

**ADAPTIVE WIRELESS SENSOR NETWORK FOR REAL-TIME
MONITORING OF WATER QUALITY**

by

Jiahong Chen

B.Eng., Xiamen University, 2015

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

May 2017

© Jiahong Chen, 2017

Abstract

Water quality problems have appeared in many places all around the world, and have caused severe public health problems. In identify the quality of different aquatic environments, wireless sensor networks have been used for monitoring large geographic areas of interest (AOI). Among the challenges of using wireless sensor networks for water quality monitoring in large areas, sensor node deployment strategy is a key consideration since an optimal sensor node deployment strategy can ensure the most appropriate utilization of the limited monitoring resources (sensor node, incorporated sensors, power supply, monitoring rates, etc.). To tackle such problems, we in the Industrial Automation Laboratory (IAL) of the Department of Mechanical Engineering, the University of British Columbia (UBC) have developed a mobile wireless sensor network for water quality monitoring. It has mobile (dynamic) sensor nodes, which can move to best sensing locations, and the ability to sense key water quality attributes. The developed platform is equipped with multiple nodes each of which having basic water quality detecting sensor probes, supports up to six propellers, and has upgradable wireless communication boards. Besides, we have also proposed an optimal sensor node deployment strategy called “Rapid Random exploring tree with Linear Reduction” (RRLR) for this mobile wireless sensor network. The proposed method removes redundant sensor nodes depending on the linear dependence of sensor readings at the current deployment location without losing information. In this manner, spatial-temporal correlation of sensor node deployment in large geographic AOI can be minimized. The developed platform is demonstrated to have good performance even when moving against water flow and has low packet loss rate (0.85%) while transmitting data under different types of obstacles in real-world tests. Furthermore, the developed optimal sensor node deployment strategy, RRLR, requires nearly 60% fewer sensor nodes to achieve the same estimation error as our benchmark.

Lay Summary

This thesis makes two contributions. First, a practical platform is developed for real-time monitoring of water quality using a network of mobile sensors having sensor mobility, efficient data transmission, and less costly has easy-to-upgrade modular design. Second, a novel strategy is proposed for optimal sensor deployment. In particular theorems are presented and proved for eliminating linear redundant sensor nodes, and minimizing the number of sensor nodes for acquiring the same information content. Also, a lower prior estimation error and a lower linear dependence among the deployed sensor nodes are achieved with the developed methodology.

Preface

This thesis is an intellectual property of the author, Jiahong Chen, under supervision and guidance of Dr. Clarence W. de Silva, Professor of Mechanical Engineering, The University of British Columbia. Dr. Clarence W. de Silva proposed and supervised the overall water quality monitoring project, acquired funding and resources for the project, suggested the topic of the thesis, provided concepts and methodologies in addressing problems in the topic, provide research facilities in his Industrial Automation Laboratory, and revised the thesis presentation. All aspects of work, except the platform development and field test in Chapter 2, including literature review, design and implementation of the proposed algorithm, algorithm simulation based on field test sampling data, analysis and discussion of results were performed solely by the author of this thesis. My colleagues Teng Li, Tongxin Shu, Zhuo Chen and I collaboratively developed the mobile sensor node platform, under the supervision and guidance of Dr. Clarence W. de Silva, which was used for real-world field tests. Dynamic and navigation system of the platform was built together, while data communication module and data processing module were developed by the author of this thesis alone, all of which were carried out under the supervision and guidance of Dr. Clarence W. de Silva.

Table of Contents

Abstract.....	ii
Lay Summary	iii
Preface.....	iv
Table of Contents	v
List of Figures.....	vii
List of Abbreviations	x
Acknowledgement.....	xi
Chapter 1 : Introduction	1
<i>1.1 Motivation and Background</i>	<i>1</i>
<i>1.2 Work on Platform Design</i>	<i>2</i>
<i>1.3 Literature Review on Optimal Sensor Deployment</i>	<i>5</i>
<i>1.4 Problem Formulation.....</i>	<i>9</i>
<i>1.4.1 System Model</i>	<i>9</i>
<i>1.4.2 Energy Consumption Analysis</i>	<i>10</i>
<i>1.4.2.1 Energy for Node Movement</i>	<i>10</i>
<i>1.4.2.2 Communication Energy</i>	<i>11</i>
<i>1.4.3 Problem Formulation.....</i>	<i>12</i>
<i>1.5 Thesis Organisation.....</i>	<i>12</i>
Chapter 2 : Design of the Water Quality Monitoring System	14

2.1 Design of an Adaptive Water Quality Monitoring System.....	14
2.2 Design of Mobile Water Quality Monitoring Sensor Node	21
2.3 Sensor Node Deployment and Field Testing.....	25
2.4 Conclusion	28
Chapter 3 : Linearly Redundant Sensor Reduction	30
3.1 Linearly Dependent Neighbor Clustering.....	30
3.2 Conclusion	38
Chapter 4 : Optimal Sensor Node Deployment with Linear Reduction	39
4.1 Modeling of Environment	40
4.2 Estimation Quality of Deployment Locations	42
4.2 Optimization Scheme for Sensor Node Deployment	43
4.3 Experimental Results	47
Chapter 5 : Conclusions and Future Work	51
5.1 Contributions	51
5.2 Possible Future Work	51
Bibliography	53

List of Figures

Figure 1.1: Aquatic-based Networked Info-mechanical System [17].	3
Figure 1.2: Static Wireless Sensor Node [18].	4
Figure 1.3: HERON by Clearpath Robotics [19].	5
Figure 2.1 System Architecture of the ICT Platform.	15
Figure 2.2: Graphical User Interface for Water Quality Monitoring System.	16
Figure 2.3: (a) Data Transmission Solution From Libelium; (b) Enhanced Data Transmission Framework.	18
Figure 2.4: Enhanced Sensor Node with Wireless Network Extender.	20
Figure 2.5: Local Network Based GUI.	21
Figure 2.6: Mobile Sensor Node Operation.	23
Figure 2.7: Developed Mobile Sensor Node.	24
Figure 2.8: Controller of the Mobile Sensor Node.	25
Figure 2.9: Sensor Node, Local Server and Gateway Deployment in the ICICS Building.	26
Figure 2.10: Deployment of Sensor Nodes in UBC.	27
Figure 2.11: Sensor Readings of the Field Test at UBC.	27
Figure 2.12: Location of Field Testing in Ulhas River, Neral, India.	28
Figure 4.1: Framework of Optimal Sensor Node Deployment with Linear Reduction.	39
Figure 4.2: Example of Spatiotemporal Field for Different Times.	42
Figure 4.3: Deployment Locations Produced by RRLR.	48
Figure 4.4: Deployment Results Generated by RRC.	49
Figure 4.5: Cost Comparison between RRLR and RRC.	50

List of Symbols

$E_{\text{mob}}(n_i)$	Energy model of moving sensor node
$c_i(x, y)$	Gaussian basis function
k_{mob}	Moving coefficient
\mathbf{m}_i	Single sensor reading for a period
$v_t(x, y)$	Sensor reading in location (x, y) at time t
Σ_{∞}^i	Prior estimation error for one vertex when time goes to infinity
Σ_t	Prior estimation error
$\tilde{\sigma}$	One deployment strategy
A	Temporal information matrix
\mathcal{A}	Geographical area of interest
$\mathcal{C}(x, y)$	Spatial information matrix
E	Edge in the graph
$G = (V, E)$	Graph containing spanning tree
$J(\tilde{\sigma})$	Minimal largest eigenvalue of a deployment strategy
M	Sensor reading matrix of different sensors for a period
\mathcal{M}	Set containing sensor readings from different sensor nodes
\mathcal{N}	Set of sensor nodes
\mathcal{P}	Set of sensor positions
\mathcal{T}	Set of computed deployment locations
V	Vertex in the graph
ρ	Step size

$\rho(\Sigma_t)$

Eigenvalue of prior covariance matrix

$\psi(n_i)$

Distance travelled

List of Abbreviations

CAU	Central Assessment Unit
CPU	Central Processing Unit
DO	Dissolved Oxygen
ESC	Electronic Speed Control
GPS	Global Positioning System
ICT	Information and Communication Technologies
IMU	Inertial Measurement Unit
MSE	Mean Square Estimation Error
ORP	Oxidation-Reduction Potential
ROV	Remote Operating Underwater Vehicle
RRC	Rapidly-exploring Random Circle
RRLR	Rapid Random exploring tree with Linear Reduction
SDK	Software Development Kit
TCP	Transmission Control Protocol
UPS	Uninterruptible Power Supply
USB	Universal Serial Bus
USV	Unmanned Surface Vehicle
WQI	Water Quality Index

Acknowledgement

First, I wish to express my enduring appreciation and gratitude to my supervisor, Professor Clarence W. de Silva, who has continuously given to me rigorous advice, guidance and encouragement for the entire research program that is presented in this thesis. The opportunity, resources, and funding that he has provided for me to work in the Industrial Automation Laboratory (IAL) under his directorship have given me a life-changing and stimulating experience for me. Also, I would like to thank my colleagues, Teng Li, Tongxin Shu, Zhuo Chen, who worked on same project with me, gave me much support and cooperated in building the dynamic navigation system of the platform.

