears stronger, where the sensing field changing trend is changed. This measurement is the most informative one in a more global view, which should contain more information compared with the formerly determined one. It leads to the most important problem encountered in the active sampling algorithm using a mobile sensor network, determining the most informative measurements. Other problems include information fusion, network motion, and real-time reconstruction.

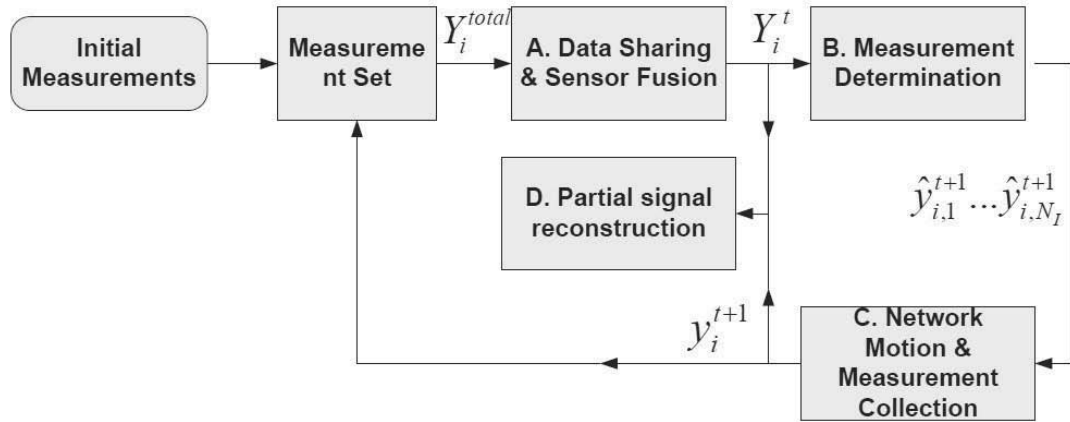


Figure 4.2: Diagram of adaptive sampling algorithm.

The adaptive sampling proposed in this chapter is a feedback driven method using previous measurements to guide mobile sensors to collect new measurements. Figure 4.2 outlines the algorithm for each sensor in the mobile sensor network, supposed to be v_i . Without any knowledge of a target sensing field, a small number of random measurements are collected

initially. The feedback loop starts with the measurement set, defined as $Y_{i,total}^t$, that is generated by merging $Y_{i,total}^{t-1}$, shared information, and newly generated measurements in previous loops. Sensors establish a network and fuse information in block **A**, generating measurement set Y_i^t that can be used to determine new measurements. Block **B** determines a set of measurement $\hat{y}_{i,1}^{t+1} \dots \hat{y}_{i,N_I}^{t+1}$ as the most informative ones given Y_i^t , where N_I is the number of informative measurements. Newly generated measurements are reevaluated, and one of them is collected, denoted as y_i^{t+1} , after sensor motion in block **C**. It is sent to both $Y_{i,total}^{t+1}$ to update the measurement set and block **D** for partial signal reconstruction after merged with Y_i^t .

A. Data sharing & sensor fusion: One of the advantages of mobile sensor networks is networked sensors can share information with others. The aim to share data is to increase the awareness of surrounding area, and it may result different new measurements determined. It is a research problem to efficiently transmit data, avoiding communication jam. Section 4.3.1 will address details on information sharing, including network graph establishment, sensor localization, and information fusion.

B. Measurement determination: As the down-sampling method is used in this paper, it requires every measurement contains as much information as possible. The specific problem is to quantify and analyze the information amount and determine the most informative measurements. In Section 4.3.2, the inference of target signal in a sparse manner is addressed, and a metric measuring the information amount is defined to describe sensing field uncertainties.

C. Network motion: With new measurements determined for a mobile sensor, it is a practical problem to reach the designed positions. In Section 4.3.3, moving constraints will be defined as to impose a minimum separation to avoid collision between mobile sensors and a decentralized moving algorithm will be given.

D. Partial signal reconstruction: Compressive sensing transfers the burdens from sampling to signal reconstruction. It is quite computation and time consuming to reconstruct signals, especially for large scale signals. In Section 4.3.4, a partial signal reconstruction scheme is proposed to save computation and time.

4.3 Adaptive Sampling with Mobile Sensor Networks

This section explains the adaptive sampling algorithm corresponding to Figure 4.2. This algorithm is built on compressive sensing theory [4]. Compressive sensing reconstructs

target signals from incomplete measurements. The target signal is supposed to be X with the dimension of $N \times 1$. A measurement matrix Φ ($M \times N$) projects the signal from higher dimension (N) to lower dimension (M) with $M < N$, subject to $Y = \Phi X$. This system cannot be solved, since there are more unknown variables than equations. X can be sparsely represented under a sparse domain Ψ , $X = \Psi^{-1}s$. s is a sparse signal of dimension $N \times 1$ with the sparsity K . There are only K significant entries in s , and $N - K$ zero or very small entries that can be ignored. Usually, the K is far smaller than the N and M . Therefore, the basic compressive sensing equation is given

$$Y = \Phi X = \Phi \Psi^{-1}s \quad (4.1)$$

After being sparsely represented under another domain, the system can be solved through convex optimization, since the sparse s has less unknown entries than equations. Only K unknown variables have to be dealt with rather N , thus reducing the signal complexity in storing and transmitting. The Haar wavelet domain is used as the sparse domain in this paper. In the following description, Ψ is exactly the Haar wavelet basis.

In this dissertation, the compressive sensing model can be explained in two aspects, global and local model. The global compressive sensing model is shown in Equation (4.1) to represent the entire sensing field. Yellow dots in Figure 4.3 are condensed measurements collected by the mobile sensor network previously. If the entire sensing field is X , all measurements in the figure correspond to Y in the global compressive sensing model in Equation (4.1). The local compressive sensing model is applied on each sensor of the mobile sensor network for measurement determination and signal reconstruction. Suppose the sensor centered in the red box of Figure (4.3) is v_i , located at p_i^t . The red box marks the local compressive sensing area. The target signal located in the red box is denoted as X_i^t at time t instead the X in Equation (4.1), while measurements in the box correspond to Y_i^t in the local model, yielding

$$Y_i^t = \Phi_i^t X_i^t = \Phi_i^t \Psi^{-1} s_i^t \quad (4.2)$$

where Φ_i^t is the measurement matrix for sensor v_i at time t , and s_i^t is the sparse representation correspondingly. In this paper, we define the local compressive sensing area (red box in Figure 4.3) as $A_{RB,i}^t$. The local compressive sensing above model is mainly used when determining the most informative measurements. In reality, measurement noise has to be added into the system, yielding

$$Y_i^t = \Phi_i^t X_i^t + n_0 = \Phi_i^t \Psi^{-1} s_i^t + n_0 \quad (4.3)$$

where n_0 is the measurement noise, modeled as zero mean Gaussian variable with precision α_0 , $n_0 \sim N(0, \alpha_0^{-1})$.

In compressive sensing, measurements are usually generated by random projections.

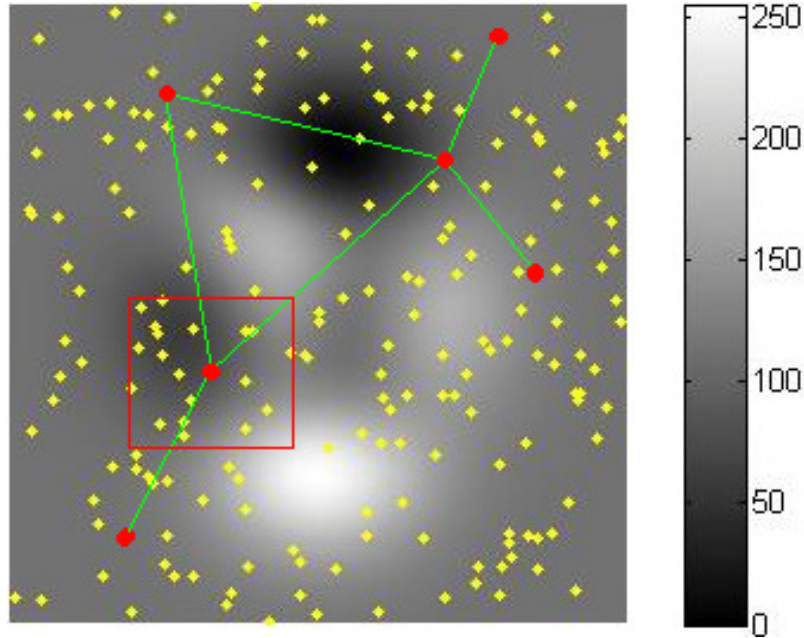


Figure 4.3: An mobile sensor network deployed in a scalar field. Red dot: sensors, green line: network connection, yellow dot: measurements.

Random projections indicate measurements have equal weights in the unknown sensing field. However, in real applications, this may not be appropriate. For example, in a scalar field, flat areas require less measurements, while rapid changing areas deserve more measurements (informative measurements). This explains the adaptivity that is proposed in this paper. The following sections will explicitly explain the proposed algorithm, which uses an mobile sensor network to collect the most informative measurements and reconstruct an unknown sensing field.

4.3.1 Data Sharing and Sensor Fusion

Using an mobile sensor network to explore an unknown area, sensors have to register themselves with each other to maintain a uniform coordinate system. This part explains sensor alignment and information fusion. To achieve energy efficiency, decentralized algorithms are developed. Both of the two issues are based on an existing network, so the first problem is to define the network topology. Figure 4.4 explains input, output and detailed steps of block **A** in Figure 4.2, and the following part explains these three steps.

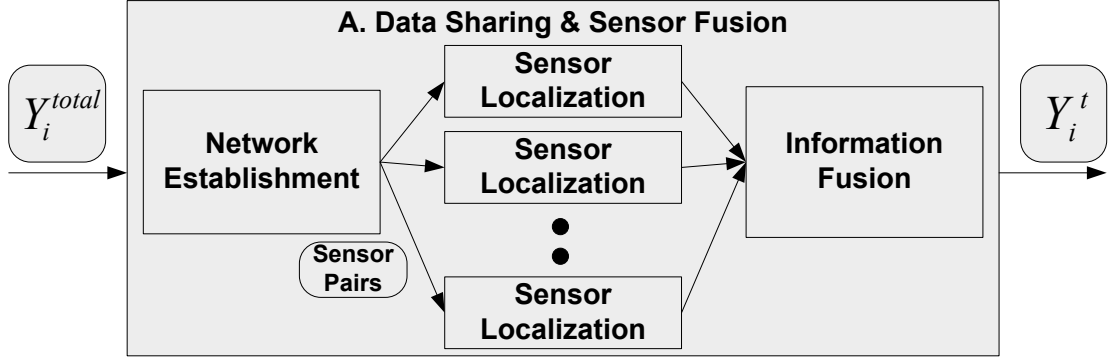


Figure 4.4: Data sharing diagram.

The first is to establish the network topology. The relative neighborhood graph (RNG) [129] is used. With a number of n_v mobile sensors $v_i, i = 1, 2, \dots, n_v$, the topology of the mobile network can be defined as a bidirectional graph $G(V, L)$, where $V = \{v_i, i = 1 \dots n_v\}$ denotes the mobile sensors, and $L = \{L_{i,j}\}$ denotes the communication links. $L_{i,j}$ exists if and only if sensor v_i and v_j are connected directly. The advantage of using graph definition is to uniquely define the data sharing and information fusion process in mobile sensor networks. Though a mobile sensor v_i may connect with many others directly or indirectly, it will only share data with its immediate neighbors $v_{N(i)}$ defined by graph G . This will significantly reduce the communication cost and lower the impact of losing a sensor in the mobile sensor network.

Sensors in a mobile sensor network have to precisely localize each other before exchanging information; otherwise, information can never be fused. It would be easy to localize sensors when GPS is available. For the scenarios without GPS, sensor localization algorithms are developed. Imaging sensors equipped on mobile sensors can capture images that contain invariant features in the sensing field. The invariant features are used to accurately localize mobile sensors from each other mutually. If one or more invariant features are captured by two sensor at the same time, indicating they have imaging overlap, they can localize each other. Otherwise, there is no overlap, thus they cannot be localized. We are only interested in the relative positions between clustered mobile sensors, where imaging overlap exists. Define the links between two sensors with imaging overlap as $OL = \{OL_{i,j}\}$, $OL \subset L$. $OL_{i,j}$ exists, if and only if v_i and v_j are directly connected and have imaging overlaps.

Scale-invariant feature transform (SIFT) [130] is used to extract the invariant features in different sensor coordinate systems. These features are regarded as references to fuse two sensors together. Figure 4.5 shows a typical example of SIFT, where invariant features can be extracted by both of the sensors, and have been highlighted. It becomes an image stitching problem when localizing sensors with each other. An open source MATLAB

toolbox ² can extract features and match them. Then v_i and v_j can be localized in the same coordinate system, either in the coordinate system of v_i or v_j . Each time network topology is newly established, neighbor sensors defined by OL have to localize each other. To avoid unnecessary fusion, an algorithm comes out as Algorithm 3.

Algorithm 3 Neighbor sensor localization

Require: $\exists OL_{i,j}$

Ensure: v_i and v_j are localized

```

while  $v_i$  and  $v_j$  have not been localized do
    if  $i < j$  then
        localize  $v_j$  in  $v_i$ 's coordinate system
    else  $\{i > j\}$ 
        localize  $v_i$  in  $v_j$ 's coordinate system
    end if
end while

```

Data sharing can be performed that each mobile sensor can share information with others by transmitting its measurements with accurate positions. Data sharing only happens between two sensors v_i and v_j , where $OL_{i,j}$ exists. A local data sharing scheme is developed. One question has to be identified, that is, what to share. For sensor v_i at position p_i^t , it will collect all the available information in $A_{RB,i}^t$ from its neighbors, while for every sensor v_j , where $OL_{i,j}$ exists, it sends all its measurements located in $A_{RB,i}^t$ to sensor v_i . v_i will choose measurements located in $A_{RB,i}^t$ from its measurement set and merge them together with measurements sent from its neighbors. Suppose at time instant t , v_i stores measurements $Y_{i,total}^t = \{y_{i,1}^t, y_{i,2}^t \dots y_{i,n_{vit}}^t\}$, where n_{vit} is the measurement index. The block

²<http://www.vlfeat.org/mdoc/mdoc.html>

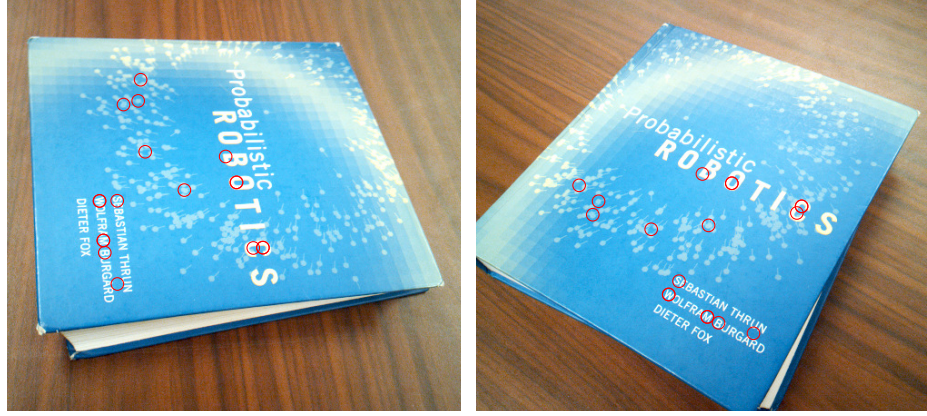


Figure 4.5: SIFT example.

output illustrated in Figure 4.4 (Y_i^t) can be retrieved

$$Y_i^t = \left\{ y_{i,k}^t \mid k = 1 \dots n_{vit}, y_{i,k}^t \in A_{RB,i}^t \right\} \bigcup_j^{\exists OL_{i,j}} \left\{ y_{j,k}^t \mid k = 1 \dots n_{vjt}, y_{j,k}^t \in A_{RB,i}^t \right\} \quad (4.4)$$

where $y_{i,k}^t \in A_{RB,i}^t$ indicates the measurement $y_{i,k}^t$ is located in $A_{RB,i}^t$. The measurement set $Y_{i,total}^t$ is updated as $Y_{i,total}^t = \{y_{i,1}^t, y_{i,2}^t \dots y_{i,n_{vit}}^t\} \cup Y_i^t$. The measurement set Y_i^t fits the local compressive sensing model in Equation (4.3), and will be used to determine new measurements.

4.3.2 Determining the Most Informative Measurement

This section deals with the problems on how to determine the new measurements by quantifying and maximizing the information amount. A mobile sensor v_i determines the most informative measurements given the measurement set Y_i^t within the surrounding area $A_{RB,i}^t$. Local compressive sensing model in Equation (4.3) is applied. In the local compressive sensing model, condensed measurements are collected, leaving the unknown sensing field filled with uncertainty. The uncertainty indicates a state that has limited knowledge to exactly describe an existing state. In this case, it is a lack of measurements

to represent the sensing field X_i^t defined in the $A_{RB,i}^t$ area. With the knowledge of Y_i^t , we have the uncertainty of X_i^t at certain degree. A new measurement within $A_{RB,i}^t$ can result in describing the sensing field more accurately, meaning that it can reduce the uncertainty to some extent. The most informative measurement in this paper exactly indicates the measurement that can diminish the uncertainty to the most extent. In the same domain where measurements are collected, it is difficult to analyze the uncertainty, because the impact of a potential measurement is unknown. Thus, in this section, a sparse domain is used when analyzing the uncertainty, which is the Haar wavelet domain in Equation (4.3). Corresponding to the block **B** in Figure 4.2, Figure 4.6 explains the how to determine the most informative measurement by two steps, which express the uncertainty under a sparse domain and choose a measurement that can maximizes it.

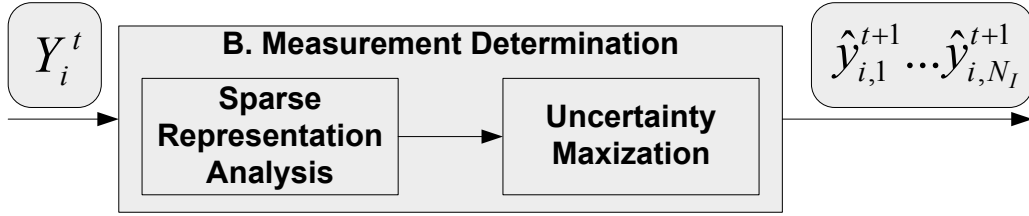


Figure 4.6: Measurement determination diagram.

Based on the local compressive sensing model in Equation (4.2), statistical features are studied. s_i^t is a sparse signal, which is a sparse representation of X_i^t . The statistical features of a sparse signal have been well studied in [115], where s_i^t is modeled as a multiple Gaussian distribution.

$$s_i^t \sim N(\mu_i^t, \Sigma_i^t) \quad (4.5)$$

If a zero covariance s_i^t exists, s_i^t is certain, so is the signal X_i^t from the linear relationship in Equation (4.2). Otherwise, the covariance Σ_i^t indicates the extent of uncertainty. Define a potential measurement as \hat{y}_i^{t+1} with its row vector projection $\hat{\phi}_i^t$, which would be a potential new row of Φ_i^t . The goal in this part is exactly to find $\hat{\phi}_i^t$ that can reduce the uncertainty to the most. Therefore, the first step is to infer mean μ_i^t and covariance Σ_i^t from incomplete measurements Y_i^t in Figure 4.6, and the second step is to determine $\hat{\phi}_i^t$ by maximizing the differential entropy of \hat{y}_i^{t+1} .

The inference of μ_i^t and Σ_i^t of sparse signals have been studied in [115]. \hat{y}_i^{t+1} can be written as Gaussian in terms of μ_i^t and Σ_i^t

$$\hat{y}_i^{t+1} \sim N(\hat{\phi}_i^{t+1} \Psi^{-1} \mu_i^t, \hat{\phi}_i^{t+1} \Psi^{-1} \Sigma (\hat{\phi}_i^{t+1} \Psi^{-1})^T) \quad (4.6)$$

This indicates the potential measurement can be inferred from the sparse representation.

Given a measurement set Y_i^t , the posterior distribution of s_i^t is given

$$p(s_i^t | Y_i^t, \alpha, \alpha_0) = \frac{p(Y_i^t | s_i^t, \alpha_0) p(s_i^t | \alpha)}{p(Y_i^t | \alpha, \alpha_0)} \quad (4.7)$$

where $\alpha = \{\alpha_1, \alpha_2 \dots \alpha_N\}$, and N is the dimension of s_i^t . α_k , $k = 1 \dots N$ corresponds to the Gaussian precision of each entry in s_i^t , and α_0 is measurement noise as shown in Equation (4.3). $p(s_i^t | \alpha)$ is Gaussian distribution from Equation (4.5). $p(Y_i^t | s_i^t, \alpha_0)$ is generated given s_i^t , $N(\Phi_i^t \Psi^T s_i^t, \alpha_0^{-1} I)$. $p(Y_i^t | \alpha, \alpha_0)$ is a likelihood function for α and α_0 , being given

$$\begin{aligned} \mathcal{L}(\alpha, \alpha_0) &= p(Y_i^t | \alpha, \alpha_0) \\ &= \int p(Y_i^t | s_i^t, \alpha_0) p(s_i^t | \alpha) ds_i^t \\ &= (2\pi)^{-\frac{N}{2}} \left| \frac{I}{\alpha_0} + \Phi A^{-1} \Phi^T \right| \exp \left\{ \frac{I}{\alpha_0} + \Phi A^{-1} \Phi^T \right\} \end{aligned} \quad (4.8)$$

where $A = \text{diag}(\alpha_1, \alpha_2 \dots \alpha_N)$. Since the three distributions in Equation (4.7) are known, the posterior distribution of s_i^t can be calculated. Given a set of measurements Y_i^t , $s_i^t | Y_i^t, \alpha, \alpha_0 \sim N(\mu, \Sigma)$ with posterior mean and variance, yielding

$$\mu_i^t = \alpha_0 \Sigma_i^t (\Phi \Psi^{-1})^T Y_i^t \quad (4.9)$$

$$\Sigma_i^t = (\alpha_0 (\Phi \Psi^{-1})^T \Phi \Psi^{-1} + A)^{-1} \quad (4.10)$$

With the posterior mean and variance calculated, it is worth noting that the parameters in Equation (4.8), which is a likelihood function, can be optimized in order to infer new precisions given the posterior mean and variance. To optimize parameters in Equation (4.8) is to maximize $\mathcal{L}(\alpha, \alpha_0)$, so that the best α and α_0 can be found for the distribution. This is known as the type-II maximum likelihood, and can be implemented through differentiation, being given in [115]. The new parameters yield:

$$\alpha_k^{new} = \frac{\gamma_k}{(\mu_i^t)_k^2} \quad (4.11)$$

$$1/\alpha_0^{new} = \frac{\|Y_i^t - \Phi_i^t \Psi^{-1} \mu_i^t\|_2^2}{N - \sum_{k=1}^N \gamma_k} \quad (4.12)$$

where $\gamma_k = 1 - \alpha_k \Sigma_{kk}$. An iterative algorithm can be executed by alternating between Equation (4.9) (4.10) and Equation (4.11) (4.12). It converges very fast, and α_k becomes large for zero or small entries of s_i^t . Suppose the iteration number is N_{iter} . α_0 , α_k , and N_{iter}

will be set initial values in simulations.

Once the distribution of s_i^t is calculated, it would be an easy task to infer the new measurement \hat{y}_i^{t+1} , subject to

$$\hat{y}_i^{t+1} = \hat{\phi}_i^{t+1} \Psi^{-1} s_i^t \quad (4.13)$$

The only unknown factor is $\hat{\phi}_i^{t+1}$, which is going to be a new row added into the projection matrix. To calculate the $\hat{\phi}_i^{t+1}$ is a maximization problem. The information amount of the new measurement \hat{y}_i^{t+1} can be quantified by its differential entropy

$$H(\hat{y}_i^{t+1}) = - \int p(\hat{y}_i^{t+1}) \log p(\hat{y}_i^{t+1}) dy \quad (4.14)$$

The larger this quantity is, the more information it can contain. The desired measurement can be determined by maximizing

$$\hat{\phi}_i^{t+1} = \arg \max_{\phi} H(\hat{y}_i^{t+1}) \quad (4.15)$$

which is the locally optimized measurement. The distribution of \hat{y}_i^{t+1} can be calculated from Equation (4.13)

$$\hat{y}_i^{t+1} \sim N(\hat{\phi}_i^{t+1} \Phi^T \mu, C_y) \quad (4.16)$$

where $C_y = \hat{\phi}_i^{t+1} \Phi^T \Sigma(\phi_{next} \Phi^T)^T$. This maximization problem can be simplified as to maximize C_y , which is exactly the variance of the potential measurement. It implies that the information amount of a potential measurement can be explained in terms of its variance. This also explains the relationship between the uncertainty and the most informative measurement. In this section, a few measurements of the most information are determined, $\{\hat{y}_{i,1}^{t+1} \dots \hat{y}_{i,N_I}^{t+1}\}$, where N_I is the number of potential informative measurements. One of these measurements will be chosen and collected by leveraging the moving cost in the next section.

4.3.3 Network Motion

In the above description, N_I most informative measurements have been determined. Figure 4.7 explains how to plan motion and collect measurements. One of these measurements will be collected by the sensor v_i . It has to be determined not only subject to the maximal information amount but also the moving cost from current position to the newly determined measurement position. After being reevaluated, one measurement is chosen. v_i will move

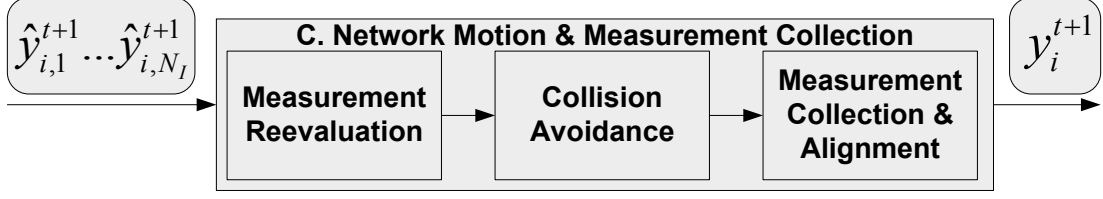


Figure 4.7: Motion control diagram.

to collect that measurement. During the movement, it has to avoid any possible collision with other sensor in the mobile sensor network.

To reevaluate the newly determined measurements, Equation (4.15) should be changed by adding a moving cost item.

$$\hat{\phi}_i^{t+1} = \arg \max_{\phi} (H(\hat{y}_i^{t+1}) - \omega C(\phi_i^t, \hat{\phi}_i^{t+1})) \quad (4.17)$$

where the first item remains the same as Equation (4.14), and the second item represents the moving cost. $C(\cdot, \cdot)$ is a function indicating the moving cost between the two measurement positions, and $C(\phi_i^t, \hat{\phi}_i^{t+1})$ represents the moving cost from current sensing position to the next one. ω is a positive weighted factor that is used to achieve a proper ratio for these two items. This parameter can adjust the relationship between the information amount of measurements and power consumption. Small ω indicates the emphasis on information amount, while large ω is used when moving cost is dominant. In the real applications, many other factors have to be considered, including mobile sensor turning, obstacles, collisions, etc. Different types of sensors have different moving cost. In the simulation addressed later, the moving cost will be defined.

Once one measurement is determined by leveraging sensor moving cost, it is time to move sensor to collect that particular measurement. The only constraint is collision with other sensors. The mobile sensor network keeps the d as the safety margin for sensors. At any time t , the dynamic collision avoidance constraint is

$$\|p_i^t - p_j^t\| \geq d \quad \forall i, j \in \{1, \dots, n_v : j \neq i\} \quad (4.18)$$

where $\|p_i^t - p_j^t\|$ is the real-time distance between mobile sensors indexed by i and j . In the simulation, parameter d will be given. When v_i achieve the designed position, the measurement determined can be collected. The collected measurement is supposed to be y_i^{t+1} . The newly generated y_i^{t+1} is sent to partial signal reconstruction, and the measurement set of v_i is then updated as $Y_{i,total}^{t+1} = Y_{i,total}^t \cup y_i^{t+1}$, which is used in a new round for block **A**.

4.3.4 Partial Signal Reconstruction

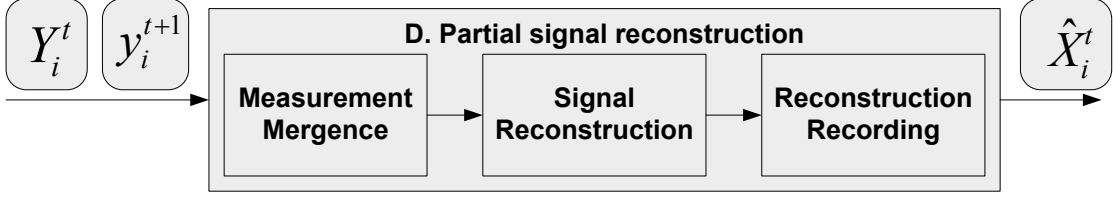


Figure 4.8: Partial signal reconstruction diagram.

To reconstruct signal from incomplete measurements is a time consuming procedure, especially for large scale signals. In this section, signals are reconstructed in small scale. Figure 4.8 shows the partial signal reconstruction. The newly collected measurement y_i^{t+1} is added into Y_i^t to reconstruct the sensing field with in $A_{RB,i}^t$. In the reconstruction, the local compressive sensing model is written as

$$\begin{bmatrix} Y_i^t \\ y_i^{t+1} \end{bmatrix} = \begin{bmatrix} \Phi_i^t \\ \phi_i^{t+1} \end{bmatrix} X_i^t + n_0 = \begin{bmatrix} \Phi_i^t \\ \phi_i^{t+1} \end{bmatrix} \Psi^{-1} s_i^t + n_0 \quad (4.19)$$

Some existing reconstruction methods include l_1 norm [89], basis pursuit [104], matching pursuit [131], gradient pursuit [132], etc. For 2-D signals, an example is given to reconstruct via total variation (TV) norm [88]. In this paper, a statistical method [118] with efficient inference via Markov chain Monte Carlo (MCMC) sampling is used. Under Haar Wavelet domain, signals appears not only sparse but also structured. The wavelet-tree structure [112] provides a powerful tool to explore signal structure under Haar Wavelet domain. Sparse signals have dependency between different Haar wavelet scales. By exploiting wavelet structure, the method addressed in [118] is able to reconstruct signals efficiently. Suppose the signal reconstruction is \hat{X}_i^t . \hat{X}_i^t is stored in sensor v_i .

As the adaptive sampling algorithm carries on, a sensor v_i in the mobile sensor network stores reconstruction pieces $\hat{X}_i^1, \hat{X}_i^2 \dots \hat{X}_i^t$. These reconstruction pieces are merged together, $\hat{X}_i = \{\hat{X}_i^\tau | \tau = 1 \dots t\}$. In \hat{X}_i , some entries of X have been reconstructed multiple times, while some are not even been covered. For the multiple reconstructed entries, mean values are calculated. This is the reconstruction portion from sensor v_i , which may just cover a small area of the entire sensing field. Once the algorithm is terminated, all the reconstruction pieces from the entire mobile sensor network are collected $X_{REC} = \{\hat{X}_i^\tau | i = 1 \dots n_v, \tau = 1 \dots t\}$. Each entry of X should be recovered multiple times by the X_{REC} . Suppose \hat{X}_k in the global compressive sensing model X and $\hat{x}_{i,j}^t$ in the local compressive sensing model X_i^t reflect the same pixel, thus denoted as $k = f(i, j, t)$. The final reconstruction is generated

by averaging all the reconstructed elements. Each entry of the reconstruction, \hat{X} , can be calculated

$$\hat{x}_k = \text{mean}\{\hat{x}_{i,j}^\tau | i = 1 \dots n_v, \tau = 1 \dots t, k = f(i, j, t)\} \quad (4.20)$$

4.4 Experimental Results

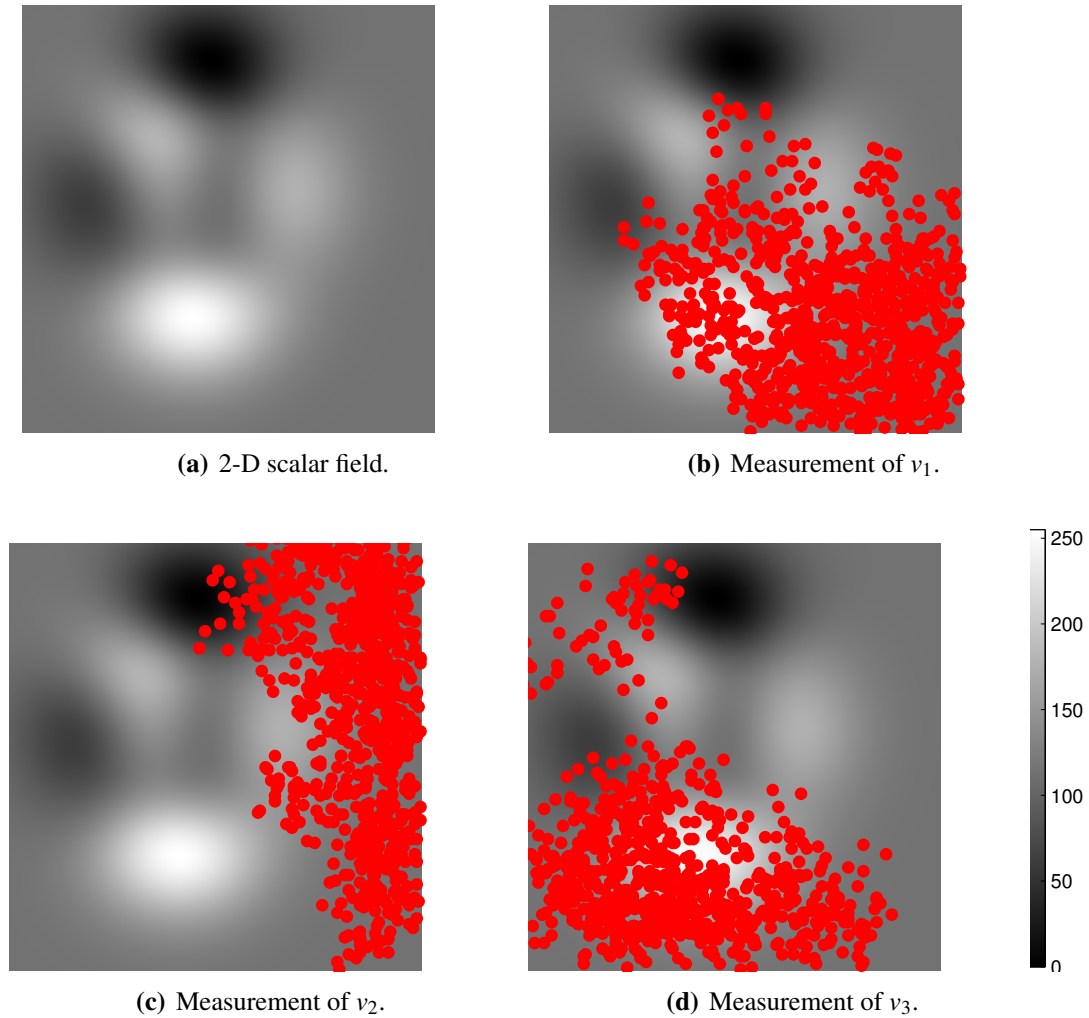
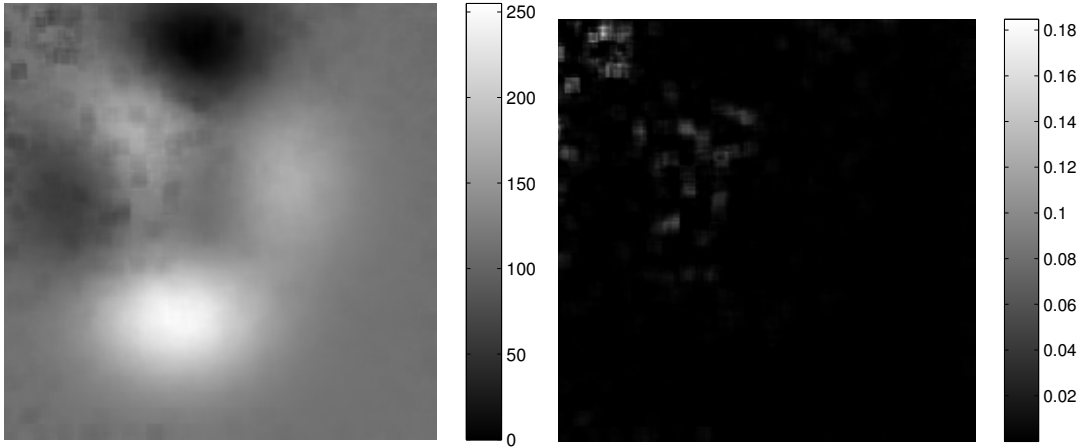


Figure 4.9: Scalar field and measurement distribution. Red dots indicate measurement locations.