This research presented in this thesis has been supported by IC-IMPACTS (Canada-India Research Center of Excellence), and research grants from Canada Foundation for Innovation (CFI), the Natural Sciences and Engineering Research Council (NSERC), the British Columbia Knowledge Development Fund (BCKDF), and the Tier 1 Canada Research Chair in Mechatronics and Industrial Automation, all held by Dr. Clarence W. de Silva.

I wish to express my thanks to my lab mates Teng Li, Dr. Muhammad Tufail, Yu Du, Hani Balkhair, Lili Meng, Min Xia, Shujun Gao, Zhuo Chen, Tongxin Shu, Lucas Falch and Swapna Premasiri of IAL, who have been supporting and inspiring me on both academic affairs and personal life.

In particular, I give special thanks to my family for their love and support throughout my upbringing.

Chapter 1 : Introduction

1.1 Motivation and Background

Poor quality of water can lead to crisis situations as seen in many places all around the world. Occurring in both developing and developed countries, water quality degradation can put public health in great danger [1] [2] [3]. A platform that can monitor the quality of water in various sources and aquatic environments in real time can help address such problems quickly [4] [5]. To detect the quality of aquatic environments, wireless sensor networks have been used, which can monitor large geographic areas of interest (AOI) [6] [7] [8]. The present work concerns the development and implementation such a network with mobile/dynamic sensor nodes. Among challenges of using wireless sensor networks with mobile sensor nodes for water quality monitoring in large geographic AOI, a leading one is finding an optimal sensor node deployment strategy. An optimal sensor node deployment strategy can ensure effective utilization of limited sensor nodes and power resources, while acquiring the needed information rapidly in the field [9] [10] [11]. In order to address such issues, we in the Industrial Automation Laboratory (IAL) of the Department of Mechanical Engineering, the University of British Columbia, have developed a mobile wireless sensor network for water quality monitoring. It has the ability to sense key water quality parameters while autonomously moving the sensor nodes to best sensing locations. The optimal location of the sensor nodes may be achieved by considering both spatial and temporal correlation [8] [12] [13] [14] [15] [16]. The developed platform is equipped with basic water quality monitoring sensor probes, and propellers for autonomously moving the nodes possibly against strong water flow conditions and currents, and upgradable wireless communication boards with low packet loss rates. The optimal sensor node deployment strategy that is developed in the present work is termed Rapid Random exploring tree with Linear Reduction (RRLR). It has shown

good performance by reducing the required number of sensor nodes while achieving low prior estimation error for the water quality.

1.2 Work on Platform Design

A real-time adaptive wireless sensor network for water quality monitoring uses sensor nodes to acquire the required information in an interested segment of water. Researchers and companies have been involved in the design and development of sensor nodes, sensor networks, and unmanned surface vehicles (USVs), which are all relevant to this application. Since water quality monitoring sensors and dynamic multi-sensor nodes are expensive, the densely deployment of them can be prohibitive. Hence, it is desirable to find ways to deploy the minimum number of sensor nodes, at low cost, and collect as much information as possible. It follows that the design of inexpensive and autonomously mobile sensor nodes and also develop strategies for optimizing the sensor node deployment locations.

Researchers have worked in the development of sensors to monitor aquatic environments. Singh et al. proposed an aquatic-based networked info-mechanical system, which is a single mobile node with the ability to relocate and easily assemble [17]. However, as shown in Figure 1.1 their system only has a single node to detect water quality, which is not adequate for monitoring large AOI. Also, it is not robust to sudden environmental changes. Moreover, their system is not self-propelling, and is driven using a cable hold by a technician. This method is impractical since it is hard to relocate a sensor node using a drive cable when monitoring a large area, and also it is hard to control when the water flow is strong. Besides, it is a single degree-of-freedom system, which can only move along one axis, and cannot go to any desired position.

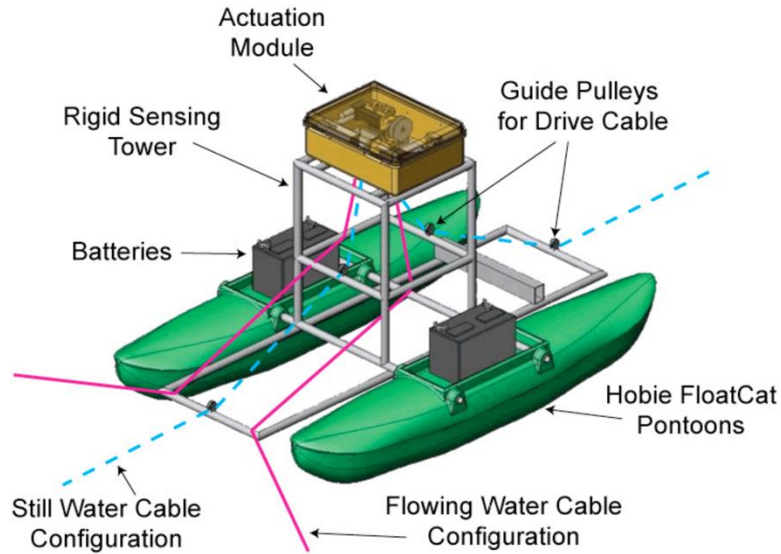


Figure 1.1: Aquatic-based Networked Info-mechanical System [17].

Alippi et al. designed a static wireless sensor network for large AOI monitoring [18]. In their sensor network, each sensor node had the ability of rapid data transmission and analysis. Figure 1.2 shows the design for their static sensor node, which integrated a data acquisition board, a wireless communication board and a solar panel. The sensor nodes are able to communicate with each other for exchanging and analysing the data, to determine the health condition of the coralline barrier and to provide quantitative indications related to cyclone formations in tropical areas. Since their framework only includes static sensor nodes, which cannot relocate as appropriate for data collection, it may not collect all necessary information from the environment.



Figure 1.2: Static Wireless Sensor Node [18].

Recently, some companies developed unmanned surface vehicles (USV) for multiple purposes such as oceanography, underwater video recording, and water quality investigation. USVs are far more capable than static nodes since they are able to navigate autonomously to desired locations, and are less costly than research vessels that needs to carrying technicians, scientists, and heavy and complex equipment. A water quality monitoring system will require a lightweight USV as the carrier for the sensors, data acquisition module, wireless communication board, and data processing module. Clearpath Robotics developed a mid-sized surface vessel named HERON for water sampling [19] as shown in Figure 1.3. It is a self-propelling unmanned surface vessel with built in GPS receiver for self-location. Furthermore, it contains open source SDK for system development, and has a high payload capacity, which are desirable features. However, it is too expensive for multiple node deployment since the cost of a single node is about fifty thousand USD for a single robot. Also, it is very heavy (around 28 kg) and is not suitable for deployment in rural areas.



Figure 1.3: HERON by Clearpath Robotics [19].

Blue Robotics developed a small-size remotely operated vehicle [20], which was targeted for discovering underwater environments. It is cost effective with powerful propellers that can easily navigate in rough waters, and also provides open source SDK that greatly facilitates system developments. However, it is intended for discovering underwater environments and cannot float. All its signals are transmitted through a cable, which method is not suitable for long range control in the field with such obstacles as trees.

1.3 Literature Review on Optimal Sensor Deployment

There exist several strategies for optimal sensor deployment location. Among them, a common approach is as follows. First, densely deploy sensor nodes in the field of interest. Then, according to the spatial and temporal correlation of the sensor node location and their sensor reading [21], keep the most important sensor nodes and remove the unnecessary (redundant) ones depending on the spatial and temporal correlation [10] [22] [23], or transmit only those data that are not identical,

in order to reduce the communication cost [24] [25] [26] [27] [28] [29]. As verified through experiments, driving the sensor nodes to desired locations cost much more energy than do communication and sampling. Hence, using the least number of sensor nodes and requiring fewer locations for the nodes to monitor are the main objectives of the deployment strategy. Gupta et al. proposed a novel strategy to determine the linear dependence of sensor readings among sensor nodes, and remove sensors in the network that provide linearly dependant information [10]. This method greatly reduces the number of active sensor nodes deployed in the field, while collecting the required information. However, this method requires pre-deployment of a large number of static sensor nodes, which is costly and not robust to environmental changes. Liu et al. proposed a model for densely deployed nodes based on balanced energy consumption [30]. But it is densely deployed and considers only the deployment density of nodes, while not manipulating the node movement. Villas et al. investigated both spatially correlated information and redundant data collected by nearby sensor nodes inside the densely deployed sensor network to reduce the active sensor nodes [15] [30]. These methods greatly reduced the energy consumption while achieving a high event detection rate. However, it too requires deployment of many sensor nodes while saving energy only during data transmission. Mocanu et al. proposed a method for establishing redundancy reduction metrics based on network centrality [31]. It focused on exploiting centrality of each node in the sensor network topology and only kept the nodes having certain centrality. However, the information of the entire AOI might be lost during this process because centrality only considers the connectivity among the nodes but not the amount of information that a node collects. This method also requires the deployment of many static sensor nodes in the initial stage, which is costly and impractical.

Besides dense deployment and redundancy reduction of sensor nodes, researchers have considered the sensor node deployment from the objective of area coverage, by assuming that each sensor node can detect at a specific probability the events that happen within a specific sensing radius. Both fixed and dynamic nodes may be considered [32] [33] where a dynamic node has mobility to relocate itself [34]. Mohammed et al. used a multi-objective artificial immune system (AIS) algorithm to determine the optimal coverage rate subjected to constraints such as communication energy consumption, distance which the mobile nodes have to move, and so on [35] [36] [37]. However, they did not consider the correlation of the collected data, which will indicate redundant deployment location of sensor nodes, which amounts to wastage of resources. Alsheikh et al. employed a neural network to estimate the sensor values at uncovered locations by studying the nonlinear relationship of the deployed static sensor nodes [38]. This approach is impractical and cannot detect sudden environmental change in the uncovered area. This is because such information is not acquired by the deployed static sensor nodes, and cannot be estimated as a result. Attea et al. proposed an multi-objective optimization method that sought both high area coverage rate and long network life [39]. Aziz et al. utilized particle swarm optimization (PSO) and Voronoi diagram to search for highest area coverage rate [40]. Ganganath et al. applied an anti-flocking algorithm to solve the dynamic coverage problem by deploying sensor nodes separately and maximizing the total area covered [41]. Li et al. designed a k -coverage sensor deployment strategy that optimized certain objectives in relatively sparse WSNs [42]. Although the area coverage problems are suitable for optimization that determines best locations for sensor nodes, they have a common drawback due to the underlying assumption that a sensor has a specific probability to detect events within its sensing radius. Although this assumption may be true for active sensor nodes, which emit active signals (e.g., microwave, radar, sonar) to detect events, it

is impractical for passive sensor nodes, which can only detect the events transmitted from the source to the location where it is being deployed (e.g., temperature sensors, pH value sensors, dissolved oxygen sensors, electrical conductivity sensors, and oxidation-reduction potential sensors, which we used in water quality monitoring). Hence, the assumption that a passive sensor has a sensor radius that covers a specific AOI is not realistic. Hence, a more realistic formulation is needed for water quality monitoring using a wireless sensor network.

Recently, researchers have improved the problem formulation. They have deployed a limited number of sensor nodes in locations that can provide most useful information [9] or can achieve least estimation error [13]. Guestrin et al. first proposed a mutual information criterion that produces better sensor node placement than entropy [43]. They tested this criterion on two real world datasets and determined that mutual information could achieve lower root mean square (RMS) for prediction [9]. However, their method only selects the best deploy locations from the environmental model of static sensor network and does not consider further environmental changes. If the environment changes, the chosen node locations may not be accurate. Du et al. proposed an optimal sensor placement strategy for monitoring hydrodynamics and water quality in an urban district, using entropy [43] and mutual information [9] to gather most information in selected locations [8]. Although they deployed the sensor network for monsoon seasons, which follows a different distribution, they do not provide a strategy for better detection of sudden changes in the environment. In the case of water quality monitoring, the quality might experience sudden changes due to unplanned events (floods, biological waste, fertilizer pollution, industrial effluent, etc.), which require quick detection and response. This necessitates moving the limited number of mobile sensor nodes to new locations that can provide most useful information about the changes.

Lan et al. proposed a method to estimate a spatiotemporal field in a dynamic environment by deciding which locations in the field which one or more robots should reach to acquire data [44] [45] [46] [47]. This algorithm first models the environment using past data. Based on that, mobile sensor nodes are deployed to explore the environment of interest. The locations are determined whose data minimize the largest eigenvalue of the error covariance matrix of the field estimate error over an infinite horizon. By doing so, they aim for the minimal number of sensor nodes that best estimate the environment. However, their method does not consider the correlation among sensory data from different sampling locations, which might make some sensor nodes redundant, since their information may be inferred from other nodes.

1.4 Problem Formulation

This section presents the model for the proposed real-time adaptive mobile wireless sensor network and then formulates the problem solution with the relevant mathematical details.

1.4.1 System Model

Assume that the sensor network consists the set of sensor nodes $\mathcal{N} = \{n_1, n_2, \dots, n_L\}$, where L is the number of nodes, $|\mathcal{N}| = L$. Also the set $\mathcal{P} = \{p_1, p_2, \dots, p_L\}$, represents the positions of the nodes. For mobility, each sensor node is equipped with some mechanisms (e.g., DC motor, propeller), which can propel the node to a required position, as in [48] [49]. Initially, the sensor nodes are arbitrarily deployed in a 2-D AOI, \mathcal{A} . The shape of this area can be either rectangular or irregular. The goal is to determine the proper locations $\mathcal{P} \in \mathcal{A}$ for sensory data acquisition, so

as to provide best estimates for the water quality in the entire environment while acquiring most information.

1.4.2 Energy Consumption Analysis

There are two main categories of energy consumption in a mobile wireless sensor network. The first category represents the continuously consumed energy for sensing. The second category consists of sporadically consumed energy, which includes energy for mobile node movement, data communication, and computation. It is known that the quantity of continuously consumed energy is small when compared to sporadically consumed energy. Also, the former can be greatly reduced/optimized through power management. So, the present analysis does not consider continuously consumed energy. Among various types of sporadically consumed energy, the one used for driving the mobile sensor nodes in the aquatic environment is the primary one. It usually consumes most energy and if the mobile sensor node needs to relocate frequently, the battery power will quickly run out. Then, without power, the dynamic sensor network will be disabled. The present work focuses on minimizing the distance travelled by the mobile sensor nodes to reach the desired locations for data acquisition, specifically by reducing the number of locations of sensor node deployment.

1.4.2.1 Energy for Node Movement

The Rapid-exploring Random tree with infinite horizon cost and Linear Reduction (RRLR) algorithm aims to achieve optimal coverage ratio, based on spatial data correlation in a dynamic monitoring environment and a dynamic sensor network. Thus, it requires the ability of the sensor nodes to move to optimal sensing locations, which consumes energy. The amount of energy

consumed by a drive propeller is incorporated an energy model for each node n_i , which is denoted by $E_{mob}(n_i)$. This is a proportional to the distance that the node travels until the sensor network is settled [37] [50]:

$$E_{mob}(n_i) = k_{mob} \cdot \psi(n_i) \quad (1-1)$$

where $\psi(n_i)$ is the total distance the node n_i travels during the entire relocating process, and k_{mob} is the coefficient of energy consumption. Since sensors are able to collect only the instant location information via GPS, field imaging positioning [51], and so on, in a certain time interval τ_{mob} , the distance that node n_i travels in a unit time (i.e., instantaneous speed) at time t is given by:

$$\Delta\psi^t(n_i) = \|\mathbf{p}_i^{t+1} - \mathbf{p}_i^t\| \quad (1-2)$$

where \mathbf{p}_i^t is the position of node n_i at time t . Hence, the total distance a node travels can be expressed as:

$$\psi(n_i) = \int_{t_0}^{t_{end}} \Delta\psi^t(n_i) dt \quad (1-3)$$

where t_0 is the starting time and t_{end} is the end time, when the node has reached its goal location. Since it considers the movement at each time instant (instantaneous speed), $\psi(n_i)$ takes into account both not uniform speed of a sensor node and nonlinear relocating path.

1.4.2.2 Communication Energy

Energy is dissipated as well while transmitting and receiving information. It varies quadratically with the communication radius R_C and also increases with the number of message exchanged [52] [53], while payload size does not directly affect the communication energy [51]. To conserve power, the frequency of message exchange should be reduced, by transmitting bundles of L sensor readings at a time instead of each sensor reading, for spatial correlation calculation. Furthermore,

sensor data compression can be used for reducing the size of the data packets that are transmitted, by using an energy-efficient data compression procedure [54].

1.4.3 Problem Formulation

The problem that has been formulated in this chapter involves the following. A set of sensor nodes $\mathcal{N} = \{n_1, n_2, \dots, n_k\}$ is deployed in the geographical AOI, at positions $\mathcal{P} = \{p_1, p_2, \dots, p_L\}$, generating the sensor readings $\mathbf{M} = [\mathbf{m}_1 \ \mathbf{m}_2 \ \dots \ \mathbf{m}_k]^T$. An exploring tree strategy will be designed to search the entire monitoring environment, which will determine the best sensor node deployment locations that will minimize the infinite-horizon cost and also make the matrix \mathbf{M} contain the maximum information without any redundancy (i.e. no linearly dependent rows or columns in the matrix; it has the full rank).

1.5 Thesis Organisation

The present chapter (Chapter 1) of the thesis presented an introduction and objectives of the research. Next a survey of the pertinent literature was given. The design of the proposed mobile, wireless sensor network platform design for water quality monitoring was outlined. The algorithm for the optimal mobile sensor node deployment strategy based on environment estimation and sensor data correlation was introduced. The associated problem was formulated starting with the system model, which includes the energy consumption for node mobility and data communication. The organization of the rest of the thesis is given below.