In this chapter, an adaptive sparse sampling approach has been developed. A mobile sensor network is used to collect measurements and reconstruct an unknown sensing field. Each

measurement collected is particularly designed. The efficiency of this method falls into the fact that each measurement is of great information amount to represent the target sensing field. A 2-D scalar field in Figure 4.9(a) is used as the target sensing field to be reconstructed. A small portion of the most informative measurements are collected by a mobile sensor network to justify the method presented in this paper. Each measurement is evaluated in sparse domain, and collected in spatial domain with the measuring noise defined by n_0 .

We assume to use networked unmanned aerial vehicles (UAVs) to reconstruct a 2-D temperature field on a $10km \times 10km$ area. Each mobile sensor is equipped with imaging sensor, which can capture invariant features and localize other sensors. Figure 4.9(a) shows the sensing field with pixels being scaled between 0 and 255. Parameters are initialized. The scalar field in Figure 4.9(a) is of resolution 256×256 . 2% initial measurements are randomly taken, which is $256^2 \times 2\% = 1311$. $A_{RB(i)}^{(t)}$ is 32×32 , with sensor v_i in the center at time t . In Section 4.3.2, α_0 and α_k are initialized as 1, and N_{iter} is 5. The moving cost, in Section 4.3.3 $C(\phi_i^t, \hat{\phi}_i^{t+1})$ is proportional to the distance, and small weighted factor ω is used, which is set 10^{-6} . The safety margin d is $10m$.



(a) Sensing field reconstruction with error 0.1029.

(b) Errors for each pixel.

Figure 4.10: Reconstruction performance with 8000 measurements.

In this simulation, a mobile sensor network of 10 sensors is deployed in the sensing field to collect the most informative measurements and reconstruct the sensing field. They are randomly deployed initially. Within the mobile sensor network, different sensors may cover different portions after executing the method addressed in this paper. Figure 4.9(b) shows all the measurements collected by sensor v_i , which covers only a portion of the sensing field. Figure 4.9(c) and 4.9(d) show the measurement distributions of another

two sensors. Though none of the sensors in the mobile sensor network covers the entire sensing field, they can collaborate to provide full coverage with certain overlaps. Each sensor can real-time reconstruct its coverage of the sensing field with high fidelity, since measurements are relatively concentrating. Signals reconstructed are merged together every 1000 measurements collected by the entire mobile sensor network in this simulation. Red curve in Figure 4.11 shows the reconstruction errors given different numbers of measurements collected by the entire network. The reconstruction error between the original signal X and reconstruction \hat{X} is defined as $\|\hat{X} - X\|_2 / \|X\|_2$. Figure 4.10(a) shows the reconstruction of Figure 4.9(a). The total measurement percentage, including initial portion and informative portion is $2\% + 12.21\% = 14.21\%$. The reconstruction error of each sensing field pixel is shown in Figure 4.10(b), where each pixel is calculated by $\|\hat{x}_k - x_k\|_2 / \|X\|_2 \times 256^2$, $k = 1 \dots 256^2$. Only a very small portion pixels are not perfectly reconstructed with the low percentage of measurements.

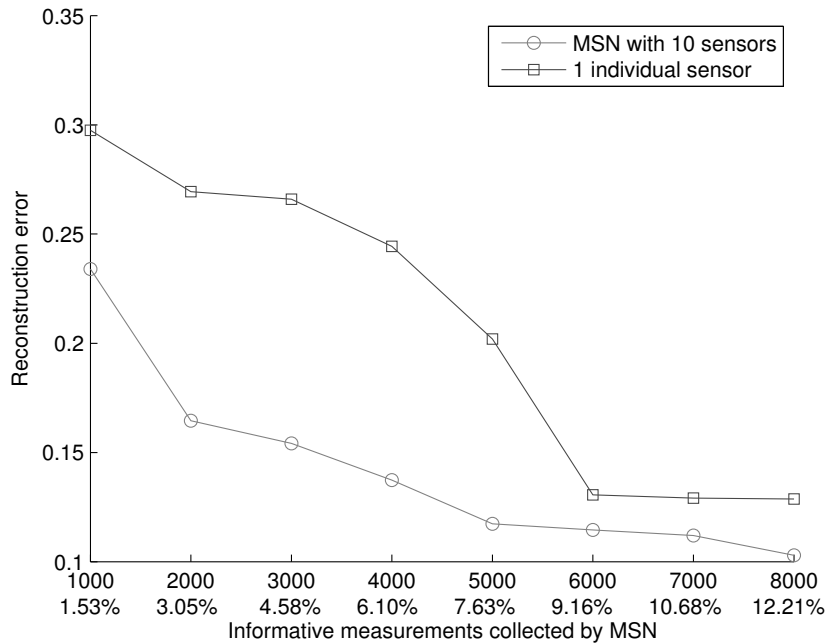


Figure 4.11: Reconstruction performance comparison.

If the number of sensors in the mobile sensor network reduces to 1, the measurement determination algorithm is still effective without network collaboration. We compare the performance between a mobile sensor network of 10 sensors and 1 individual sensor in reconstruction errors and path length. The reconstruction error curves are shown in Figure 4.11. These two curves share the same set of initial measurements. The reconstruction by the mobile sensor network converges much faster. The path length curves are shown in Figure 4.12, which is calculated at the same interval of reconstruction error. Sensor trajectories are recorded in this simulation, and path length can be therefore accumulated.

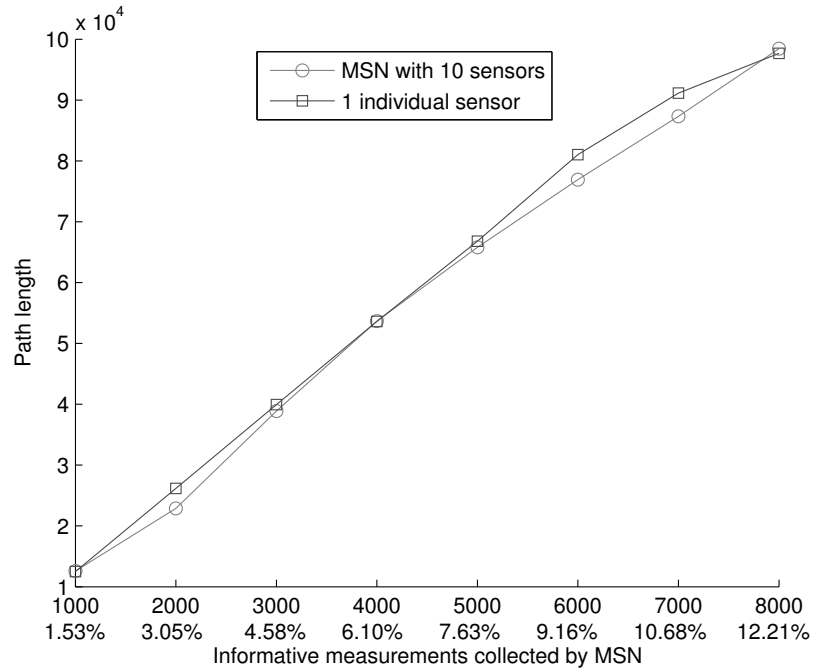


Figure 4.12: Path length comparison.

The total path length of entire mobile sensor network is shown versus the path length of the individual sensor. Though there is no much difference in the path length curves between an mobile sensor network and a individual sensor, the mobile sensor network outperforms a lot in the reconstruction errors. This is exactly the efficiency of an mobile sensor network, which can be summarized in two perspectives. First, the network can explore different areas simultaneously resulting fast drop in the reconstruction error curve for the first several thousand measurements. Second, shared information between mobile sensors gives more accuracy in measurement prediction, which increases the fidelity of the inference of the most informative measurements newly determined.

4.5 Conclusion

In this chapter, an adaptive sampling mobile sensor networks is proposed to reconstruct an unknown sensing field. Measurements are evaluated and optimized under a sparse domain by information amount. Network collaboration has been emphasized, making the measurement determination more robust. Networked sensors have been proven more efficient compared with an individual sensor.

Chapter 5

Development of Quadrotor UAV

The adaptive sampling algorithms have been developed for both single mobile sensors and mobile sensor networks to reconstruct 2-D scenarios. To expand the proposed algorithm to 3-D scenarios, this chapter will build a unmanned aerial vehicle(UAV) and develop control algorithms. The proposed UAV is four-rotor micro aerial robot. This quadrotor is a mechatronic system, whose four rotors provide forces for lift and control. Rhe quadrotor differs a lot from the traditional helicopter that has one main rotor and one tail rotor. The antitorque control mechanisms are different.

Traditional helicopters with a big rotor on top create torque when running, turning the helicopter opposite. To eliminate the opposite rotation a vertical tail rotor is used for torque compensation. However, the tail rotor compensation generates a sideways torque, which has to be compensated by tilting the main rotor blades. Thus, the control of traditional helicopters would be a complicated process. The quadrotor's lift and control is relatively simple. It has four propellers driven by four motors separately in a cross configuration. Two more control input variables than traditional helicopters makes life easier.

The attitude control of quadrotor should be considered as the prerequisite to accomplish other tasks, including target tracking, landing, etc. Many control systems have been proposed. Conventional PD or PID control has been implemented in [133, 134, 135, 136, 137]. Specific model parameters are not required by a PID or PD structure, and the control law is simple to implement. Lyapunov Theory control has been discussed in [138, 139, 140, 141], according to which the asymptotical stability of the helicopter can be ensured. Also, adaptive techniques [142, 143] provide good control performance with uncertain parameters and unmodeled dynamics. Linear Quadratic Regulator based control are presented in [136, 144]. Backstepping control is done in [145, 146, 147], which guarantees the convergence of quadrotor internal states, however, requires quite

a lot of computation. Dynamic feedback is implemented in some quadrotor projects [148, 149], where the closed loop system control part is transformed into a decoupled, controllable, and linear subsystem. Visual feedback control is also used; cameras or visual sensors can be either mounted on the helicopter [150, 151, 152] or fixed on the ground [153, 154]. Other types of quadrotor control include fuzzy techniques [155], neural networks [156], reinforcement learning [157], etc. As existing literature has provided many control mechanisms, this chapter is developing both software and hardware for quadrotor control.

5.1 Quadrotor Kinematics and Dynamics

A quadrotor is a powerful mobile sensing platform. The moving flexibility enables the quadrotor to fast respond to the environment for particular sensing demands. It carries various types of sensors, collecting both environmental data and quadrotor flight data. Figure 5.1 shows the quadrotor we developed with sensors built on. As shown in Figure 5.1 lasers, ultrasonics, IMU and reflective sensors collect different typys of information. On board processors fuse sensed data and translate sensing demands to quadrotor motion commands. To ensure an accurate and fast respond, quadrotor kinematics is addressed in this section, starting with the analysis of quadrotor moving model.

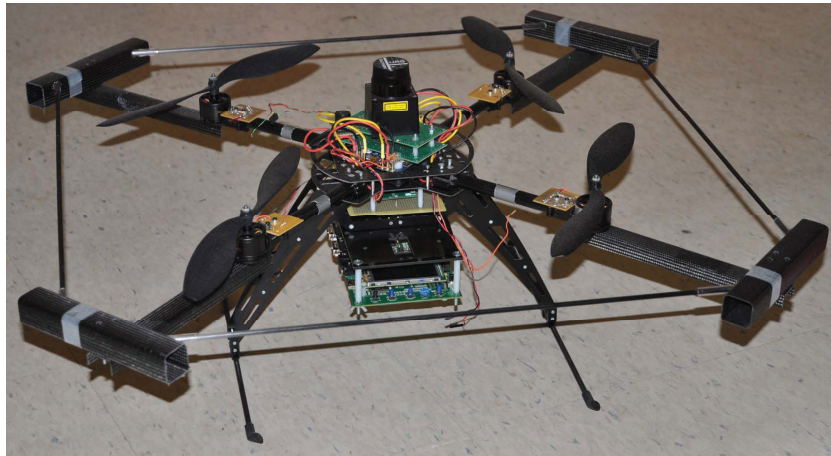


Figure 5.1: Mobile sensing platform: quadrotor with various sensors built on: lases range finder, ultrasonic, IMU, on board processor, and reflective sensors (rotor speed reader).

5.1.1 Quadrotor Kinematics

To describe quadrotor kinematics, it has to be defined in a coordinate system. A popular method to define the quadrotor is Cartesian coordinate, where the x and y axis are along the quadrotor levers, and the z axis is perpendicular to the x and y . Figure 5.2 illustrates this situation with the moving direction along the x axis. In our design of Figure 5.1, cameras would be installed between two levers later on. To ensure cameras facing the moving direction, an X model is used as shown in Figure 5.3. z axis remains the same, while x and y axis are rotated by 45 degrees. The position of quadrotor is defined by the vector $r^T = \{x, y, z\}$ in the given earth frame as shown in Figure 5.3. By comparing the earth frame and the quadrotor body-fixed frame, three Euler angles, Roll, pitch, and yaw, can be defined as the rotation angles along the axis of the three axes separately, forming the angle vector $\Omega^T = \{\phi, \theta, \psi\}$. These angles describes the attitude of the quadrotor and can be directly controlled by rotor speed.

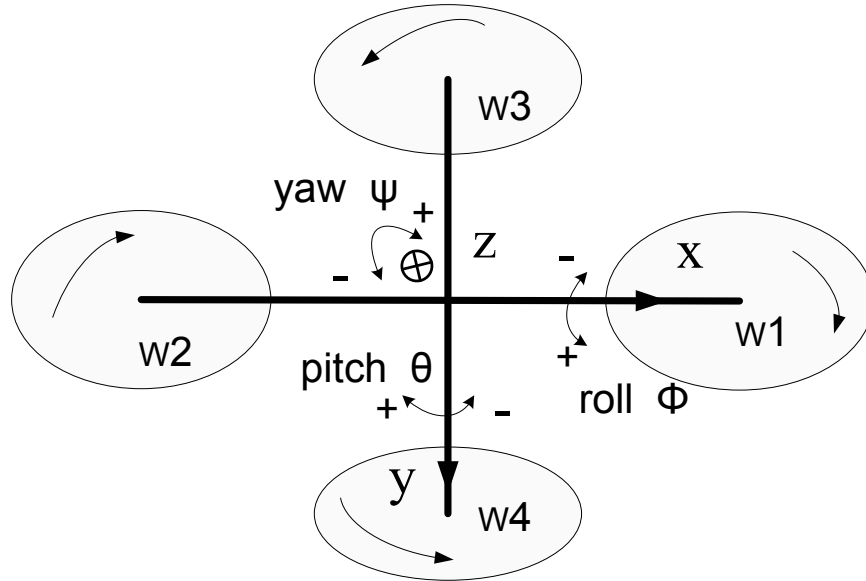


Figure 5.2: Quadrotor + model with roll, pitch, and yaw angles illustrated.

The kinematics of a quadrotor includes six degrees of freedom, three angles (ϕ, θ, ψ) and three coordinates (x, y, z). Thus, the control of the quadrotor falls into two categories: attitude control which is to control the three angles, and position control which is to control the position coordinates. As shown in Figure 5.3, the roll movement is provided by speeding rotor 1 and 3 up or down, and at the same time speeding rotor 2 and 4 down or up. It leads to a rotation with the respect to x axis. The pitch movement is achieved by increasing or decreasing the speed of rotor 1 and 4, and at the same time decreasing or

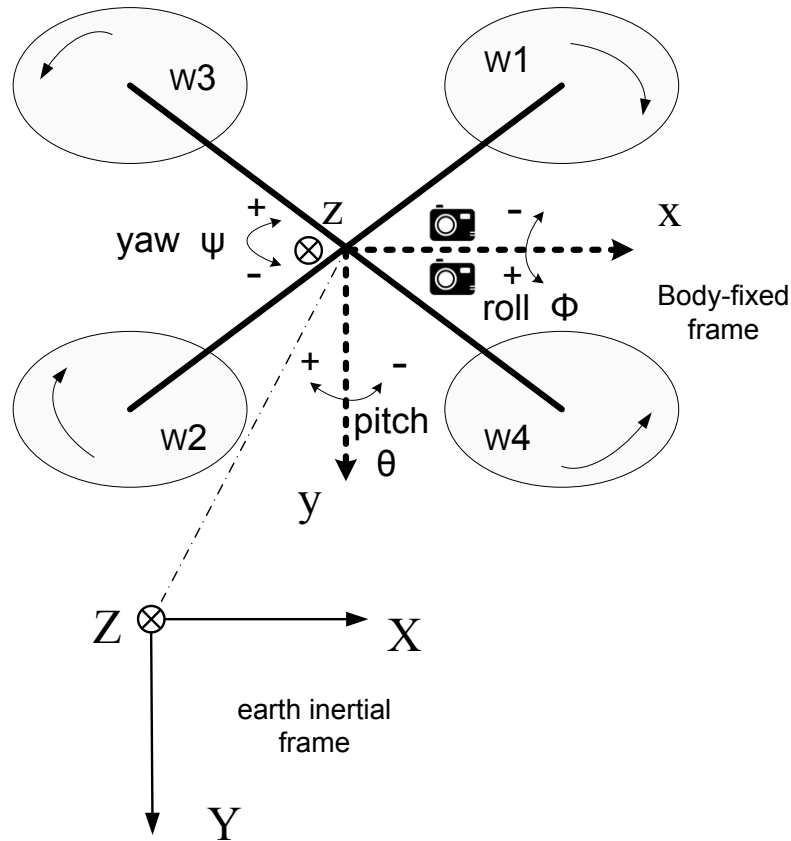


Figure 5.3: Quadrotor X models with roll, pitch, and yaw angles illustrated.

increasing the speed of rotor 2 and 3. It leads to a rotation with the respect to y axis. The yaw movement is provided by increasing or decreasing the speed of rotor 1 and 2, and at the same time decreasing or increasing the speed of rotor 3 and 4. It leads to a rotation with the respect to z axis. The z movement is provided by controlling the speed of all the four rotors at the same time, which leads to a vertical force. x and y movements are based on the tilt angles of pitch and roll separately. With a certain value of the tilt angle, the vertical force pushes the quadrotor horizontally.

The control inputs that can directly affect the quadrotor angle and position are combinations of the four rotors. A rotor can provide forces proportional to the square of its rotational speed for both vertical thrust and horizontally rotation torque. Therefore, four artificial

input variables are defined as follows:

$$\begin{aligned}
U_1 &= b_1 \omega_1^2 + b_2 \omega_2^2 + b_3 \omega_3^2 + b_4 \omega_4^2 \\
U_2 &= b_1 \omega_1^2 - b_2 \omega_2^2 + b_3 \omega_3^2 - b_4 \omega_4^2 \\
U_3 &= b_1 \omega_1^2 - b_2 \omega_2^2 - b_3 \omega_3^2 + b_4 \omega_4^2 \\
U_4 &= d_1 \omega_1^2 + d_2 \omega_2^2 - d_3 \omega_3^2 - d_4 \omega_4^2
\end{aligned} \tag{5.1}$$

where b_i and d_i , $i = 1, 2, 3, 4$, are the proportional factor of motor i for force and torque separately. U_1 denotes the thrust force applied to the quadrotor vertically. It directly generates the movement along x , y , and z directions with corresponding tilt angles. U_2 and U_3 denote the forces which lead to the roll torque and pitch torque separately. U_4 denotes the yaw torque. These four inputs are linear combinations of rotation speed square of rotors. The control of the quadrotor is to design the control inputs U_i , $i = 1, 2, 3, 4$, and the rotor speed, which is equivalent to the speed control of rotors due to the linear relationship, and the speed of the four rotors can be controlled by the onboard processor.

5.1.2 Quadrotor Dynamics

This part presents how a quadrotor responds to the system inputs defined previously. Newton-Euler equations are used to illustrate a quadrotor dynamics. Newton's equation describes the impact of a force, while Euler's equation illustrates a torque impact to quadrotor body. The dynamics of the quadrotor can be summarized as two sets differential equations, corresponding to force and torque. The force equations apply to the position vector $r^T = \{x, y, z\}$, and the torque equations apply to the angle $\Omega^T = \{\phi, \theta, \psi\}$. Given an Euler angle vector $\Omega^T = \{\phi, \theta, \psi\}$ between two frames, the transformation of from the quadrotor body frame to the earth frame is given

$$R = \begin{pmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \tag{5.2}$$

where c and s denote cos and sin functions separately.

On a quadrotor, forces applied include the gravity and thrust force U_1 without considering the body rotation. Thus, according to Newton's law the first set of differential equations

about three direction acceleration yield

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix} = g \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - R \cdot \frac{U_1}{m} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (5.3)$$

\ddot{x} , \ddot{y} , and \ddot{z} are the second order derivatives of the position of the quadrotor in the earth inertial frame. g is the gravity, and $g \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ expresses the gravity in three directions. m

is the quadrotor mass, and $R \cdot \frac{U_1}{m} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ indicates the U_1 impact in the earth inertial frame.

Expanding Equation (5.3) can result three differential equations

$$\ddot{x} = -(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \frac{U_1}{m} \quad (5.4)$$

$$\ddot{y} = -(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \frac{U_1}{m} \quad (5.5)$$

$$\ddot{z} = g - (\cos \phi \cos \theta) \frac{U_1}{m} \quad (5.6)$$

Euler's equations, describing rigid body rotation, involves the applied torques, moments of inertia, and angular velocity. The second set of differential equations can be obtained accordingly

$$\tau = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \cdot \begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{pmatrix} + \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \times \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \cdot \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \quad (5.7)$$

where τ is the total applied torques, and I_x , I_y and I_z are principal moments of inertia. The total torque τ can be written as

$$\tau = \begin{pmatrix} LU_2/\sqrt{2} \\ LU_3/\sqrt{2} \\ U_4 \end{pmatrix} + I_R \left(\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right) g(U) \quad (5.8)$$

where I_R is the rotor inertia, L indicates the lever length of quadrotor, and $g(U) = \omega_1 + \omega_2 - \omega_3 - \omega_4$. The first item in Equation (5.8) describes the total torque applied to quadrotor body, and the second item is gyroscopic torques caused by rotor rotation. Summarizing Equation (5.7) and (5.8), three differential equations about quadrotor angles

can be acquired.

$$\ddot{\phi} = \dot{\theta} \dot{\psi} \left(\frac{I_y - I_z}{I_x} \right) - \frac{I_R}{I_x} \dot{\theta} g(U) + \frac{L}{\sqrt{2} I_x} U_2 \quad (5.9)$$

$$\ddot{\theta} = \dot{\phi} \dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) + \frac{I_R}{I_y} \dot{\phi} g(U) + \frac{L}{\sqrt{2} I_y} U_3 \quad (5.10)$$

$$\ddot{\psi} = \dot{\theta} \dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) + \frac{1}{I_z} U_4 \quad (5.11)$$

The entire dynamical model described in Equation (5.4), (5.5), (5.6), (5.9), (5.10), and (5.11) can be written in the form of $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$, where \mathbf{x} is the vector of state variables.

$$\mathbf{x}^T = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}) \quad (5.12)$$

Then the equations above can be rewritten as

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{x}_4 \\ \dot{\mathbf{x}}_2 &= \mathbf{x}_5 \\ \dot{\mathbf{x}}_3 &= \mathbf{x}_6 \\ \dot{\mathbf{x}}_4 &= -(\cos \mathbf{x}_7 \sin \mathbf{x}_8 \cos \mathbf{x}_9 + \sin \mathbf{x}_7 \sin \mathbf{x}_9) \frac{U_1}{m} \\ \dot{\mathbf{x}}_5 &= -(\cos \mathbf{x}_7 \sin \mathbf{x}_8 \sin \mathbf{x}_9 - \sin \mathbf{x}_7 \cos \mathbf{x}_9) \frac{U_1}{m} \\ \dot{\mathbf{x}}_6 &= g - (\cos \mathbf{x}_7 \cos \mathbf{x}_8) \frac{U_1}{m} \\ \dot{\mathbf{x}}_7 &= \mathbf{x}_{10} \\ \dot{\mathbf{x}}_8 &= \mathbf{x}_{11} \\ \dot{\mathbf{x}}_9 &= \mathbf{x}_{12} \\ \dot{\mathbf{x}}_{10} &= \mathbf{x}_{11} \mathbf{x}_{12} \left(\frac{I_y - I_z}{I_x} \right) - \frac{I_R}{I_x} \mathbf{x}_{11} g(U) + \frac{L}{\sqrt{2} I_x} U_2 \\ \dot{\mathbf{x}}_{11} &= \mathbf{x}_{10} \mathbf{x}_{12} \left(\frac{I_z - I_x}{I_y} \right) + \frac{I_R}{I_y} \mathbf{x}_{10} g(U) + \frac{L}{\sqrt{2} I_y} U_3 \\ \dot{\mathbf{x}}_{12} &= \mathbf{x}_{10} \mathbf{x}_{12} \left(\frac{I_z - I_x}{I_y} \right) + \frac{1}{I_z} U_4 \end{aligned} \quad (5.13)$$

5.2 Quadrotor Control

As an advanced mobile sensor, the quadrotor should fast respond to the environment for particular sensing tasks like target tracking. This requires both sensor measurement processing and real-time flight control. By collecting and processing sensor data from IMU, laser, camera, control tasks can be acquired. An elegant control mechanism adjusts the quadrotor to achieve the control tasks.

The quadrotor system involves six degrees of freedom, including position and angle vectors $r^T = \{x, y, z\}$ and $\Omega^T = \{\phi, \theta, \psi\}$. The control task has two aspects, stabilizing control and moving control. The former one is to control variables z , ϕ , θ , and ψ , which is supposed that the quadrotor is fixed at certain location x and y . The later one is to control variables x , y , z , and ψ , which can be used for target and setpoint tracking. This chapter is mainly focusing on the latter one.

As previously defined, four artificial control inputs are U_1, U_2, U_3 , and U_4 . In the stabilizing control, the four control inputs are used to control variables z , ϕ , θ , and ψ . ϕ and θ should always be kept as zero, and z and ψ should be maintained at certain values. In the stabilizing, the position x and y are regarded invariant. In moving control, z and ψ control remain the same, while position x and y are controlled instead of ϕ and θ . To be kept at certain position x and y , angles ϕ and θ have to be adjusted all the time. In the quadrotor model, it can be assumed that the tilt angle ϕ and θ are small, and changing smoothly. Therefore, Equation (5.4, 5.5, 5.6, 5.9, 5.10, 5.11) can be simplified as

$$\ddot{x} = -(\theta \cos \psi + \phi \sin \psi) \frac{U_1}{m} \quad (5.14)$$

$$\ddot{y} = -(\theta \sin \psi - \phi \cos \psi) \frac{U_1}{m} \quad (5.15)$$

$$\ddot{z} = g - (\cos \phi \cos \theta) \frac{U_1}{m} \quad (5.16)$$

$$\ddot{\phi} = \frac{L}{\sqrt{2}I_x} U_2 \quad (5.17)$$

$$\ddot{\theta} = \frac{L}{\sqrt{2}I_y} U_3 \quad (5.18)$$

$$\ddot{\psi} = \frac{1}{I_z} U_4 \quad (5.19)$$

5.2.1 Stabilizing Control

The stabilizing control involves Equation (5.16), (5.17), (5.18), and (5.19). Each control input will control an individual variable. In Equation (5.16), it is supposed that the only control variable is U_1 , and ϕ and θ are constant. Let $\tilde{U}_1 = g - (\cos \phi \cos \theta)U_1/m$, and we have the z control equation expressed in the form of the artificial input \tilde{U}_1 .

$$\ddot{z} = \tilde{U}_1 \quad (5.20)$$

where the angle ϕ and θ are considered constant while controlling z . Thus, the four variables required in the stabilizing control have the same form,

$$\ddot{x} = aU \quad (5.21)$$

where a is a constant, U is the control input, and x is the control variable. For Equation (5.20), (5.18), (5.17), and (5.19), a has different expressions separately, which can be calculated using quadrotor parameters directly.

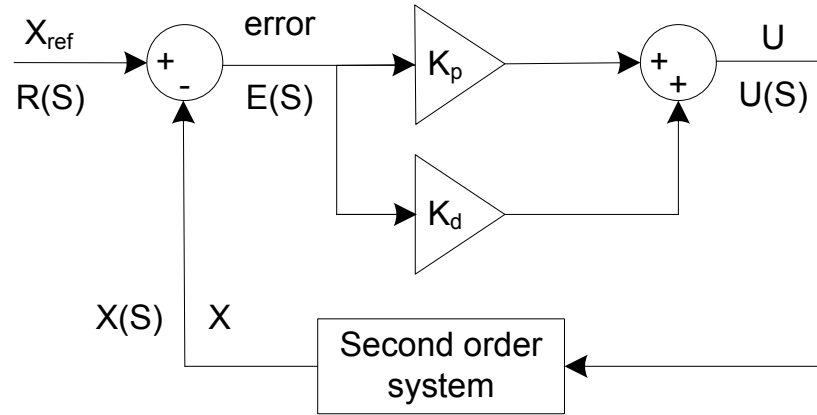


Figure 5.4: PD controller with X_{ref} as the control target.

The second order equation in Equation (5.21) can be controlled by a PD controller. Figure 5.4 shows the diagram of such a PD controller, where the equations in Laplace domain include

$$\frac{U(S)}{E(S)} = K_p + SK_d \quad (5.22)$$

$$\frac{X(S)}{U(S)} = \frac{a}{S^2} \quad (5.23)$$

$$E(S) = R(S) - X(S) \quad (5.24)$$

The close loop system transfer function can be summarized as

$$\frac{X(S)}{R(S)} = \frac{S^2}{S^2 + aK_d S + aK_p} \quad (5.25)$$

where the parameters K_d and K_p can be adjusted in order to make all the poles at the left side of the plane to ensure a stable system. Corresponding to Equation (5.20), (5.17), (5.18), and (5.19), closed loop functions yield

$$\frac{X_Z(S)}{R_Z(S)} = \frac{S^2}{S^2 + K_{d,z}S + K_{p,z}} \quad (5.26)$$

$$\frac{X_\phi(S)}{R_\phi(S)} = \frac{S^2}{S^2 + K_{d,\phi}S + K_{p,\phi}} \quad (5.27)$$

$$\frac{X_\theta(S)}{R_\theta(S)} = \frac{S^2}{S^2 + K_{d,\theta}S + K_{p,\theta}} \quad (5.28)$$

$$\frac{X_\psi(S)}{R_\psi(S)} = \frac{S^2}{S^2 + K_{d,\psi}S + K_{p,\psi}} \quad (5.29)$$

Thus, in the stabilizing control, eight parameters should be calculated and adjusted, including $K_{p,\phi}$, $K_{d,\phi}$, $K_{p,\theta}$, $K_{d,\theta}$, $K_{p,\psi}$, $K_{d,\psi}$, $K_{p,z}$, and $K_{d,z}$, where for example, $K_{d,\psi}$ corresponds to aK_d in Equation (5.25).

5.2.2 Moving Control

In the moving control, variables that need to be controlled include x , y , z , and ψ , where the control of z and ψ are the same as in the stabilizing control. In the control of x and y , θ and ϕ are immediate variables. By decoupling Equation (5.14) and (5.15), we have

$$\ddot{x}_{new} = -\frac{U_1}{m}\phi \quad (5.30)$$

$$\ddot{y}_{new} = -\frac{U_1}{m}\theta \quad (5.31)$$

where

$$\ddot{x}_{new} = \ddot{x} \cdot \sin \psi - \ddot{y} \cdot \cos \psi \quad (5.32)$$

$$\ddot{y}_{new} = \ddot{x} \cdot \cos \psi + \ddot{y} \cdot \sin \psi \quad (5.33)$$

By substituting variables, the control of x and y can be transferred to control x_{new} and y_{new} . Thus, Equation (5.17), (5.18), (5.30), and (5.31) construct the position control equations in four order

$$x_{new}^{(4)} = -\frac{U_1}{m} \frac{L}{\sqrt{2}I_x} U_2 \quad (5.34)$$

$$y_{new}^{(4)} = -\frac{U_1}{m} \frac{L}{\sqrt{2}I_y} U_3 \quad (5.35)$$

Both of the above equations are four order system, which has the form

$$x^{(4)} = aU \quad (5.36)$$

where $x^{(4)}$ indicates the fourth order derivative. A PID controller can handle a third order system at most. In this case, a fourth order control system is designed. The control diagram is shown in Figure 5.5. In the Laplace domain, we have

$$\frac{U(S)}{E(S)} = K_1 + K_2S + K_3S^2 + K_4S^3 \quad (5.37)$$

$$\frac{X(S)}{U(S)} = \frac{a}{S^4} \quad (5.38)$$

$$E(S) = R(S) - X(S) \quad (5.39)$$

The close loop system transfer function can be summarized as

$$\frac{X(S)}{R(S)} = \frac{S^4}{aK_1 + aK_2S + aK_3S^2 + aK_4S^3 + S^4} \quad (5.40)$$

To guarantee a stable system, poles have to be placed on the left plain in the Laplace domain. Eight parameters should be tuned to place left plain poles for the two 4th order system, including $K_{1,x}$, $K_{2,x}$, $K_{3,x}$, $K_{4,x}$, $K_{1,y}$, $K_{2,y}$, $K_{3,y}$, $K_{4,y}$. Corresponding to Equation

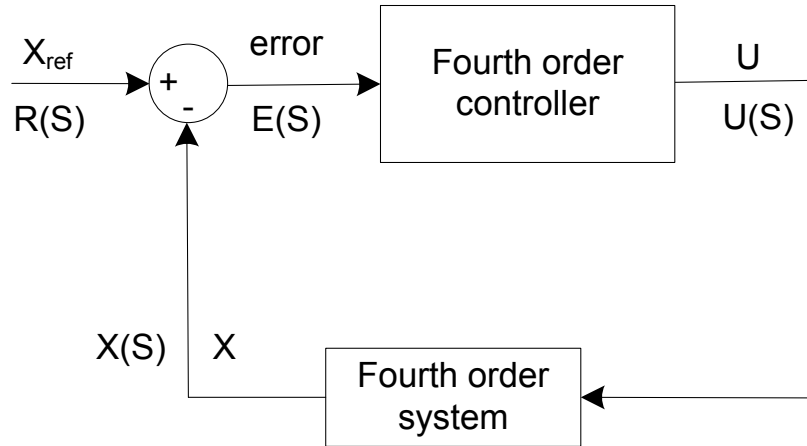


Figure 5.5: A 4th order controller with X_{ref} as the control target.

(5.34) and (5.35), closed loop functions can be found

$$\frac{X_x(S)}{R_x(S)} = \frac{S^4}{K_{1,x} + K_{2,x}S + K_{3,x}S^2 + K_{4,x}S^3 + S^4} \quad (5.41)$$

$$\frac{X_y(S)}{R_y(S)} = \frac{S^4}{K_{1,y} + K_{2,y}S + K_{3,y}S^2 + K_{4,y}S^3 + S^4} \quad (5.42)$$

5.2.3 Speed Control

As quadrotor moves, speed limit has to be set. Speed limit is discussed under the quadrotor frame, which corresponds to x_{new} and y_{new} . To setup speed limit is exactly to keep \dot{x}_{new} and \dot{y}_{new} at constant values. Let $V_x = \dot{x}_{new}$ and $V_y = \dot{y}_{new}$ in Equation (5.34) and (5.35), and we have the speed control equations in the third order

$$V_x^{(3)} = -\frac{U_1}{m} \frac{L}{\sqrt{2}I_x} U_2 \quad (5.43)$$

$$V_y^{(3)} = -\frac{U_1}{m} \frac{L}{\sqrt{2}I_y} U_3 \quad (5.44)$$

Both of the above equations are three order system, which has the form

$$x^{(3)} = aU \quad (5.45)$$

where $x^{(3)}$ indicates the third order derivative. The control diagram is shown in Figure 5.6. In the Laplace domain, we have

$$\frac{U(S)}{E(S)} = K_1 + K_2S + K_3S^2 \quad (5.46)$$

$$\frac{X(S)}{U(S)} = \frac{a}{S^3} \quad (5.47)$$

$$E(S) = R(S) - X(S) \quad (5.48)$$

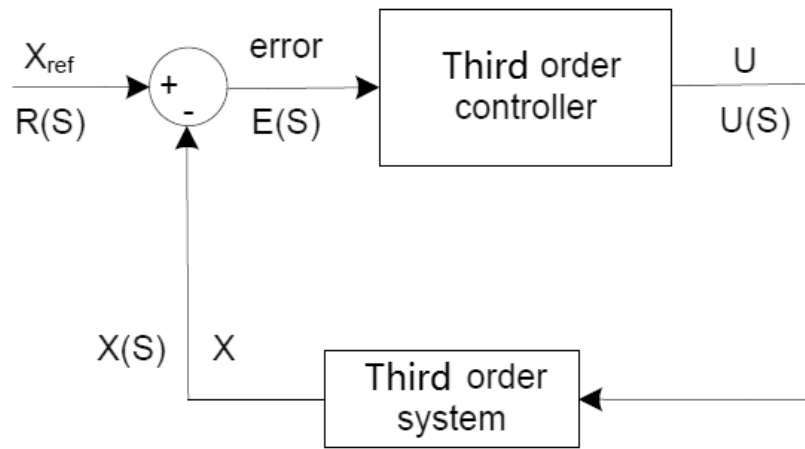


Figure 5.6: A 3rd order controller with X_{ref} as the control target.