Chapter 2 presents the framework of the adaptive water quality monitoring system and the design and development of the mobile sensor nodes for water quality monitoring. Also, field test results and baseline comparison are presented.

Chapter 3 demonstrates and verifies that sensor nodes with linearly dependent data can be removed from the network without sacrificing the monitoring accuracy while improving the system efficiency. This approach, called linear reduction, is used in Chapter 4 to reduce and optimize the number of deployed sensor nodes.

Chapter 4 presents the complete algorithm for sensor node deployment with linear reduction. It employs a rapid-exploring random tree to find the optimal locations for sensor node deployment. Results of the proposed algorithm, called Rapid-exploring Random tree with infinite horizon cost and Linear Reduction (RRLR), are compared with those of the benchmark algorithm called Rapidly-exploring Random Cycles (RRC) [13]. This comparison shows that RRLR provides a significant reduction in the number of sensor nodes while maintaining the same level of estimation error.

Chapter 5 concludes the thesis. It summarizes the main contributions of the research and suggests possible future work.

Chapter 2 : Design of the Water Quality Monitoring System

Advances of information and communication technologies (ICT) in sensing, processing and wireless communication have led to useful progress in the applications of remote environmental monitoring. Wireless sensor networks with capabilities of fast data acquisition and wireless transmission have been designed and deployed to provide quantitative monitoring and evaluation of aquatic environments. Mobility of sensor nodes is required to provide the capability of relocation of the sensor nodes to locations that can best estimate the environment or collect most information. This requirement is important in the present application of water quality monitoring. This chapter present the design details of our water quality monitoring system.

2.1 Design of an Adaptive Water Quality Monitoring System

An adaptive water quality monitoring system should minimize the number of sensor nodes that are used while maximizing the useful information that is collected in the area of interest. Aquatic sensors are expensive, which prohibits their dense deployment. Hence, it is important to use the least number of sensor nodes to collect as much information as possible. To collect most information, the system needs an optimal deployment strategy as well as the sensor mobility. This means sensor nodes should be able to communicate with each other in order to optimize their current deployment and move themselves to suitable locations. To meet this objectives, the architecture of our water quality monitoring system is designed as in Figure 2.1.

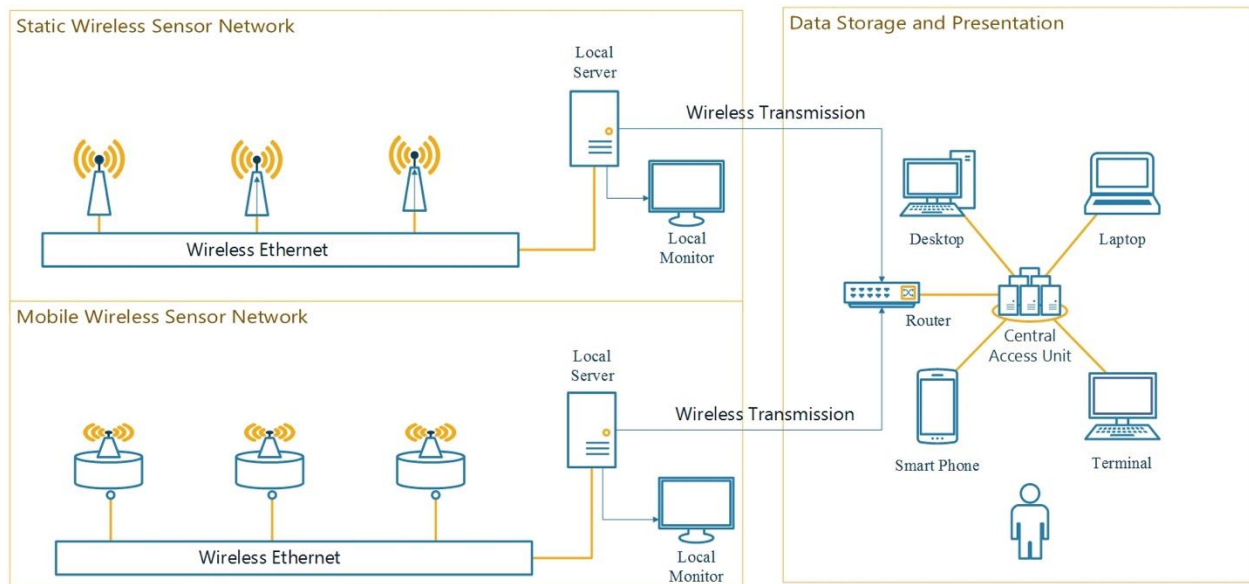


Figure 2.1 System Architecture of the ICT Platform.

As Figure 2.1 indicates, the proposed system is able to monitoring large aquatic areas using wireless communication technologies. Mobile sensor nodes are able to collect sensory data and then transmit them to a base station via radio or Wi-Fi to carry out data preprocessing, which may lead to abnormal value alerts, redundant/failure value removal and data compression. Then GPRS or Ethernet can be used to transmitting stored data to the Central Assessment Unit (CAU). Collected data can then be used for early warning generation and decision making in CAU. Besides, the users can have access to the visualized data easily through a web based Graphical User Interface (GUI), as shown in Figure 2.2.

On-line readings: Sensor Node #1

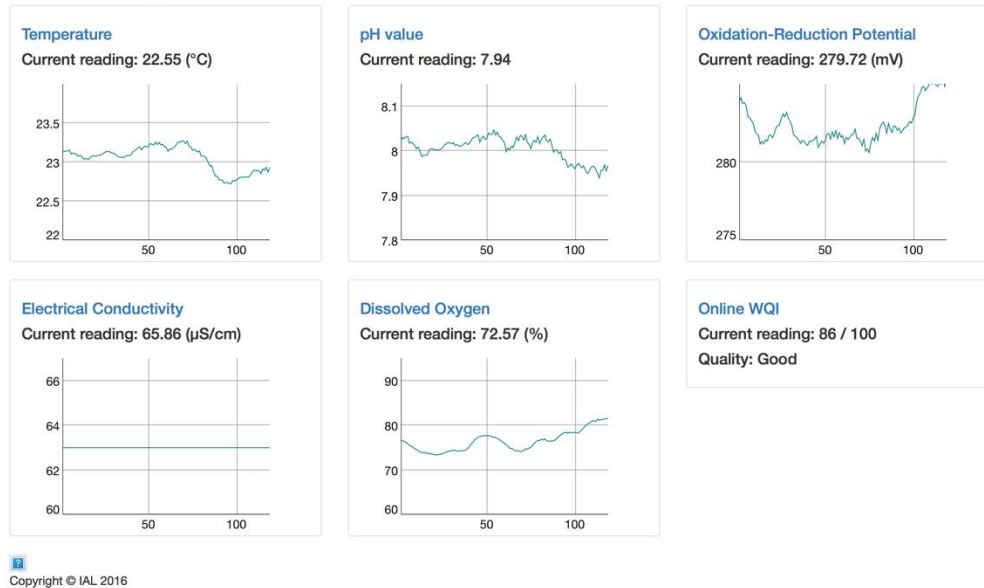


Figure 2.2: Graphical User Interface for Water Quality Monitoring System.

It is seen that five curves are displayed on the GUI, representing temperature, pH value, conductivity, oxidation-reduction potential (ORP), and dissolved oxygen (DO), along with online calculated water quality index (WQI) value. Using this web-based GUI, users can easily find out the historical readings of sampled water quality data.

This system can help technicians to access sensor data directly if they are carrying out field testing in remote areas without access to the Internet. This feature is essential because when carrying out field tests in some rural area, base stations are not able to connect to the Internet, which means the technicians will not get real time sensor readings from the web-based GUI. The original solution provided by Libelum [55] is to send the collected sensor data from sensor nodes to a local gateway, and then, forward these messages to a local server. This method is not efficient and the gateway limits the bandwidth of the transmission within the network. Also, it requires a

steady power supply, which is not easy when deploying in rural areas. In our system, we developed a GUI based on local area network for the transmission of data, which represent sensor readings and real time water quality index (WQI). Real time data transmission and representation system is designed as in Figure 2.3. Figure 2.3 (a) represents the framework provided by a sensor supplier, which requires an extra gateway for data transmission. Figure 2.3 (b) indicates the proposed system framework that skips the gateway. It transmits data from the sensor nodes directly to the local server. In this enhanced framework, mobile sensor nodes use Rapidly-exploring Random Cycles to continuously collect sensor readings and transmit them to the processor for data transmission and processing via a serial port. Then, the microcontroller generates a local wireless network that allows a technician to access it using local servers, where the local GUI is running. After successful connection, localized GUI are able to plot real time data values from the sensor nodes.