The close loop system transfer function can be summarized as

$$\frac{X(S)}{R(S)} = \frac{S^3}{aK_1 + aK_2S + aK_3S^2 + S^3} \quad (5.49)$$

Six parameters should be tuned to place left plain poles for the two 3rd order system, including $K_{1,vx}$, $K_{2,vx}$, $K_{3,vx}$, $K_{1,vy}$, $K_{2,vy}$, $K_{3,vy}$. Corresponding to Equation (5.43 5.44), closed loop functions can be found

$$\frac{X_{v,x}(S)}{R_{v,x}(S)} = \frac{S^3}{K_{1,vx} + K_{2,vx}S + K_{3,vx}S^2 + S^3} \quad (5.50)$$

$$\frac{X_{v,y}(S)}{R_{v,y}(S)} = \frac{S^3}{K_{1,vy} + K_{2,vy}S + K_{3,vy}S^2 + S^3} \quad (5.51)$$

When the quadrotor flies, the two controller involving x and y alternates between moving control and speed control. The control input U would choose whichever is smaller to guarantee the speed limit.

Parameters related to stabilizing control, moving control, and speed control will be given in the simulations, which include $K_{p,\psi}$, $K_{d,\psi}$, $K_{p,z}$, $K_{d,z}$, $K_{1,x}$, $K_{2,x}$, $K_{3,x}$, $K_{4,x}$, $K_{1,y}$, $K_{2,y}$, $K_{3,y}$, $K_{4,y}$, $K_{1,vx}$, $K_{2,vx}$, $K_{3,vx}$, $K_{1,vy}$, $K_{2,vy}$, $K_{3,vy}$.

5.3 Hardware Setup

This section mainly focuses on hardware design of the quadrotor UAV. The overall hardware architecture will be introduced, and important quadrotor components will be explicated in details. For experimental use and field applications, a quadrotor could equip with multiple sensors, including cameras, lasers, ultrasonics, inferred, IMU, etc. Quadrotor control is the main topic in this chapter, only sensors related with UAV control are equipped. As discussed, six variables are of our concern, including the position vector $r^T = \{x, y, z\}$ and angle vector $\Omega^T = \{\phi, \theta, \psi\}$. x , y and ψ can be acquired from laser scanner by locating invariant features in the environment. z can be collected from ultrasonic sensor, and ϕ and θ can be exported by IMU module. Thus, only lasers, ultrasonics and IMU have been set up onto the UAV, which are adequate for quadrotor control.

5.3.1 Hardware Architecture

Figure 5.7 shows the hardware architecture for quadrotor control. An onboard processor plays the core role in the system, which is connected with different types of sensors. Once the moving command is given, the onboard processor will translate the moving command into PWM signals, and send to motor drivers. Then propellers rotate and move the quadrotor. As the UAV moves, sensor readings may change accordingly. Onboard processor receives new sensor readings, adjusts the moving demands and sends new PWM signals to driver peopellers. This process iterates until the moving demand is achieved.

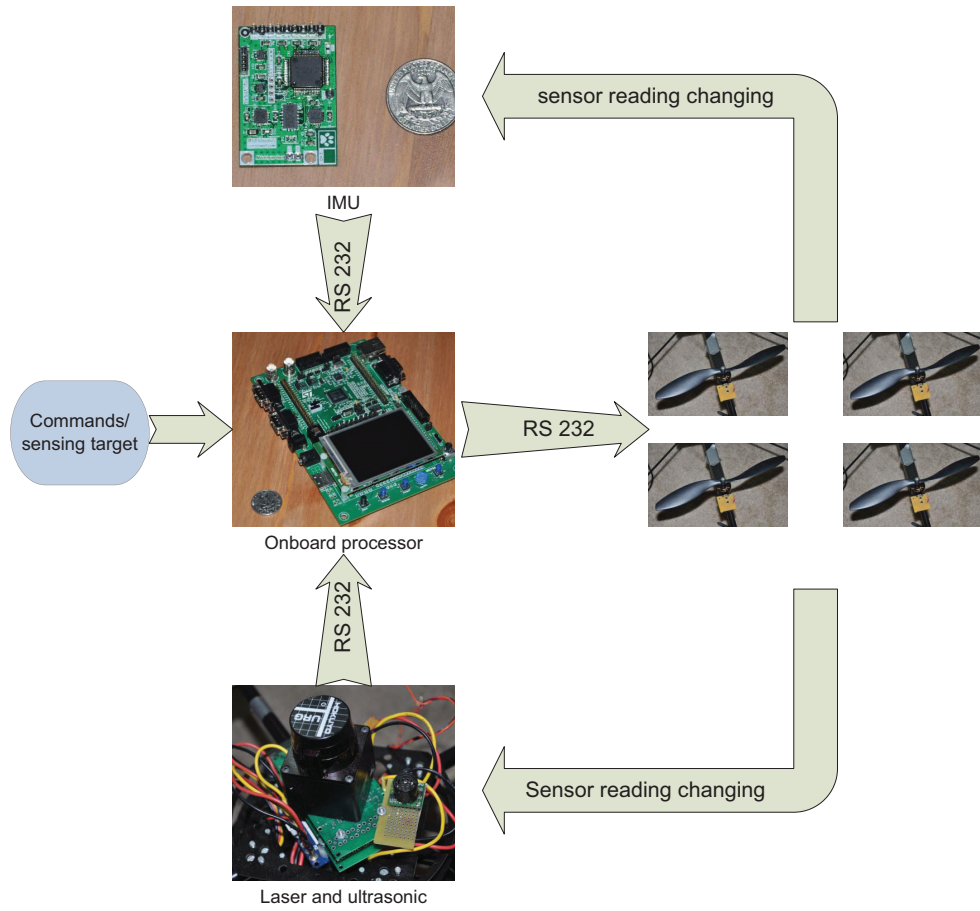
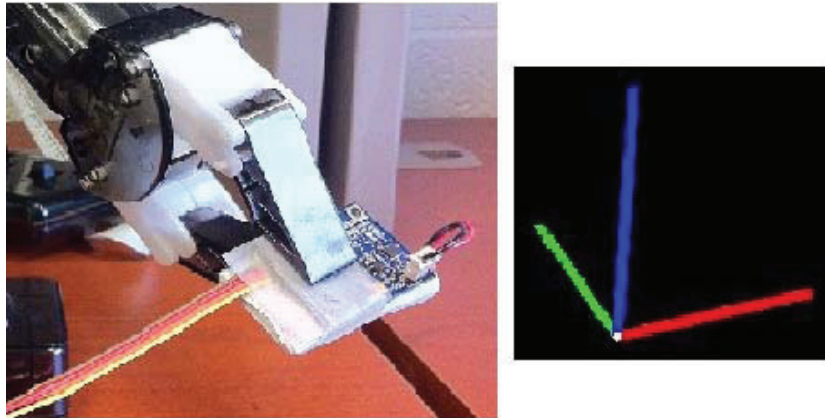


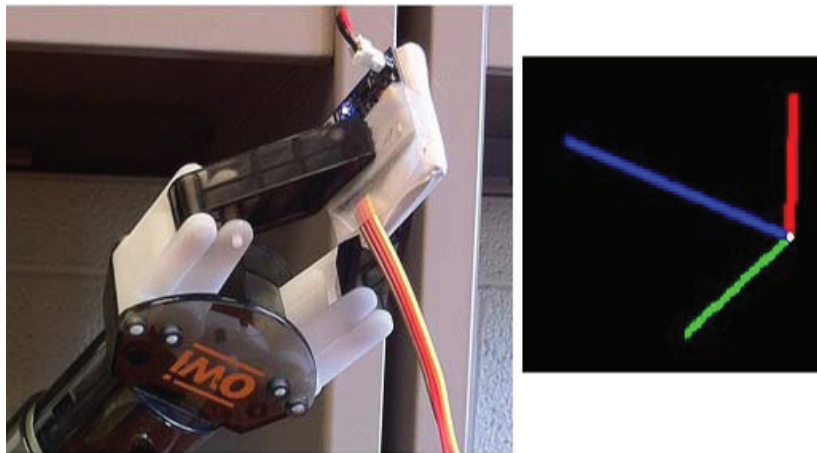
Figure 5.7: UAV architecture.



Figure 5.8: IMU module.



(a) IMU position 1.



(b) IMU position 2.

Figure 5.9: IMU module orientation displayed on PC screen.

5.3.2 IMU

The IMU sensor in Figure 5.8 used in this project is developed by Robotics Lab, Department of Electrical and Computer Engineering, Michigan Tech University. Sensors

soldered on the IMU module include accelerometer ¹, gyroscope ², and magnetometers ³. IMU sensors have accelerometers and gyroscopes built in, which tracks rotational movements. Gyroscopes cannot provide an accurate orientation value due to the accumulated gyroscope measurement errors. An accelerometer measures the earth's gravitational field, which can be used as an absolute reference. IMU with magnetometers could locate absolute direction aligned by the earth's magnetic field. Since the magnetometers can be easily interrupted in an indoor environment, the quadrotor developed in this chapter only uses accelerometer and gyroscope. Sensor fusion and filter algorithms have been discussed in various literatures [158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169]. Algorithm implemented into the IMU module is based on [170]. To test the IMU module as well as the implemented algorithm, the module is connected to a PC, where the IMU position could be displayed as three axis. Figure 5.9 shows two examples of different IMU positions to justify the implemented algorithm is working.

5.3.3 Laser Scanner and Ultrasonic Sensor

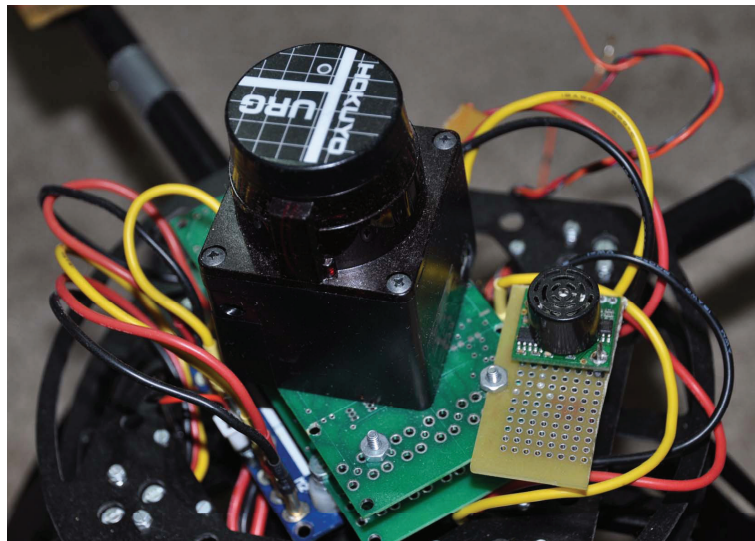


Figure 5.10: Laser and ultrasonic sensors.

Laser ⁴ and ultrasonic ⁵ are installed on the top of the quadrotor UAV to ensure their sensing is not blocked. The laser sensor can cover 240° semicircle with maximal detection distance

¹LIS3LV02DL, <http://www.st.com/internet/analog/product/127514.jsp>

²ITG-3200, <http://invensense.com/mems/gyro/itg3200.html>

³HMC5843, <http://www.sparkfun.com/products/9371>

⁴URG-04LX http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html

⁵MB1000 MaxSonar EZ0, <http://www.maxbotix.com/products/MB1000.htm>

4000mm and angular sensing resolution 0.36° . The accuracy of laser scanner is $\pm 1\%$. The ultrasonic sensor features 6.45m sensing range and 25.4mm resolution. It can detect the distance between UAV and ceiling roof. Both of the sensors are suitable for indoor environment, and they can send out sensed data through RS-232.

5.3.4 Motors and Propeller



Figure 5.11: UAV sitting on force and torque sensors.

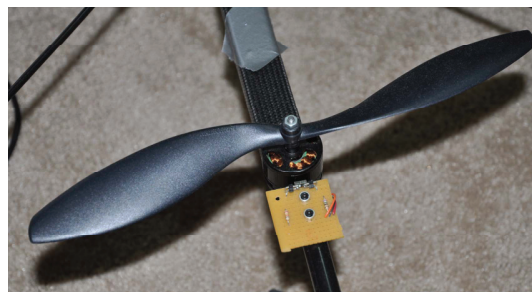
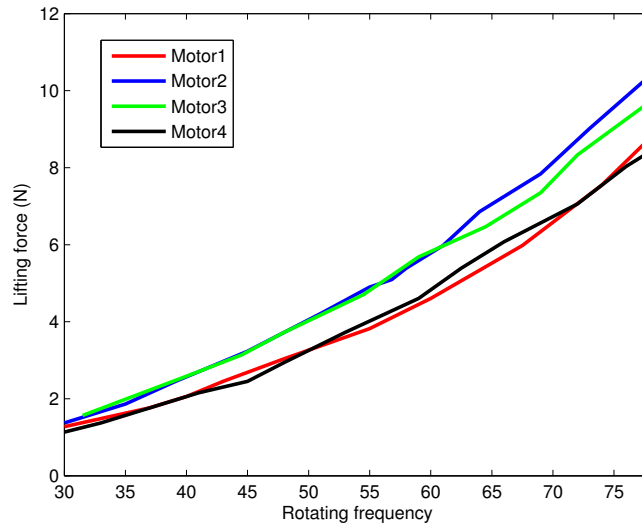
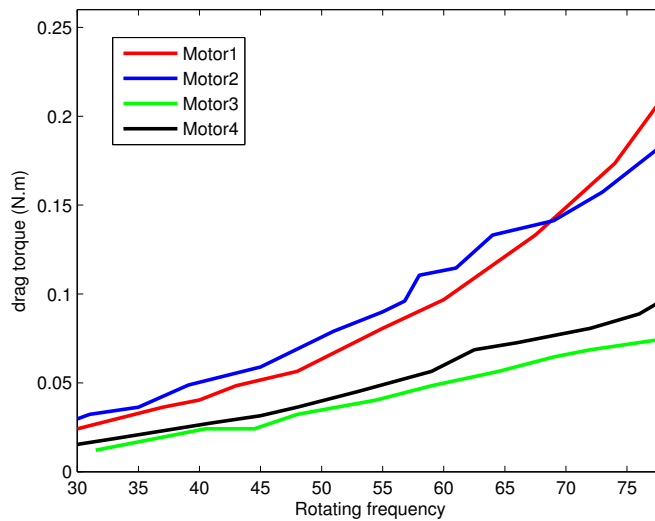


Figure 5.12: UAV motor and propeller with reflective sensor installed.

The quadrotor UAV uses 12V battery, and motor driver module converts DC to three phase AC to drive brushless motors. Each motor has a reflective sensor installed closely as shown in Figure 5.12, which monitors the rotor speed and generates correspond PWM signals.



(a) Motor lifting force measurement.



(b) Motor drag torque measurement.

Figure 5.13: Motor force and torque v.s. rotation speed.

With the reflective sensors, onboard processor can easily control motors' speed. As a motor rotates, it has two effects: lift force along the rotation axis, and drag torque centered on the mass point of the UAV. Experiments have been carried out to measure both the lift force Figure 5.13(a) and drag torque Figure 5.13(b) for different rotation speeds. Experiment setup is shown in Figure 5.11, where the quadrotor is sitting on torque and force sensors. These force and torque data is used as lookup tables when determining motor speed from system torque and force requirements. Based on the these figures, factors b_i and d_i for

$i = 1, 2, 3, 4$ can also be calculated.

5.3.5 Onboard Processor

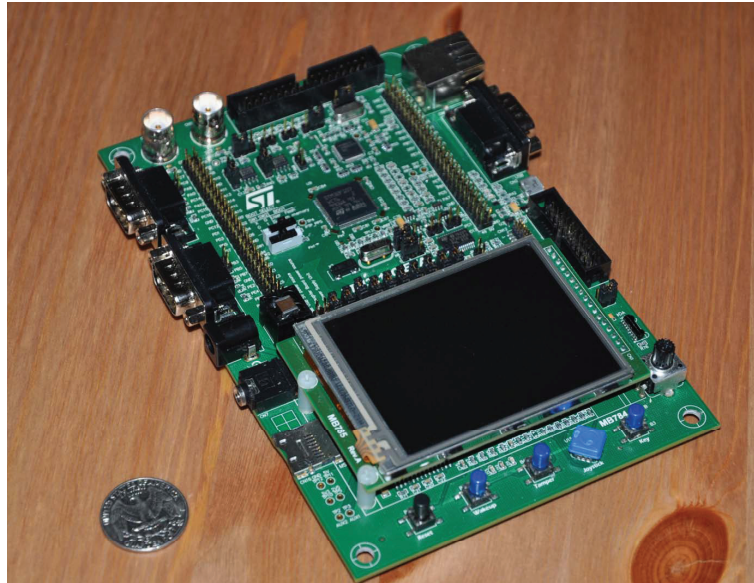


Figure 5.14: IMU module.

The onboard processing board ¹ is shown in Figure 5.14, which has a *STM32F107* chip built in. This embedded processor features *72MHz* CPU speed, *256K* flash and *64K* RAM. It is fast enough to process all the incoming data. Regarding the communication between onboard processor and sensors, IMU, laser and ultrasonic use RS232, while motors and reflective sensors use general I/O ports.