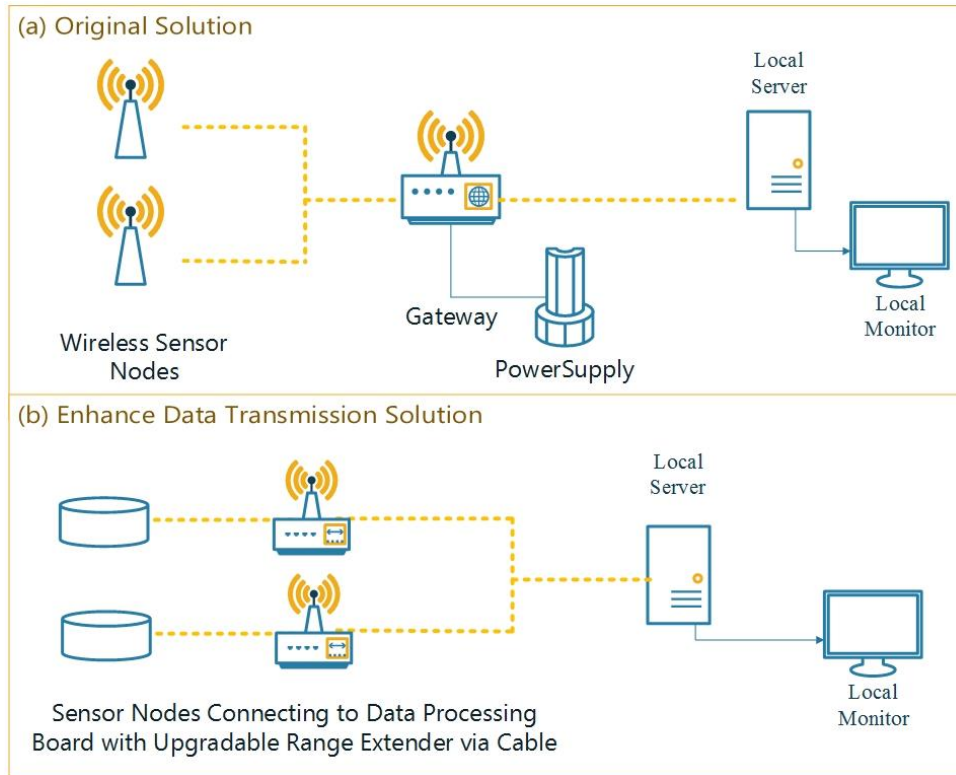


Figure 2.3: (a) Data Transmission Solution From Libelium; (b) Enhanced Data Transmission Framework.

The capability of data transmission is enhanced in the developed system. However, the packet loss rate is found to be high and the transmission distance is short, according to our tests. This is because the sensor node (Wasp mote) that was purchased from Libelium [55] uses a low performance processor and Wi-Fi board. The solution from sensor node provider, Libelium, requires the data collected from the sensor nodes to be first transmitted to a gateway and then forwarded to local servers, as shown in Figure 2.3 (a). However, the gateway limits the performance of data acquisition in the water quality monitoring system and it is somewhat difficult to deploy it in a rural area. The data on route to the local server is first stored in the gateway, which introduces considerable delay thereby hampering real-time data acquisition. Besides, the system bandwidth is not adequate for receiving data from the sensor nodes and forwarding to local server

at the same time. Moreover, the power supply is a key challenge. In particular, technicians will have to carry an Uninterruptible Power Supply (UPS) as an energy supply to provide adequate voltage that will enable booting the gateway. To get rid of limitations of the gateway, a new data transmission framework will be designed to both increase the system reliability and save labor. As shown in Figure 2.3 (b) and Figure 2.4, raspberry pi 3 microcontroller is chosen as the data processor and transmitter, which has a CPU with four ARM Cortex-A53 at 1.2GHz, and EDUP USB Wi-Fi extender [56] as the data transmitter, which is an industry leading and affordable USB wireless adapter. This data transmitting framework can then transmit data directly from a sensor node to the local server. Another advantage of using a commercial wireless communication board is that the power and the bandwidth can be easily increased by switching to higher end ones. Also, the Transmission Control Protocol (TCP) is used to ensure reliable and error checked data transmission [57]. Its three-way TCP handshake establishes stable connection between data transmitting server and client and thus addresses the problem of packet loss. By doing so, the packet loss during transmitting sensor readings from the sensor nodes to the local server is minimized. According to the tests, there will be very little packet loss within a distance of 50 meters (without obstacles). Further comparison of the two data transmitting schemes will be made in section 2.3 Sensor Node Deployment and Field Testing.

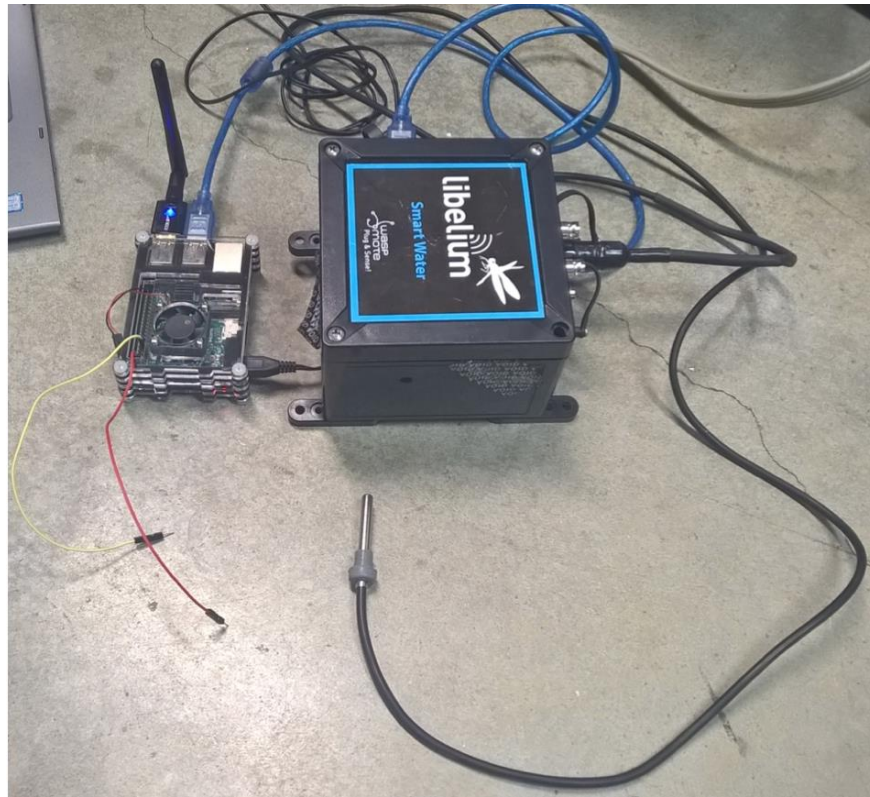


Figure 2.4: Enhanced Sensor Node with Wireless Network Extender.

The GUI at the local server is shown in Figure 2.5. All five sensor readings will be plotted. Thus, technicians who work in a remote rural area and who may not have access of the Internet, are able to easily monitor the current and the past water quality in the area that they are interested in. Sometimes, technicians may only be interested in the sensor reading after they arrived to help them study on-going trends of different water quality attributes. In this way, we also place a reset button on the GUI that enables the user to monitor sensor readings and WQI curve after they arrive the area of interest.

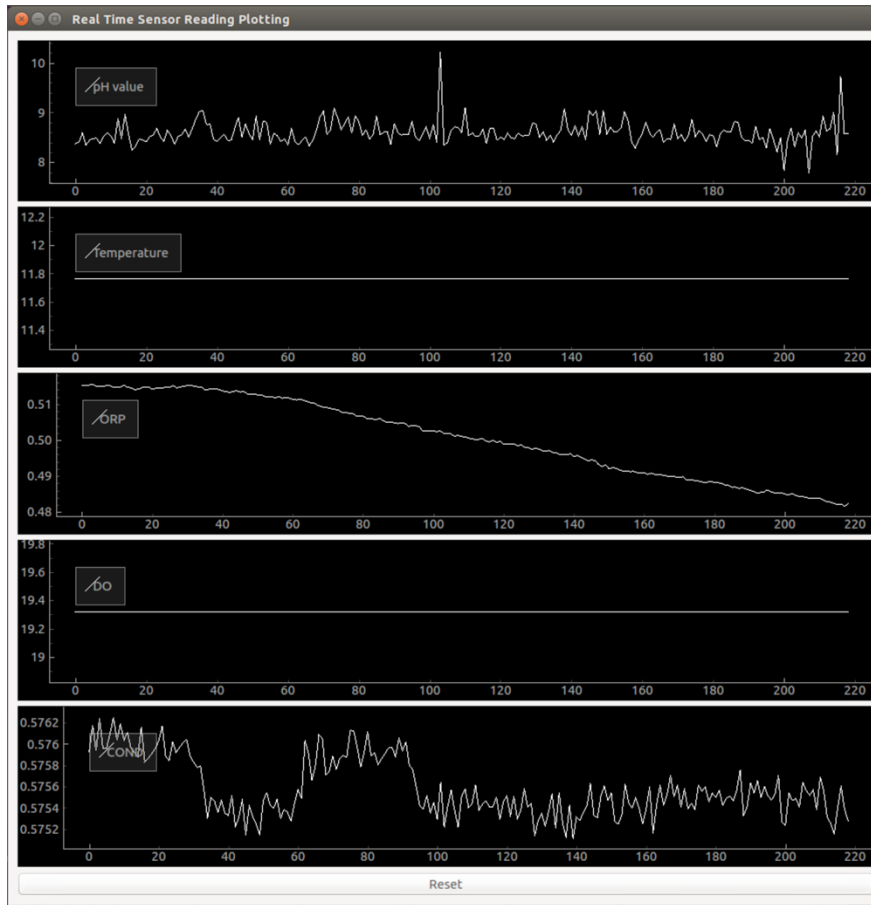


Figure 2.5: Local Network Based GUI.