5.4 Simulations

In this section, quadrotor UAV is implemented in MATLAB, and simulation results will be shown. The quadrotor UAV developed in this chapter is used for mobile sensing, so it should be able to fly to a specific location and stay in the air. In this simulation, the quadrotor UAV flies to a specific position defined by the position vector $r^T = \{x, y, z\}$, and turn to a particular orientation defined by ψ . In such case, there are totally four parameters

¹STM3210C-EVAL <http://www.st.com/internet/evalboard/product/217965.jsp>

Table 5.1
Quadrotor UAV constants and parameters used in simulations

Parameter	Unit	Value	Note
g	m/s^2	9.8	Gravity constant
m	kg	1.5	Mass of the quadrotor UAV
L	m	0.25	UAV lever length
b_1	$kg \cdot m$	4.05×10^{-5}	Force factor of motor 1
b_2	$kg \cdot m$	4.28×10^{-5}	Force factor of motor 2
b_3	$kg \cdot m$	3.60×10^{-5}	Force factor of motor 3
b_4	$kg \cdot m$	3.60×10^{-5}	Force factor of motor 4
d_1	$kg \cdot m^2$	8.11×10^{-7}	Torque factor of motor 1
d_2	$kg \cdot m^2$	7.66×10^{-7}	Torque factor of motor 2
d_3	$kg \cdot m^2$	4.05×10^{-7}	Torque factor of motor 3
d_4	$kg \cdot m^2$	3.60×10^{-7}	Torque factor of motor 4
I_x	$kg \cdot m^2$	0.01	Inertia around x axis
I_y	$kg \cdot m^2$	0.01	Inertia around y axis
I_z	$kg \cdot m^2$	0.02	Inertia around z axis
$K_{p,\psi}$	N/A	5.2	Proportional parameter for yaw angle ψ
$K_{d,\psi}$	N/A	0.12	Derivative parameter for yaw angle ψ
$K_{p,z}$	N/A	40	Proportional parameter for z
$K_{d,z}$	N/A	16	Derivative parameter for z
$K_{1,x}$	N/A	0.9	1st parameter in the fourth order x controller
$K_{2,x}$	N/A	6	2nd parameter in the fourth order x controller
$K_{3,x}$	N/A	16	3rd parameter in the fourth order x controller
$K_{4,x}$	N/A	12	4th parameter in the fourth order x controller
$K_{1,y}$	N/A	0.9	1st parameter in the fourth order y controller
$K_{2,y}$	N/A	6	2nd parameter in the fourth order y controller
$K_{3,y}$	N/A	16	3rd parameter in the fourth order y controller
$K_{4,y}$	N/A	12	4th parameter in the fourth order y controller
$K_{1,vx}$	N/A	1	1st parameter in the x axis speed controller
$K_{2,vx}$	N/A	3	2nd parameter in the x axis speed controller
$K_{3,vx}$	N/A	3	3rd parameter in the x axis speed controller
$K_{1,vy}$	N/A	1	1st parameter in the y axis speed controller
$K_{2,vy}$	N/A	3	2nd parameter in the y axis speed controller
$K_{3,vy}$	N/A	3	3rd parameter in the y axis speed controller

to control. For the tilt angles along x and y axis, there is not too much concern, since those two angles would be stabilized around zero as long as the quadrotor UAV stays at a particular location. Before presenting the simulation results, constants and parameters used in the simulation are given in Table 5.1.

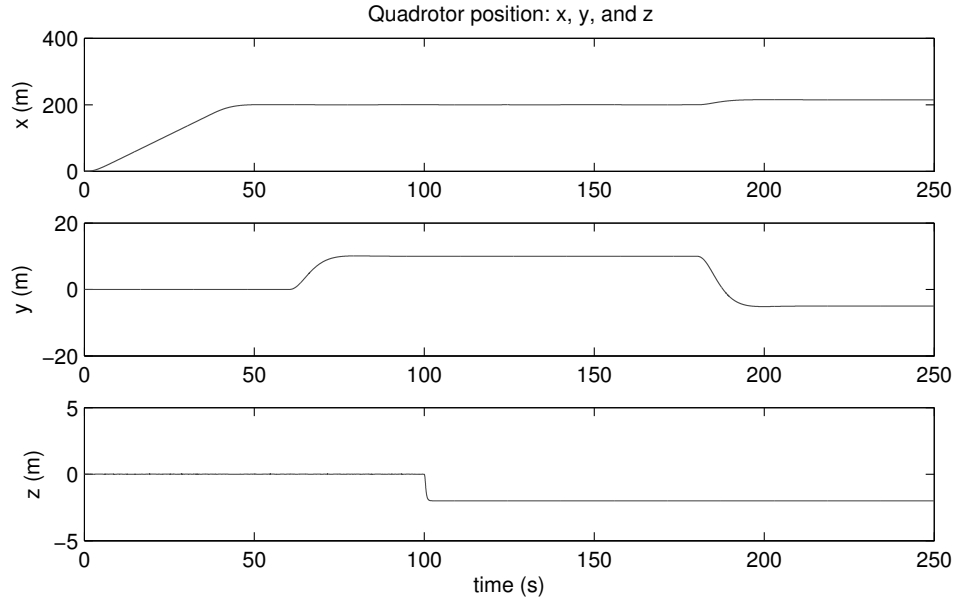


Figure 5.15: Set-point control: quadrotor position.

To fulfil sensing tasks, two simulations have been carried out, set-point control and trajectory tracking. In the set-point control simulation, the quadrotor would fly given a series of set points, which are defined by $\{x, y, z, \psi\}$. In the trajectory tracking simulation, a non-straight trajectory is given, which is supposed to be a target trajectory, and the quadrotor UAV would try to track the target.

5.4.1 Set-point Control

The starting point of the quadrotor is defined as $\{x, y, z, \psi\} = \{0, 0, 0, 0^\circ\}$, and it will follow a series of set-points, including $\{0, 0, 0, 0^\circ\}$, $\{200, 0, 0, 0^\circ\}$, $\{200, 10, 0, 0^\circ\}$, $\{200, 10, -2, 0^\circ\}$, $\{200, 10, -2, 54^\circ\}$, and $\{215, -5, -2, 54^\circ\}$. Quadrotor position is shown in Figure 5.15. The speed limit is set as $5m/s$, which can be observed from the constant slope in the x position and the speed information in Figure 5.16. The angle information is shown in Figure 5.17. Since the roll and pitch angles are not directly controlled, they look not as neat as yaw angle figure. The four artificial inputs are shown in Figure 5.18. These inputs change every time the quadrotor turns to a new set-point. Corresponding to the artificial inputs, motor speed is calculated as shown in Figure 5.19.

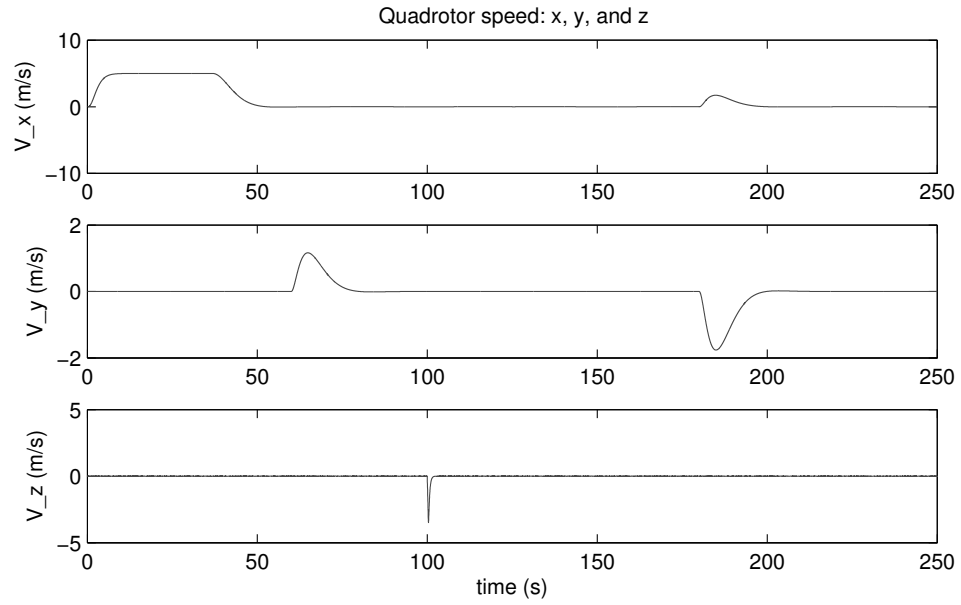


Figure 5.16: Set-point control: quadrotor speed.

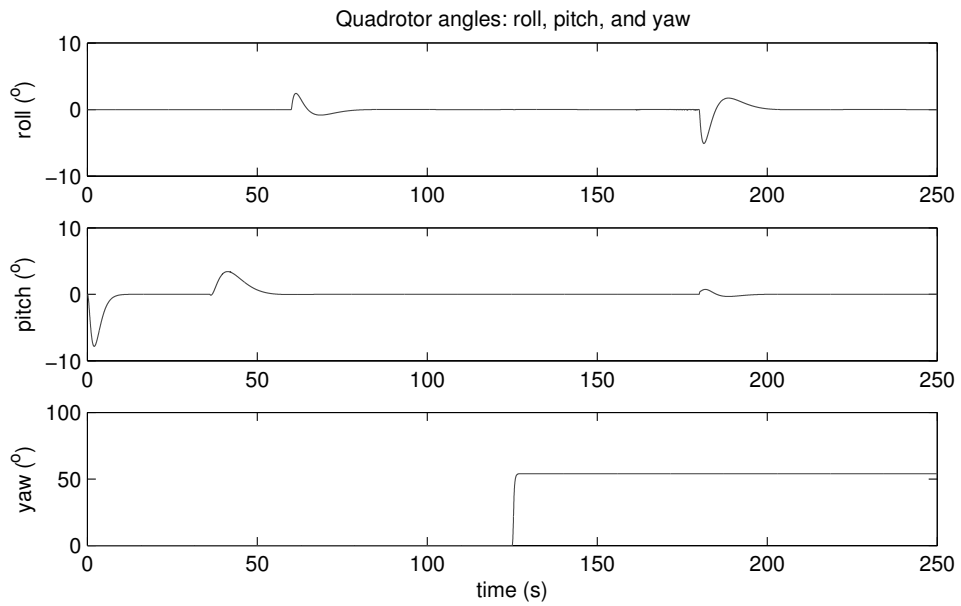


Figure 5.17: Set-point control: quadrotor angle.

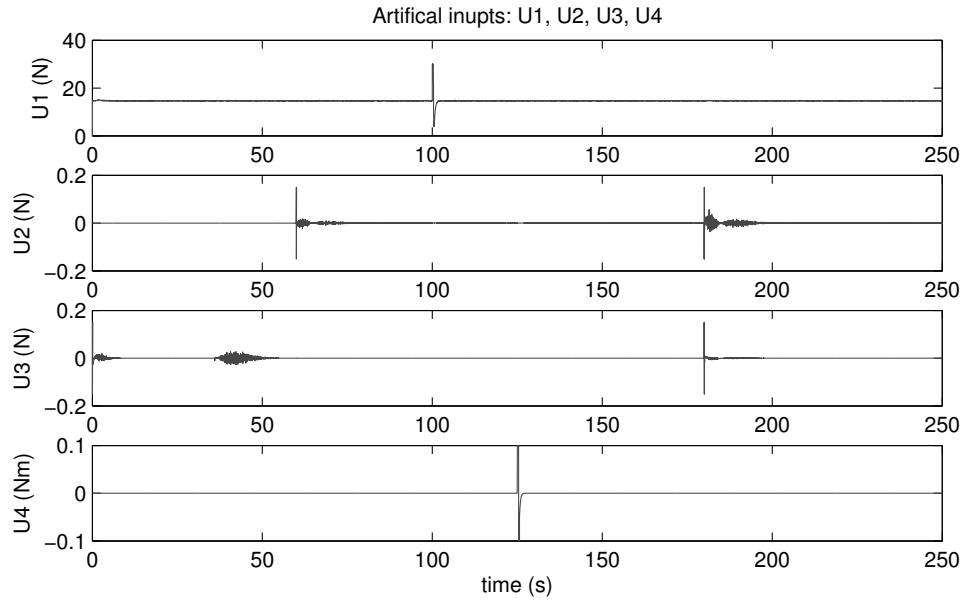


Figure 5.18: Set-point control: artificial inputs.

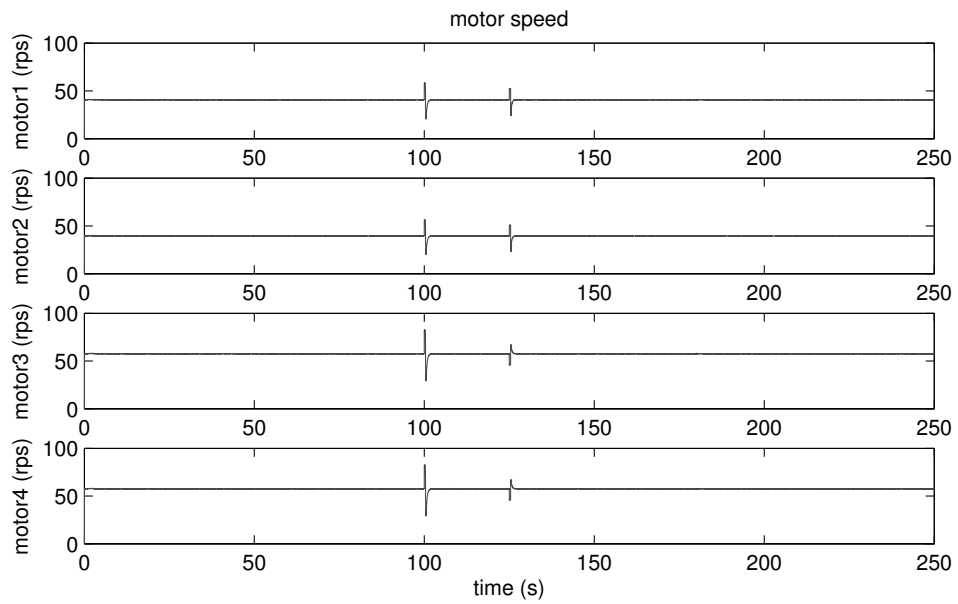


Figure 5.19: Set-point control: motor speed.

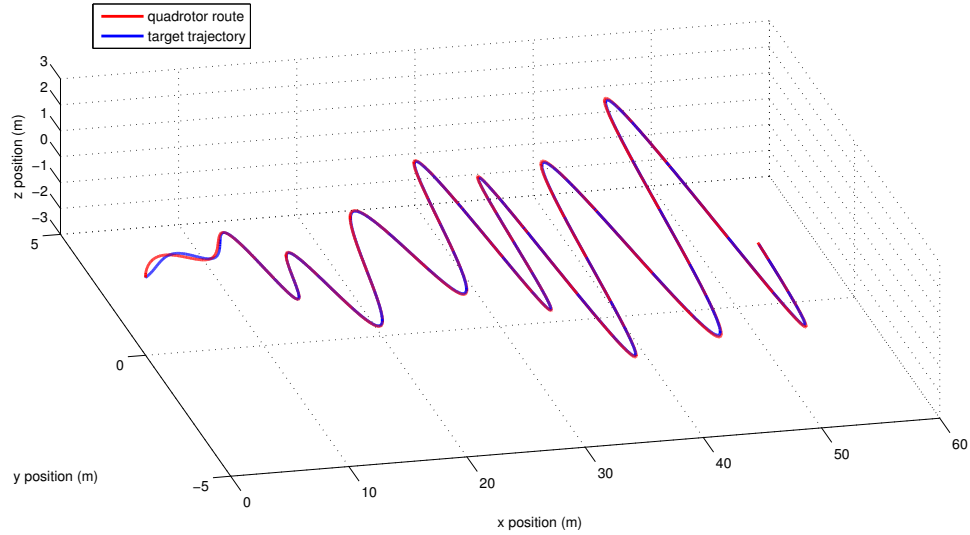


Figure 5.20: Tracking control: tracking trajectory.

5.4.2 Tracking Control

In this simulation, the quadrotor will track a pre-computed trajectory, which is assumed to be a potential target. The starting point is $(0, 0, 0)$, and the trajectory to track is defined as

$$\begin{aligned}
 x &= 0.5t \\
 y &= 0.02t \sin(0.5t) \\
 z &= \sin(0.15t)
 \end{aligned}
 \tag{5.52}$$

Figure 5.20 shows both the trajectory and the quadrotor moving route. For the few seconds at the very beginning, the tracking difference looks bigger than later, because it starts from zero speed and requires seconds to warm up. Figure 5.21, 5.22 and 5.23 show the position, speed, and angle separately.

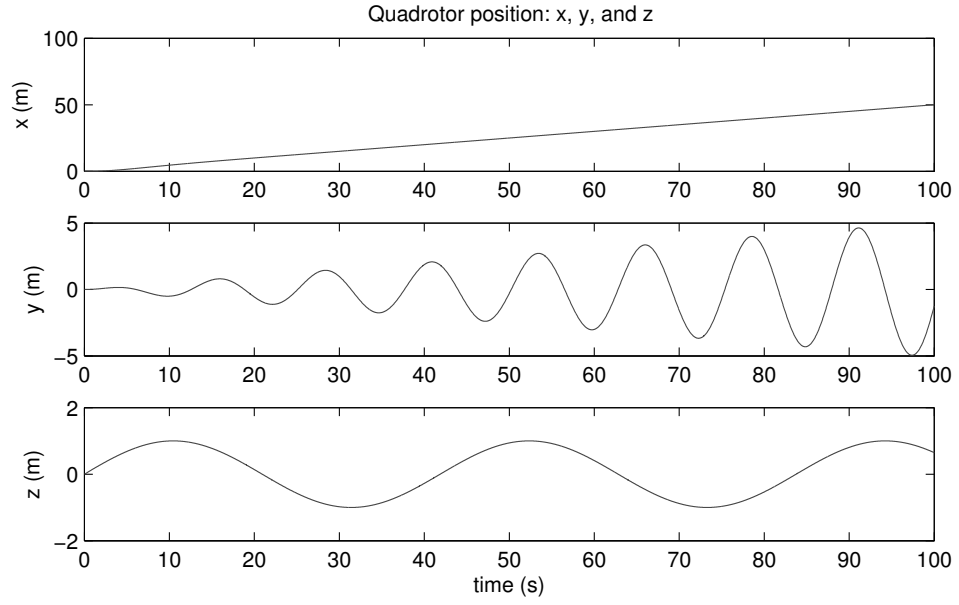


Figure 5.21: Tracking control: quadrotor position.