2.2 Design of Mobile Water Quality Monitoring Sensor Node

The purpose of the mobile sensor node is to propel itself to any desired location in an aquatic environment and acquire information at that location using its sensors. Technicians can use a remote controller (e.g., a joystick) to drive the mobile sensor node directly or the node may move autonomously according to some algorithm. The information/data acquired by the mobile sensor nodes may be analyzed to determine the best deployment locations [13] that would generate most useful information [9] [58]. An algorithm may command a node to follow a certain path and stop at several points in this path for short periods to acquire environmental data. Then, an optimal planning strategy can be determined by studying the spatial and temporal relationships in the

collected data. Commonly, mobile sensor nodes are controlled manually. Controlling a mobile sensor node through the remote controller is a convenient way for technicians to operate the node for it to move to any desired location quickly, but such control is costly and may not be accurate. One technician is only able to control one sensor node at a time and has to control it continuously. Besides, humans have limited sight, which restricts the operation range if the sensor node is controlled by sight, directly through the remote controller. Hence, in the developed system, both automatic control and environmental information computation are located on board in the sensor node, which can automatically navigate the sensor node to acquire information. The generated data includes the GPS location and the water quality attributes at that location.

The mobile sensor nodes are operated as in Figure 2.6. In the beginning, the GPS location and the mobile node inertia are captured by the sensors and used as system input. The processed sensory data for a specific period is treated as the system feedback. The position controller processes both spatial and temporal information and makes a decision on the path that the mobile node should follow to acquire the environmental information. The position and velocity control can be used to move the node by controlling the propellers. The information is processed by the on-board microcontroller for driving the node to a desired location.

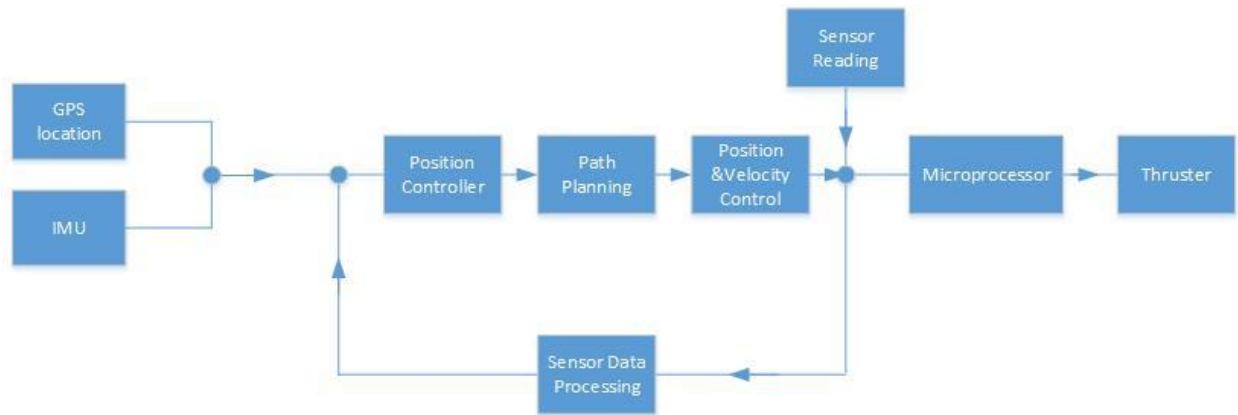


Figure 2.6: Mobile Sensor Node Operation.

Based on the mentioned requirements for a sensor node, its components are chosen. For position and velocity control, one 3DR Pixhawk Mini [59] is chosen, which is a well-known programmable autopilot controller for unmanned vehicles. It includes an advanced 32-bit ARM Cortex M4 Processor, 8 servo outputs and the capability to connect to a GPS module and a dual inertial measurement unit (IMU). Chosen sensors for a node include the GPS module [59], which is compatible with Pixhawk mini that could receive up to 3 GNSS (GPS, Galileo, GLONASS, BeiDou) with industry-leading 167 dBm navigation sensitivity. Also, Libelium Waspote [55] is chosen as the aquatic environment sensor, since it has an open source SDK with ultra-low power consumption ($7 \mu\text{A}$), supports up to 110 different types of sensor and equipped with 16 different radio technologies. Finally, as actuators, two T100 thruster and their electronical servo controller (ESC) from BlueRobotics [20] are chosen, each of them can produce over 5.2 lbf forward thrust and 4.1 lbf reverse thrust.

The designed sensor node is shown as in Figure 2.7, which consists of two parts. The first part is the motion control module, as shown in Figure 2.8 where a four-cell battery is the power source for 3DR Pixhawk Mini and ESC. The 3DR Pixhawk Mini is the position and velocity

controller, which connects to the controller Wi-Fi/Radio module to communicate with the remote control server for the technicians, the GPS module to receive the location information, and ESC of the propellers to direct the movement of the sensor node. The second part is the data transmission module, as shown in Figure 2.4, where the data transmission microcontroller Raspberry Pi is powered by one of two ESCs through a jumper cable and connects to the aquatic information collection sensors and the data transmission Wi-Fi module.



Figure 2.7: Developed Mobile Sensor Node.

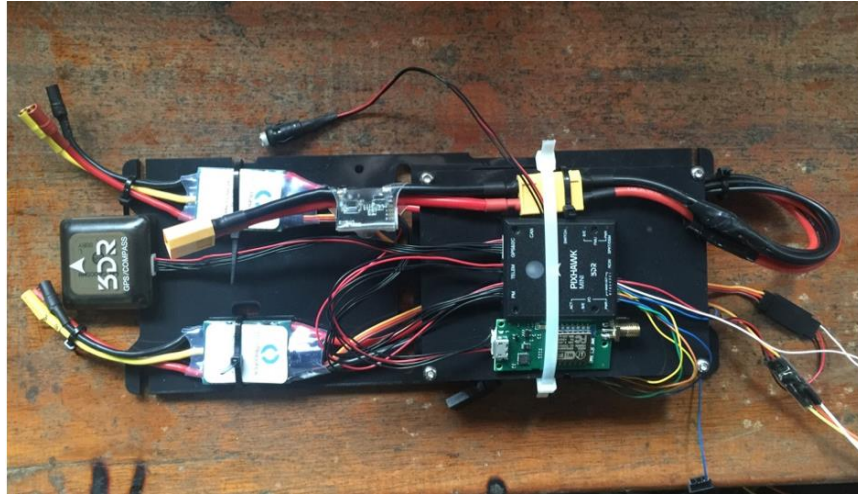


Figure 2.8: Controller of the Mobile Sensor Node.

An operating requirement determines proper control of the mobile sensor node. The plant can only be deployed in an outdoor aquatic environment without strong waves, since electrical components are not water proof and the GPS cannot function accurately inside buildings. Besides, it is assumed that there are no obstacles in the aquatic environment since the node is not equipped with obstacle avoidance devices such as camera or ultrasonic distance sensor and associated schemes. Also, the battery life is assumed to be adequately long for a mobile node to execute its tasks.

2.3 Sensor Node Deployment and Field Testing

First, the rate of packet loss is tested for both data transmitting schemes. Two sets of sensor nodes are deployed in the ICICS (Institute of Computing, Information and Cognitive Systems) building, University of British Columbia, Vancouver, Canada for a period of 3 hours to test the performance of the different schemes. They are deployed in the level 1 of the ICICS building as shown in Figure 2.9. A blue dot indicates the location of the sensor node, which has a Raspberry Pi microcontroller as the data transmitter; an orange dot indicates the location of a sensor node that directly connects

to the gateway; a red node denotes the location of the local server and the gateway. Test results show that the packet loss rates are 0.85% and 2.56% for the enhanced sensor node and the original one, respectively. Also, data collected from the two nodes has a large gap. During the three-hour data collection operation, both sensor nodes are set to collect data at full speed. However, the enhanced sensor node collects 1651 reading samples (each sample includes a reading for all five water quality attributes), while the original node only collects 468 samples. Moreover, the original node cannot connect to the server when it is placed in room 061D during the test, and the data collection is initiated when it is placed in room 065. Also, data forwarding is not done from the gateway to the local server, because the performance of the original sensor could be worse since it would use a two hop wireless sensor network.



Figure 2.9: Sensor Node, Local Server and Gateway Deployment in the ICICS Building.

After the water quality monitoring system is developed, field testing has been carried out with it in both Canada and India. In the first stage, static sensor nodes are deployed in Yosef Wosk reflecting pool at UBC. The deployment is shown in Figure 2.10, where five static sensors are spread over the entire pool.