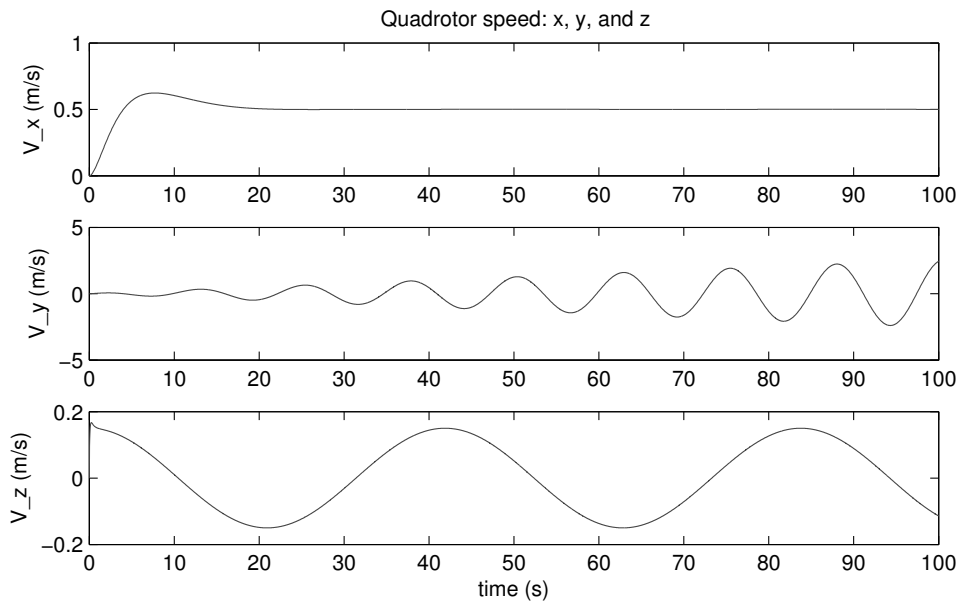


Figure 5.22: Tracking control: quadrotor speed.

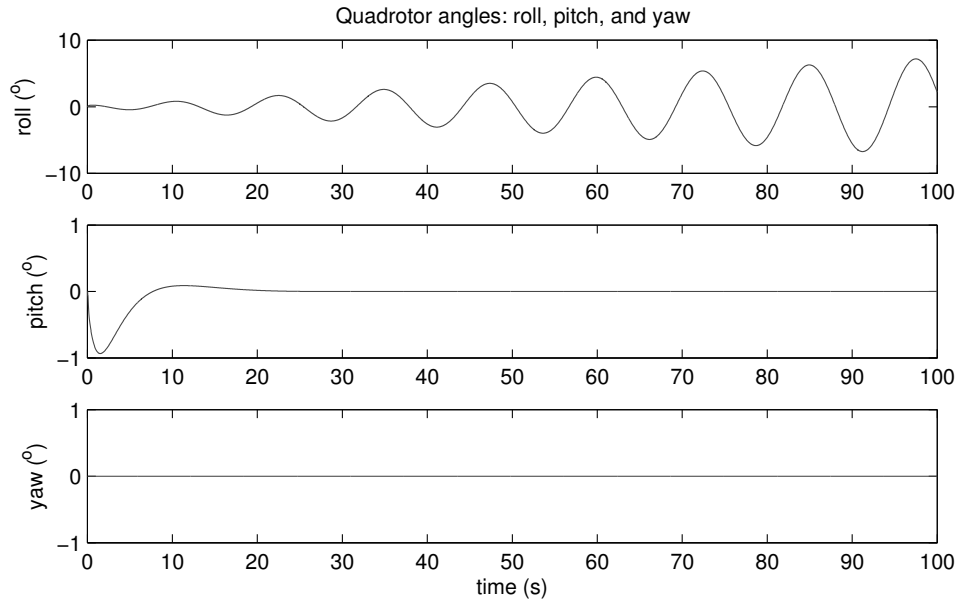


Figure 5.23: Tracking control: quadrotor angle.

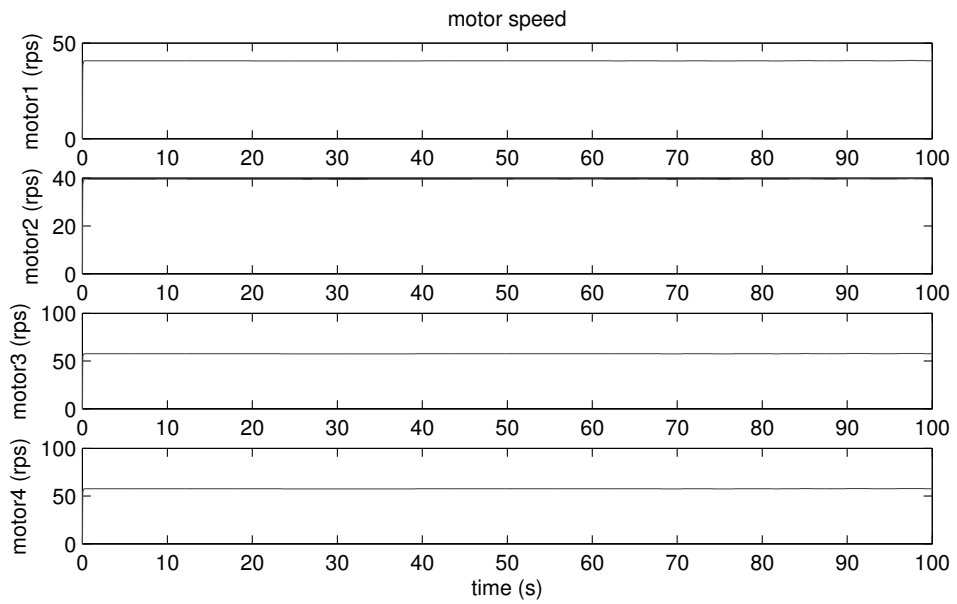


Figure 5.24: Tracking control: quadrotor motor speed.

5.5 Conclusion

In this chapter, a quadrotor UAV is developed. Linear control methods are applied to control position, speed, and angles. Hardware is developed, and practical data has been used in simulations. Simulations are designed for set-point control and tracking, which are quite widely used as a mobile sensor. Simulation results imply that the control is successful. Future work includes the algorithm implementation into practical quadrotor UAV, and test it.

Chapter 6

Conclusions

In this dissertation, an adaptive sampling method with mobile sensor networks is presented. Based on the consideration of sensing resource constraint, computing ability, and onboard energy limitation, a down sampling scheme has been developed with the intent to lower the cost at sampling stage. This dissertation is based on the compressive sensing, which is a typical down sampling method, collecting a small number of measurements and reconstructing original signal with little information loss. Due to the unique features of mobile sensors, measurements are collected sequentially in nature. The proposed adaptive sampling scheme include selection measurements based on sparsity analysis, mobility control of the mobile sensors, and reconstruction of the signals.

As the down-sampling/compressive sensing decreases the number of measurements collected by mobile sensors, the tradeoff is the total information amount from collected measurements has been turned down. As a result, it requires each measurements contains as more information as possible. This dissertation maximized the information amount of each individual measurement by analyzing existing data under Haar Wavelet domain, where the information amount can be easily found. To control the networked mobile sensors, moving cost has been evaluated in addition to the information amount. With enough measurements collected, an MCMC method reconstructed the original signal.

Simulations have been done to justify the adaptive sampling method addressed in the thesis. The proposed method has been implemented into a ground mobile sensor for an indoor mapping application. In this application, a mobile sensor moved adaptively and automatically to collect mapping information and reconstructed a binary map. Based on existing experiments, future experimental plan would focus on sensing field reconstruction using networked mobile sensors, including UAVs and UGVs.

References

- [1] G. Hoffmann and C. Tomlin, “Mobile sensor network control using mutual information methods and particle filters,” in *IEEE Transactions on Automatic Control*, vol. 55, pp. 32–47, 2010.
- [2] N. Yilmaz, C. Evangelinos, P. Lermusiaux, and N. Patrikalakis, “Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming,” in *IEEE Journal of Oceanic Engineering*, vol. 33, pp. 522–537, 2008.
- [3] Y. Mostofi, T. Chung, R. Murray, and J. Burdick, “Communication and sensing trade-offs in decentralized mobile sensor networks: a cross-layer design approach,” in *Fourth International Symposium on Information Processing in Sensor Networks*, pp. 118–125, april 2005.
- [4] R. Baraniuk, “Compressive sensing [lecture notes],” in *IEEE Signal Processing Magazine*, vol. 24, pp. 118–121, 2007.
- [5] A. Singh, R. Nowak, and P. Ramanathan, “Active learning for adaptive mobile sensing networks,” in *Proceedings of 5th International Conference on Information Processing in Sensor Networks*, pp. 60–68, 2006.
- [6] B. Zhang and G. S. Sukhatme, “Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network,” in *IEEE International Conference on Robotics and Automation*, pp. 3673–3680, 2007.
- [7] M. Rahimi, R. Pon, W. Kaiser, G. Sukhatme, D. Estrin, and M. Srivastava, “Adaptive sampling forenvironmental robotics,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3537–3544, 2004.
- [8] T. Chung, V. Gupta, J. Burdick, and R. Murray, “On a decentralized active sensing strategy using mobile sensor platforms in a network,” in *IEEE Conference on Decision and Control*, vol. 2, pp. 1914–1919, 2004.

- [9] P. Ogren, E. Fiorelli, and N. E. Leonard, “Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment,” *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2004.
- [10] J. Leonard and H. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” in *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 376–382, 1991.
- [11] R. Willett, A. Martin, and R. Nowak, “Backcasting: adaptive sampling for sensor networks,” in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pp. 124–133, 2004.
- [12] D. Popa, A. Sanderson, R. Komerska, S. Mupparapu, D. Blidberg, and S. Chappel, “Adaptive sampling algorithms for multiple autonomous underwater vehicles,” in *IEEE/OES Autonomous Underwater Vehicles*, pp. 108–118, 2004.
- [13] M. A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava, and D. Estrin, “Towards event-aware adaptive sampling using static and mobile nodes,” Tech. Rep. 38, University of California, Los Angeles, 2004.
- [14] B. Zhang, G. Sukhatme, and A. Requicha, “Adaptive sampling for marine microorganism monitoring,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1115–1122, 2004.
- [15] M. Mauve, J. Widmer, and H. Hartenstein, “A survey on position-based routing in mobile ad hoc networks,” in *IEEE Network*, vol. 15, pp. 30–39, 2001.
- [16] E. Ayday, F. Delgosha, and F. Fekri, “Location-aware security services for wireless sensor networks using network coding,” in *IEEE International Conference on Computer Communications*, pp. 1226–1234, 2007.
- [17] B. Dil, S. Dulman, and P. Havinga, “Range-based localization in mobile sensor networks,” in *Proceedings of the Third European conference on Wireless Sensor Networks*, pp. 164–179, 2006.
- [18] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *the 9th Annual International Conference on Mobile Computing and Networking*, pp. 81–95, 2003.
- [19] T. Camp, J. Boleng, and V. Davies, “A survey of mobility models for ad hoc network research,” in *Wireless Communications and Mobile Computing: Special Issue on Mobile Ad Hoc Networking Research, Trends and Applications*, vol. 2, pp. 483–502, 2002.

- [20] J. Yi, J. Koo, and H. Cha, "A localization technique for mobile sensor networks using archived anchor information," in *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 64–72, 2008.
- [21] K. Ssu, C. Ou, and H. Jiau, "Localization with mobile anchor points in wireless sensor networks," in *IEEE Transactions on Vehicular Technology*, vol. 54, pp. 1187–1197, 2005.
- [22] S. Zhang, J. Cao, C. Li-Jun, and D. Chen, "Accurate and energy-efficient range-free localization for mobile sensor networks," in *IEEE Transactions on Mobile Computing*, vol. 9, pp. 897–910, 2010.
- [23] A. Baggio and K. Langendoen, "Monte carlo localization for mobile wireless sensor networks," in *Ad Hoc Networks*, vol. 6, pp. 718–733, 2008.
- [24] L. Hu and D. Evans, "Localization for mobile sensor networks," in *10th Annual International Conference on Mobile Computing and Networking*, pp. 45–57, 2004.
- [25] Z. Zhou, Z. Peng, J.-H. Cui, Z. Shi, and A. Bagtzoglou, "Scalable localization with mobility prediction for underwater sensor networks," in *IEEE Transactions on Mobile Computing*, vol. 10, pp. 335–348, march 2011.
- [26] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *22nd Annual Joint Conference of the IEEE Computer and Communications*, vol. 2, pp. 1293–1303, 2003.
- [27] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems*, pp. 299–308, 2002.
- [28] A. Howard, M. J. Matarić, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," in *Autonomous Robots, Special Issue on Intelligent Embedded Systems*, vol. 13, pp. 113–126, 2002.
- [29] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," in *IEEE Transactions on Mobile Computing*, vol. 5, pp. 640–652, 2006.
- [30] M. A. Batalin and G. S. Sukhatme, "Spreading out: A local approach to multi-robot coverage," in *6th International Symposium on Distributed Autonomous Robotic Systems*, pp. 373–382, 2002.
- [31] J. L. Pearce, P. E. Rybski, S. A. Stoeter, and N. Papanikolopoulos, "Dispersion behaviors for a team of multiple miniature robots," in *IEEE International Conference on Robotics and Automation*, pp. 1158–1163, 2003.

- [32] S. Poduri and G. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 165 – 171 Vol.1, 2004.
- [33] J. Tan and N. Xi, "Peer-to-peer model for the area coverage and cooperative control of mobile sensor networks," in *Society of Photo-Optical Instrumentation Engineers*, vol. 5403, pp. 439–450, 2004.
- [34] S. Yang, M. Li, and J. Wu, "Scan-based movement-assisted sensor deployment methods in wireless sensor networks," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, pp. 1108–1121, 2007.
- [35] W. Wang, V. Srinivasan, and K.-C. Chua, "Coverage in hybrid mobile sensor networks," in *IEEE Transactions on Mobile Computing*, vol. 7, pp. 1374 –1387, 2008.
- [36] A. Sekhar, B. Manoj, and C. Murthy, "Dynamic coverage maintenance algorithms for sensor networks with limited mobility," in *IEEE International Conference on Pervasive Computing and Communications*, pp. 51 –60, 2005.
- [37] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 300–308, 2005.
- [38] I. Hussein and D. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," in *IEEE Transactions on Control Systems Technology*, vol. 15, pp. 642–657, 2007.
- [39] S. Thrun and Y. Liu, "Multi-robot slam with sparse extended information filters," in *Robotics Research*, 2005.
- [40] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," in *International Journal of Robotics Research*, vol. 25, pp. 1243–1256, 2006.
- [41] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1384 –1397, 2006.
- [42] H. J. S. Feder, H. Jacob, S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," in *The International Journal of Robotics Research*, 1999.
- [43] A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte, "An experiment in integrated exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 534–539, 2002.

- [44] S. Huang, N. Kwok, G. Dissanayake, Q. Ha, and G. Fang, “Multi-step look-ahead trajectory planning in slam: Possibility and necessity,” in *IEEE International Conference on Robotics and Automation*, pp. 1091 – 1096, 2005.
- [45] C. Leung, S. Huang, and G. Dissanayake, “Active slam using model predictive control and attractor based exploration,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026 –5031, 2006.
- [46] C. Leung, S. Huang, N. Kwok, and G. Dissanayake, “Planning under uncertainty using model predictive control for information gathering,” in *Robotics and Autonomous Systems*, 2006.
- [47] M. Bryson and S. Sukkarieh, “Observability analysis and active control for airborne slam,” in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, pp. 261 –280, 2008.
- [48] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, “A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot,” in *Autonomous Robots*, pp. 93–103, 2009.
- [49] T. Vidal-Calleja, A. Sanfeliu, and J. Andrade-Cetto, “Action selection for single-camera slam,” in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 40, pp. 1567 –1581, 2010.
- [50] M. Bryson and S. Sukkarieh, “Architectures for cooperative airborne simultaneous localisation and mapping,” in *Journal of Intelligent and Robotic Systems*, vol. 55, pp. 267–297, 2009.
- [51] C. Stachniss, D. Hahnel, W. Burgard, and G. Grisetti, “On actively closing loops in grid-based fastslam,” in *Advanced Robotics*, vol. 19, 2005.
- [52] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, “Coordinated multi-robot exploration,” in *IEEE Transactions on Robotics*, vol. 21, pp. 376 – 386, 2005.
- [53] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, “Distributed multirobot exploration and mapping,” in *Proceedings of the IEEE*, vol. 94, pp. 1325 –1339, 2006.
- [54] R. Aragues, J. Cortes, and C. Sagues, “Motion control strategies for improved multi robot perception,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1065–1070, 2009.
- [55] B. Jung and G. S. Sukhatme, “Detecting moving objects using a single camera on a mobile robot in an outdoor environment,” in *International Conference on Intelligent Autonomous Systems*, pp. 980–987, 2004.

- [56] K.-S. Tseng, “A stream field based partially observable moving object tracking algorithm,” in *International Conference on Control, Automation, Robotics and Vision*, pp. 1850–1856, 2008.
- [57] G. Hollinger, S. Singh, J. Djughash, and A. Kehagias, “Efficient multi-robot search for a moving target,” in *The International Journal of Robotics Research*, vol. 28, pp. 201–219, 2009.
- [58] B. P. Gerkey, S. Thrun, and G. Gordon, “Visibility-based pursuit-evasion with limited field of view,” in *International Journal of Robotics Research*, pp. 20–27, 2004.
- [59] A. Kolling and S. Carpin, “Cooperative observation of multiple moving targets: an algorithm and its formalization,” in *The International Journal of Robotics Research*, vol. 26, pp. 935–953, 2007.
- [60] G. Xing, J. Wang, Z. Yuan, R. Tan, L. Sun, Q. Huang, X. Jia, and H. C. So, “Mobile scheduling for spatiotemporal detection in wireless sensor networks,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 1851–1866, 2010.
- [61] F. Zhao, J. Shin, and J. Reich, “Information-driven dynamic sensor collaboration,” in *IEEE Signal Processing Magazine*, vol. 19, pp. 61–72, 2002.
- [62] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, “Detection, classification and tracking of targets in distributed sensor networks,” in *IEEE Signal Processing Magazine*, vol. 19, pp. 17–29, 2002.
- [63] J.-F. Chamberland and V. Veeravalli, “Decentralized detection in sensor networks,” in *IEEE Transactions on Signal Processing*, vol. 51, pp. 407–416, 2003.
- [64] T. Clouqueur, K. Saluja, and P. Ramanathan, “Fault tolerance in collaborative sensor networks for target detection,” in *IEEE Transactions on Computers*, vol. 53, pp. 320–333, 2004.
- [65] J. R. Spletzer and C. J. Taylor, “Dynamic sensor planning and control for optimally tracking targets,” in *The International Journal of Robotics Research*, pp. 7–20, 2003.
- [66] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, “Information-theoretic coordinated control of multiple sensor platforms,” in *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 1521–1526, 2003.
- [67] C. Leung, S. Huang, G. Dissanayake, and T. Furukawa, “Trajectory planning for multiple robots in bearing-only target localisation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3978–3983, 2005.