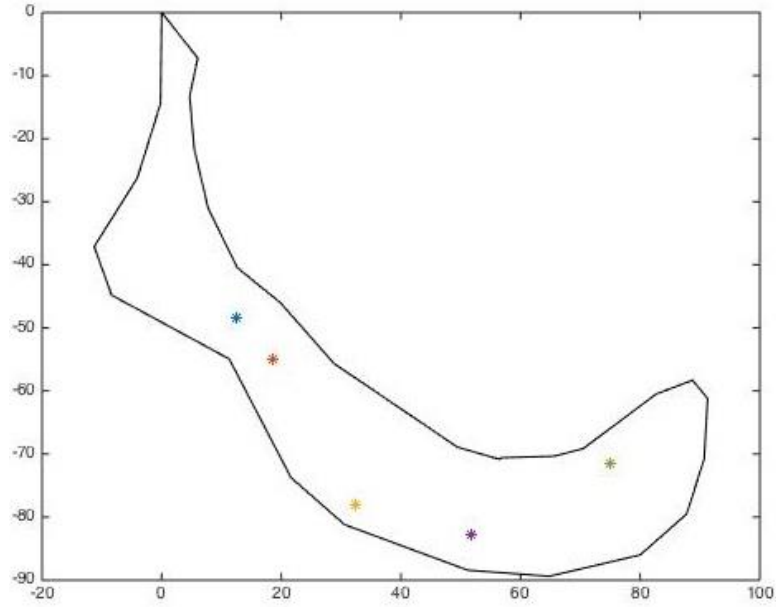


Figure 2.10: Deployment of Sensor Nodes in UBC.

Interval of sampling is about 5 seconds for the first stage deployment in UBC, when 306 data samples have been collected in total. They are plotted as in Figure 2.11.

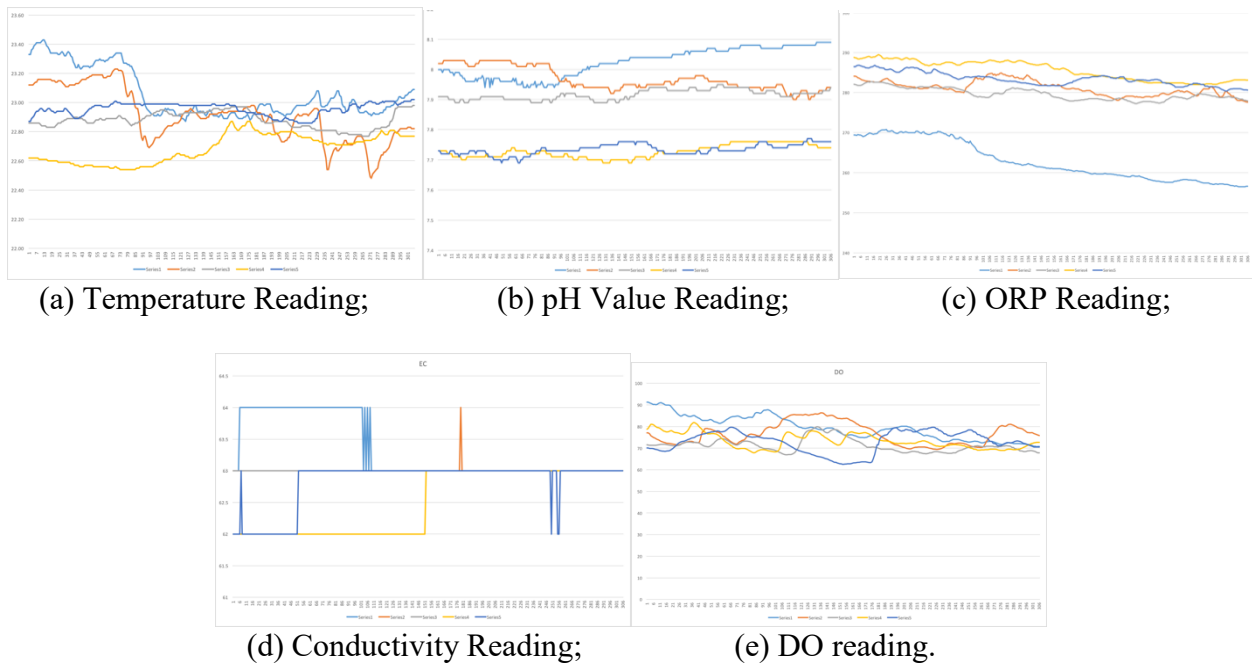


Figure 2.11: Sensor Readings of the Field Test at UBC.

It is seen that in view of the small aquatic environment, the sensor readings are rather similar when the node locations are close. This indicates the existence of some redundancy if the sensor nodes are closely deployed. In view of this, a Linear Reduction method is proposed in Chapter 3 to reduce the linear dependency within the sensor network.

Furthermore, we also carried out field test in Ulhas River, Neral, India. The location is shown in Figure 2.12,. The dynamic system of sensor node and control flow were tested at that location. Results show that the mobile sensor node is able to move against the water flow using only 2 propellers, and a lightweight Wi-Fi-based communication board is able to control the mobile sensor node within 100 meters.



Figure 2.12: Location of Field Testing in Ulhas River, Neral, India.

2.4 Conclusion

This chapter presented the design of the overall platform for water quality monitoring, and some results from field testing using the platform. A practical platform for water quality monitoring was

developed and tested to determine the ability of the sensor node to move against strong water flow, and data transmission under obstacles. The cost of the platform was low, and the node had easy to upgrade components. According to the test results, the sensing platform has shown good performance under different conditions like moving in an aquatic environment with strong water flow and transmitting data in an indoor environment with many obstacles and moving humans.

Chapter 3 : Linearly Redundant Sensor Reduction

This chapter presents the Linear Reduction (LR) algorithm. It uses the correlated neighbor detection strategy to determine the linearly dependent sensor locations, for optimal deployment of sensor nodes within the AOI.

3.1 Linearly Dependent Neighbor Clustering

Given a set of sensor nodes that is deployed in a geographical AOI, $\mathcal{N} = \{n_1, n_2, \dots, n_k\}$ and their readings at a specific time $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k\}$, where $\mathbf{m}_i \in \mathbb{R}^{M \times 1}$ and M is the length of this sensor reading. The readings of this k nodes are placed in a matrix $\mathbf{M} \in \mathbb{R}^{k \times M}$ as $\mathbf{M} = [\mathbf{m}_1 \ \mathbf{m}_2 \ \dots \ \mathbf{m}_k]^T$. If some readings among different \mathbf{m}_i are linearly correlated to others, they are linearly redundant columns (or rows), i.e., \mathbf{M} does not have the full rank, and they can be expressed as a linear combination of some other rows:

$$\mathbf{m}_i = \sum k_j \cdot \mathbf{m}_j, (k_j \neq 0 \text{ and } \mathbf{m}_i, \mathbf{m}_j \in \mathcal{M}) \quad (3-1)$$

Thus, the linear dependence relationship of nodes indicates the presence of redundancy of the node deployment. Any non-zero \mathbf{m}_j with the associated non-zero k_j is a correlated neighbor of \mathbf{m}_i . Then, \mathbf{m}_i together with its correlated neighbors composes a linearly correlated set. If several nodes are linearly dependent, some of these nodes (called redundant nodes) can be inferred from the remaining nodes. As a precursor to the algorithm, a way to detect linearly correlated clusters for a given matrix \mathbf{A} is presented first.

Theorem 1. *Zero rows in the result of Gaussian Elimination represent the cluster of linearly dependent rows. The corresponding rows in the original matrix are redundant and can be removed.*

Proof. Consider a $K \times M$ matrix \mathbf{A} , where K (rows) denotes the number of sensor nodes and M (columns) is the number of readings generated at each node. Assume that the rank of matrix \mathbf{A} is r . If $r < \min(K, M)$, there are linearly dependent rows in this matrix, and they can be inferred from other rows.

Matrix \mathbf{A} may be expressed as:

$$\mathbf{A}_{K \times M} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_K \end{bmatrix} = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \vdots \\ \mathbf{m}_K^T \end{bmatrix} \quad (3-2)$$

After the first step in the Gaussian elimination, each row will be of the form:

$$\mathbf{a}_p = \mathbf{a}_p - \frac{A_{p,i}}{A_{i,i}} \mathbf{a}_i \quad (A_{i,i} \neq 0, i \neq p) \quad (3-3)$$

After several steps of row addition in the Gaussian elimination, each row in the final result of the row reduction matrix \mathbf{B} , may be expressed as

$$\mathbf{b}_i = \sum_{j=1}^K k_j \cdot \mathbf{a}_j, (k_k = 1) \quad (3-4)$$

where k_j is the linear coefficient. This linear combination is produced by Eqn. (3-3).