- [68] G. Hoffmann, S. Waslander, and C. Tomlin, “Mutual information methods with particle filters for mobile sensor network control,” in *IEEE Conference on Decision and Control*, pp. 1019–1024, 2006.
- [69] K. Zhou and S. Roulletiotis, “Optimal motion strategies for range-only constrained multisensor target tracking,” in *IEEE Transactions on Robotics*, vol. 24, pp. 1168–1185, 2008.
- [70] K. Zhou and S. Roulletiotis, “Multi-robot active target tracking with distance and bearing observations,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2209–2216, 2009.
- [71] R. Rao and G. Kesidis, “Purposeful mobility for relaying and surveillance in mobile ad hoc sensor networks,” in *IEEE Transactions on Mobile Computing*, vol. 3, pp. 225–231, 2004.
- [72] Y. Zou and K. Chakrabarty, “Distributed mobility management for target tracking in mobile sensor networks,” in *IEEE Transactions on Mobile Computing*, vol. 6, pp. 872–887, aug. 2007.
- [73] T.-L. Chin, P. Ramanathan, and K. K. Saluja, “Analytic modeling of detection latency in mobile sensor networks,” in *International Conference on Information Processing in Sensor Networks*, pp. 194–201, 2006.
- [74] N. Bisnik, A. Abouzeid, and V. Isler, “Stochastic event capture using mobile sensors subject to a quality metric,” in *12th Annual International Conference on Mobile Computing and Networking*, pp. 98–109, 2006.
- [75] T. Wimalajeewa and S. Jayaweera, “Impact of mobile node density on detection performance measures in a hybrid sensor network,” in *IEEE Transactions on Wireless Communications*, vol. 9, pp. 1760–1769, 2010.
- [76] D. Marthaler and A. L. Bertozzi, “Tracking environmental level sets with autonomous vehicles,” in *Recent Developments in Cooperative Control and Optimization*, 2003.
- [77] J. Clark and R. Fierro, “Cooperative hybrid control of robotic sensors for perimeter detection and tracking,” in *American Control Conference*, vol. 5, pp. 3500–3505, 2005.
- [78] K. Dantu and G. Sukhatme, “Detecting and tracking level sets of scalar fields using a robotic sensor network,” in *IEEE International Conference on Robotics and Automation*, pp. 3665–3672, 2007.
- [79] M. Kemp, A. Bertozzi, and D. Marthaler, “Multi-uuv perimeter surveillance,” in *IEEE/OES Autonomous Underwater Vehicles*, pp. 102–107, 2004.

- [80] C. H. Hsieh, Z. Jin, D. Marthaler, B. Q. Nguyen, D. J. Tung, A. L. Bertozzi, and R. M. Murray, “Experimental validation of an algorithm for cooperative boundary tracking,” in *American Control Conference.*, pp. 1078–1083, 2005.
- [81] Z. Jin and A. Bertozzi, “Environmental boundary tracking and estimation using multiple autonomous vehicles,” in *IEEE Conference on Decision and Control*, pp. 4918–4923, 2007.
- [82] D. Kingston, R. Beard, and R. Holt, “Decentralized perimeter surveillance using a team of uavs,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1394–1404, 2008.
- [83] C. E. Shannon, “Communication in the presence of noise,” in *Proc. Institute of Radio Engineers*, vol. 37, pp. 10–21, 1949.
- [84] E. Candes and M. Wakin, “An introduction to compressive sampling,” in *Signal Processing Magazine, IEEE*, vol. 25, pp. 21–30, 2008.
- [85] D. Donoho, M. Elad, and V. Temlyakov, “Stable recovery of sparse overcomplete representations in the presence of noise,” in *IEEE Transactions on Information Theory*, vol. 52, pp. 6–18, 2006.
- [86] E. Candl s and J. Romberg, “Sparsity and incoherence in compressive sampling,” in *Inverse Problems*, vol. 23, pp. 969–985, 2007.
- [87] E. J. Candes, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” in *Communications on Pure and Applied Mathematics*, vol. 59, pp. 1207–1223, 2006.
- [88] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” in *IEEE Transactions on Information Theory*, vol. 52, pp. 489–509, 2006.
- [89] D. Donoho, “Compressed sensing,” in *IEEE Transactions on Information Theory.*, vol. 52, pp. 1289–1306, 2006.
- [90] R. Baraniuk and P. Steeghs, “Compressive radar imaging,” in *Radar Conference, IEEE*, pp. 128–133, 2007.
- [91] W. Bajwa, J. Haupt, G. Raz, and R. Nowak, “Compressed channel sensing,” in *42nd Annual Conference on Information Sciences and Systems.*, pp. 5–10, 2008.
- [92] D. Malioutov, S. Sanghavi, and A. Willsky, “Compressed sensing with sequential observations,” in *IEEE International Conference on Acoustics, Speech and Signal Processing.*, pp. 3357–3360, 2008.
- [93] Y. Mostofi and P. Sen, “Compressed mapping of communication signal strength,” in *Military Communications Conference, IEEE*, pp. 1–7, 2008.

- [94] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, “Compressive wireless sensing,” in *The Fifth International Conference on Information Processing in Sensor Networks*, pp. 134–142, 2006.
- [95] Y. Mostofi and P. Sen, “Compressive cooperative sensing and mapping in mobile networks,” in *American Control Conference.*, pp. 3397–3404, 2009.
- [96] Z. Tian and G. Giannakis, “Compressed sensing for wideband cognitive radios,” in *IEEE International Conference on Acoustics, Speech and Signal Processing.*, vol. 4, pp. 1357–1360, 2007.
- [97] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, “Single-pixel imaging via compressive sampling,” in *Signal Processing Magazine, IEEE*, vol. 25, pp. 83–91, 2008.
- [98] J. Romberg, “Imaging via compressive sampling,” in *Signal Processing Magazine, IEEE*, vol. 25, pp. 14–20, 2008.
- [99] J. C. Ye, “Compressed sensing shape estimation of star-shaped objects in fourier imaging,” in *Signal Processing Letters, IEEE*, vol. 14, pp. 750–753, 2007.
- [100] J. Haupt, W. Bajwa, M. Rabbat, and R. Nowak, “Compressed sensing for networked data,” in *Signal Processing Magazine, IEEE*, vol. 25, pp. 92–101, 2008.
- [101] R. Berinde and P. Indyk, “Sparse recovery using sparse matrices,” tech. rep., Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory, Jan 10 2008.
- [102] R. G. Baraniuk, M. Davenport, R. A. DeVore, and M. B. Wakin, “A simplified proof of the restricted isometry property for random matrices (aka the johnson-lindenstrauss lemma meets compressed sensing),” in *Constructive Approximation*, 2007.
- [103] R. Baraniuk and P. Steeghs, “Compressive radar imaging,” in *IEEE Radar Conference*, pp. 128–133, 2007.
- [104] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” in *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1999.
- [105] E. J. Candes, M. Wakin, and S. Boyd, “Enhancing sparsity by reweighted ℓ_1 minimization,” tech. rep., California Institute of Technology, 2007.
- [106] J. Tropp, “Greed is good: algorithmic results for sparse approximation,” in *IEEE Transactions on Information Theory*, vol. 50, pp. 2231–2242, 2004.
- [107] C. La and M. Do, “Tree-based orthogonal matching pursuit algorithm for signal reconstruction,” in *IEEE International Conference on Image Processing*, pp. 1277–1280, 2006.

- [108] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, “An architecture for compressive imaging,” in *IEEE International Conference on Image*, pp. 1273–1276, 2006.
- [109] M. A. Sheikh, O. Milenkovic, S. Sarvotham, and R. G. Baraniuk, “Compressed sensing dna microarrays,” tech. rep., Rice University, May 2007.
- [110] J. Tropp, M. Wakin, M. Duarte, D. Baron, and R. Baraniuk, “Random filters for compressive sampling and reconstruction,” in *IEEE International Conference on Acoustics, Speech and Signal*, vol. 3, 2006.
- [111] J. Bobin, J.-L. Starck, and R. Ottensamer, “Compressed sensing in astronomy,” in *IEEE Journal of Signal Processing*, vol. 2, pp. 718 – 726, 2008.
- [112] M. Crouse, R. Nowak, and R. Baraniuk, “Wavelet-based statistical signal processing using hidden markovmodels,” in *IEEE Transactions on Signal Processing*, vol. 6, pp. 886–902, 1998.
- [113] M. F. Duarte, V. Cevher, and R. G. Baraniuk, “Model-based compressive sensing for signal ensembles,” in *47th Annual Allerton Conference on Communication, Control, and Computing*, pp. 244–250, 2009.
- [114] T. Blumensath and M. E. Davies, “Sampling theorems for signals from the union of linear subspaces,” in *IEEE Transactions on Information Theory*, vol. 55, pp. 1872–1882, 2009.
- [115] M. Tipping, “Sparse bayesian learning and the relevance vector machine,” in *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
- [116] S. Ji, Y. Xue, and L. Carin, “Bayesian compressive sensing,” in *IEEE Transactions on Signal Processing*, vol. 56, 2008.
- [117] J. Meng, H. Li, and Z. Han, “Sparse event detection in wireless sensor networks using compressive sensing,” in *43rd Annual Conference on Information Sciences and Systems*, pp. 181 –185, march 2009.
- [118] L. He and L. Carin, “Exploiting structure in wavelet-based bayesian compressive sensing,” in *IEEE Transactions on Signal Processing*, vol. 57, pp. 3488–3497, 2008.
- [119] X. Tan and J. Li, “Compressed sensing via sparse bayesian learning and gibbs sampling,” in *IEEE 13th Digital Signal Processing Workshop and 5th Signal Processing Education Workshop*, pp. 690–695, 2009.
- [120] D. Baron, S. Sarvotham, and R. G. Baraniuk, “Bayesian compressive sensing via belief propagation,” in *IEEE Transactions on Signal Processing*, vol. 58, pp. 269–280, 2010.

- [121] S. Babacan, R. Molina, and A. Katsaggelos, “Bayesian compressive sensing using laplace priors,” in *IEEE Transactions on Image Processing*, vol. 19, pp. 53–63, 2010.
- [122] T. Zhang, S. Madhani, and E. van den Berg, “Sensors on patrol (sop): using mobile sensors to detect potential airborne nuclear, biological, and chemical attacks,” *MILCOM, IEEE*, pp. 2924–2929, 2005.
- [123] Y. Dong, W.-K. Hon, and D. Yau, “On area of interest coverage in surveillance mobile sensor networks,” in *IEEE International Workshop on Quality of Service*, pp. 87–90, 2007.
- [124] S. Thrun, “Robotic mapping: A survey,” in *Exploring Artificial Intelligence in the New Millenium*, pp. 1–35, 2002.
- [125] C. P. Robert and G. Casella, “Monte carlo statistical methods,” 2nd Edition 2004.
- [126] V. Rankov, R. J. Locke, R. J. Edens, P. R. Barber, and B. Vojnovic, “An algorithm for image stitching and blending,” *SPIE*, vol. 5701, pp. 190–199, 2005.
- [127] L. Merino, J. Wiklund, F. Caballero, A. Moe, J. De Dios, P.-E. Forssen, K. Nordberg, and A. Ollero, “Vision-based multi-UAV position estimation,” in *Robotics Automation Magazine, IEEE*, vol. 13, pp. 53–62, 2006.
- [128] 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 *Proceedings of the IEEE Conference on Decision and Control*, 2004.
- [129] G. T. Toussaint, “The relative neighbourhood graph of a finite planar set,” in *Pattern Recognition*, vol. 12, pp. 261–268, 1980.
- [130] D. Lowe, “Object recognition from local scale-invariant features,” in *IEEE International Conference on Computer Vision*, 1999.
- [131] J. Tropp and A. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” in *IEEE Transactions on Information Theory*, vol. 53, pp. 4655–4666, 2007.
- [132] T. Blumensath and M. Davies, “Gradient pursuits,” in *IEEE Transactions on Signal Processing*, vol. 56, pp. 2370–2382, 2008.
- [133] A. Salih, M. Moghavvemi, H. Mohamed, and K. Gaeid, “Modelling and pid controller design for a quadrotor unmanned air vehicle,” in *IEEE International Conference on Automation Quality and Testing Robotics*, 2010.
- [134] A. Tayebi and S. McGilvray, “Attitude stabilization of a four-rotor aerial robot,” in *IEEE Conference on Decision and Control*, 2004.

- [135] T. Bresciani, "Modeling, identification and control of a quadrotor helicopter," in *Master Thesis, Department of Automatic Control, Lund University*, 2008.
- [136] S. Bouabdallah, A. Noth, and R. Siegwart, "Pid vs lq control techniques applied to an indoor micro quadrotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2451 – 2456, 2004.
- [137] A. Tayebi and S. McGilvray, "Attitude stabilization of a vtol quadrotor aircraft," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 562 – 571, 2006.
- [138] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *IEEE International Conference on Robotics and Automation*, vol. 5, pp. 4393 – 4398 Vol.5, 2004.
- [139] P. Castillo, A. Dzul, and R. Lozano, "Real-time stabilization and tracking of a four-rotor mini rotorcraft," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 4, pp. 510 – 516, 2004.
- [140] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini-rotorcraft having four rotors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2693 – 2698, 2004.
- [141] S. Salazar-Cruz, A. Palomino, and R. Lozano, "Trajectory tracking for a four rotor mini-aircraft," in *IEEE Conference on Decision and Control and 44th European Control Conference*, pp. 2505 – 2510, 2005.
- [142] B. Andrievsky, A. Fradkov, and D. Peaucelle, "Adaptive control experiments for laas "helicopter" benchmark," in *International Conference on Physics and Control*, pp. 760 – 766, 2005.
- [143] Y. Morel and A. Leonessa, "Direct adaptive tracking control of quadrotor aerial vehicles," in *Florida Conference on Recent Advances in Robotics*, pp. 1–6, 2006.
- [144] P. Castillo, R. Lozano, and A. Dzul, "Stabilization of a mini rotorcraft with four rotors," *Control Systems, IEEE*, vol. 25, pp. 45 – 55, dec 2005.
- [145] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3255 – 3260, 2006.
- [146] T. Madani and A. Benallegue, "Backstepping sliding mode control applied to a miniature quadrotor flying robot," in *32nd Annual IEEE Industrial Electronics Conference*, pp. 700 – 705, 2006.

- [147] T. Madani and A. Benallegue, “Control of a quadrotor mini-helicopter via full state backstepping technique,” in *IEEE Conference on Decision and Control*, pp. 1515 –1520, 2006.
- [148] A. Mokhtari and A. Benallegue, “Dynamic feedback controller of euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle,” in *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2359 – 2366, 2004.
- [149] V. Mistler, A. Benallegue, and N. M’Sirdi, “Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback,” in *IEEE International Workshop on Robot and Human Interactive Communication*, pp. 586 –593, 2001.
- [150] N. Guenard, T. Hamel, and R. Mahony, “A practical visual servo control for an unmanned aerial vehicle,” in *IEEE Transactions on Robotics*, vol. 24, pp. 331 –340, 2008.
- [151] G. P. Tournier, M. Valenti, and J. P. How, “Estimation and control of a quadrotor vehicle using monocular vision and moire patterns,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006.
- [152] N. Metni, T. Hamel, and F. Derkx, “Visual tracking control of aerial robotic systems with adaptive depth estimation,” in *IEEE Conference on Decision and Control*, pp. 6078 – 6084, 2005.
- [153] E. Altug, J. Ostrowski, and C. Taylor, “Quadrotor control using dual camera visual feedback,” in *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 4294 – 4299, 2003.
- [154] M. Earl and R. D’Andrea, “Real-time attitude estimation techniques applied to a four rotor helicopter,” in *IEEE Conference on Decision and Control*, vol. 4, pp. 3956 – 3961, 2004.
- [155] C. Coza and C. Macnab, “A new robust adaptive-fuzzy control method applied to quadrotor helicopter stabilization,” in *Annual meeting of the North American Fuzzy Information Processing Society*, pp. 454 –458, 2006.
- [156] J. Dunfied, M. Tarbouchi, and G. Labonte, “Neural network based control of a four rotor helicopter,” in *IEEE International Conference on Industrial Technology*, vol. 3, pp. 1543 – 1548 Vol. 3, 2004.
- [157] S. Waslander, G. Hoffmann, J. S. Jang, and C. Tomlin, “Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3712 – 3717, 2005.

- [158] A. Kallapur, I. Petersen, and S. Anavatti, "A robust gyroless attitude estimation scheme for a small fixed-wing unmanned aerial vehicle," in *Asian Control Conference*, pp. 666–671, 2009.
- [159] E. Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter," in *Virtual Reality Annual International Symposium, IEEE*, pp. 185–194, 267, 1996.
- [160] H. Luinge, P. Veltink, and C. Baten, "Estimation of orientation with gyroscopes and accelerometers," in *BMES/EMBS Conference*, vol. 2, p. 844 vol.2, 1999.
- [161] J. Marins, X. Yun, E. Bachmann, R. McGhee, and M. Zyda, "An extended kalman filter for quaternion-based orientation estimation using marg sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, pp. 2003–2011 vol.4, 2001.
- [162] A. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," in *IEEE Transactions on Biomedical Engineering*, vol. 53, pp. 1346–1356, 2006.
- [163] H. Luinge and P. Veltink, "Measuring orientation of human body segments using miniature gyroscopes and accelerometers," in *Medical and Biological Engineering and Computing*, vol. 43, pp. 273–282, 2005.
- [164] D. Jurman, M. Jankovec, R. Kamnik, and M. Topic, "Calibration and data fusion solution for the miniature attitude and heading reference system," in *Sensors and Actuators A: Physical*, vol. 138, pp. 411–420, 2007.
- [165] M. Haid and J. Breitenbach, "Low cost inertial orientation tracking with kalman filter," *Applied Mathematics and Computation*, vol. 153, no. 2, pp. 567–575, 2004.
- [166] D. Roetenberg, H. Luinge, C. Baten, and P. Veltink, "Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 395–405, 2005.
- [167] D. Gebre-Egziabher, R. Hayward, and J. Powell, "Design of multi-sensor attitude determination systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 2, pp. 627–649, 2004.
- [168] H. Q. P. Nguyen, H.-J. Kang, Y. S. Suh, and Y. S. Ro, "A dcm based orientation estimation algorithm with an inertial measurement unit and a magnetic compass," *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 859–876, 2009.
- [169] R. Hyde, L. Ketteringham, S. Neild, and R. Jones, "Estimation of upper-limb orientation based on accelerometer and gyroscope measurements," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 2, pp. 746–754, 2008.

- [170] S. O. Madgwick, “An efficient orientation filter for inertial and inertial/magnetic sensor arrays,” tech. rep., 2010.

Appendix

Appendix A Copyright Permission for Chapter 3 and 4

The following letter from IEEE gives permission to reprint Chapter 3 and Chapter 4.

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.