In the normal matrix representation, if matrix \mathbf{A} does not have the full rank, then zero rows exist in \mathbf{B} . For these rows, we have:

$$\mathbf{b}_k = \mathbf{0} \quad (3-5)$$

Combining (3-4) and (3-5) we get:

$$\sum_{j=1}^K k_j \cdot \mathbf{a}_j = \mathbf{0}, (k_k = 1) \quad (3-6)$$

This can be further expressed by q non-zero linearly correlated rows, say \mathbf{a}_{l_i} , where l_i is the index of these correlated rows in the matrix \mathbf{A} and i ranges from 1 to q :

$$\mathbf{a}_k = \sum_{j=1}^q k'_{l_i} \cdot \mathbf{a}_{l_i} \quad (k'_{l_i} \neq 0, k \neq l_i, 1 \leq q \leq K) \quad (3-7)$$

It follows that row \mathbf{a}_k in the matrix \mathbf{A} can be represented by other rows. Hence it is redundant and can be removed.

■

According to Theorem 1, from the information of the row addition process in Gaussian Elimination, we are able to cluster linearly correlated rows and linearly correlated sets of the matrix. Linear Reduction is applied to group the linearly redundant nodes as clusters by calculating their linear correlation.

As shown in Algorithm 1, the inputs of algorithm LR are the data matrix \mathbf{M} , its dimensions M and K , error bound ε , and the manipulation matrix Θ , which records the row addition actions while carrying out Gaussian elimination. Algorithm LR initializes Θ to a diagonal matrix with all elements equal to 1 in Line 3. This means i th row is present by itself in matrix \mathbf{M} . Line 4 indicates that LR is about to carry out row reduction, column by column, as in Gaussian elimination. Line 5 identifies the unselected row, which is not used for row reduction, with the largest element in the current reducing column. This improves the stability of the algorithm [60] and avoids adding the same row multiple times. Line 13 carries out row addition for removing row redundancy while line 14 records this action in Θ . After the iteration ends, if the input matrix \mathbf{M} has row redundancy, there should be zero rows in the output matrix \mathbf{M} . The number of zero rows should be equal to the row dimension M minus the rank of the input \mathbf{M} . Next, extract the rows Θ_i in matrix Θ , where the corresponding location in the output matrix \mathbf{M} is $\mathbf{0}$. Then, in each row Θ_i , the column indices of non-zero elements represent linearly correlated rows of \mathbf{M} , which form the linearly correlated set, and their values are the linear correlation coefficients, which satisfy Eqn. (3-5).

Input: $\mathbf{M}, \Theta, M, k, \epsilon$
Output: \mathbf{M}, Θ
 Initialization;
while $i \leq M$ **do**
 find unselected \vec{m}_p^\top with largest \mathbf{M}_{pi} ;
 if $\mathbf{M}_{.i}$ is a zero column within ϵ **then**
 | $i ++$;
 else
 set row \vec{m}_p^\top as selected;
 normalize \vec{m}_p^\top to $[-1,1]$ as \vec{m}'_p^\top ;
 foreach rows j *except* \vec{m}_p^\top **do**
 | **if** $\mathbf{M}_{ji} \geq \epsilon$ **then**
 | $\vec{m}_j^\top -= \mathbf{M}_{ji} \cdot \vec{m}'_p^\top$;
 | $\Theta_{j.} -= \mathbf{M}_{ji} \cdot \Theta_{p.}$;
 | **end**
 end
 $i ++$;
 ϵ Remove;
 end
end

Algorithm 1: Linearly Dependent Neighbor Cluster Detection by Linear Reduction.

Once the correlated set of rows (sensor nodes) is identified in this manner, the corresponding redundant nodes can be relocated to other uncovered places, based on their readings. It is important to note that the removal of any nodes from a correlating cluster will keep other clusters correlated, regardless of whether the removed row/node is part of another correlating cluster, as clear from Theorem 2.

Theorem 2. *Removal of any nodes in a correlating cluster will not affect other correlating clusters and will not result in information loss, even if the removed node is part of another correlating cluster.*

Proof. Suppose that there are m nodes, and that q rows could be inferred from other rows, where $q < m$. Then q_j th rows are zero at the end of LR ($1 \leq j \leq q$). This means any q_j th row can be inferred from some other rows. Similar to (3-6), we have:

$$\mathbf{a}_{q_j} = \sum_{\mathbf{a}_j \in I} k_j \cdot \mathbf{a}_j \quad (3-8)$$

where I is a set containing all possible rows. One node p that is used for inferring node q_j will be removed as it is a linearly redundant node, where \mathbf{a}_p corresponds to the left side of Eqn. (3-7).

If \mathbf{a}_p does not belong to any other correlation set, it can be just swapped with \mathbf{a}_{q_j} :

$$\begin{aligned} \mathbf{a}_{q_j} &= \sum_{\mathbf{a}_j \in I, i \neq p} k_j \cdot \mathbf{a}_j + k_p \cdot \mathbf{a}_p \\ \mathbf{a}_p &= - \sum_{\mathbf{a}_j \in I, i \neq p} \frac{k_j}{k_p} \cdot \mathbf{a}_j + \frac{1}{k_p} \cdot \mathbf{a}_{q_j} \\ \mathbf{a}_p &= \sum_{\mathbf{a}_j \in I'} k'_i \cdot \mathbf{a}_j, I' = (I \setminus \{\mathbf{a}_p\}) \cup \{\mathbf{a}_{q_j}\} \end{aligned}$$

Thus, according to Theorem 1, \mathbf{a}_p is redundant and can be removed while no information will be lost. Besides, since \mathbf{a}_p does not belong to any other correlation set, other sets will not be affected.

Moreover, if \mathbf{a}_p is in another correlation set I'' , we have:

$$\mathbf{a}_p = \sum_{\mathbf{a}_i \in I''} n_i \cdot \mathbf{a}_i \quad (3-9)$$

$$\mathbf{a}_{q_j} = \sum_{\mathbf{a}_j \in I, i \neq p} k_j \cdot \mathbf{a}_j + k_p \cdot \mathbf{a}_p \quad (3-10)$$

Combining (3-9) and (3-10) we have:

$$\begin{aligned} \mathbf{a}_{q_j} &= \sum_{\mathbf{a}_j \in I, i \neq p} k_j \cdot \mathbf{a}_j + k_p \cdot \sum_{\mathbf{a}_i \in I''} n_i \cdot \mathbf{a}_i \\ \mathbf{a}_{q_j} &= \sum_{\mathbf{a}_j \in I, i \neq p} k_j \cdot \mathbf{a}_j + \sum_{\mathbf{a}_i \in I''} k_p n_i \cdot \mathbf{a}_i \\ \mathbf{a}_{q_j} &= \sum_{\mathbf{a}_j \in I'''} k'_i \cdot \mathbf{a}_j, I''' = I \cup I'' \setminus \{\mathbf{a}_p\} \end{aligned} \quad (3-11)$$

Thus, information in both sets I and I'' will not be lost. ■

Corollary 1. *Removing single linearly dependent row from the matrix will not result in information loss until the smaller dimension of the matrix has been reduced to its rank.*

Proof. According to Theorem 2, linearly redundant rows can be removed without affecting the linear dependence in other linearly dependent sets. Thus, once there are linearly dependent rows in the matrix, as detected by LR, the selected rows in the linearly correlating sets can be removed iteratively until the matrix become full rank. ■

We have proved that the removal of a single linearly dependent row within a linearly correlating set will not result in information lost. Another observation is that if the average data value in one sampling period M exceeds the early warning threshold in the removed rows during real-time water quality monitoring, it should be represented by the rows that infer it.

Theorem 3. *When the average sensor reading of a redundant node n_p over the sampling period M achieves the threshold value ρ , there exists at least one other node, n_j , which can detect this*

situation (without knowing sensor reading \mathbf{m}_p of sensor node n_p). Specifically, under this condition, there exists a sensor node n_j , whose average data value over the same sampling period M will be $\geq \frac{\rho}{k_j}$ where k_j is the linear dependent coefficient of sensor node n_j . Then, the sensor node n_j will provide the early warning.

Proof. Assume the redundant node to be relocated as $n_p \in \mathcal{N}$ and its value is $\mathbf{m}_p \in \mathcal{M}$. Then, it can be represented by the data from the other nodes as:

$$\mathbf{m}_p = \sum_{\mathbf{m}_i \in \mathcal{M}} k_i \cdot \mathbf{m}_i + \epsilon, (i \neq p, k_i \neq 0) \quad (3-12)$$

where \mathbf{m}_i is the reading for node n_i , and k_i is its linear dependence coefficient.

The number of \mathbf{m}_i in Eqn. (3-12) is denoted as K , which is the total number of sensor nodes in the field, and ϵ is the error variable, which mainly represents noise.

Let $avg(\mathbf{m}_i)$ denote the average value of one data vector \mathbf{m}_i , specifically,

$$avg(\mathbf{m}_i) = \frac{sum(\mathbf{m}_i)}{M}.$$

Then, we can formulate Theorem 3 as: $\exists sum(\mathbf{m}_j) \geq \frac{\rho}{k_j}$, if $avg(\mathbf{m}_p) \geq \rho$, where $j \neq p$, and k_j is the linear dependence coefficient of sensor node n_j for the sensor node n_p . Thus,

$$avg(\mathbf{m}_p) = avg\left(\sum_{\mathbf{m}_i \in \mathcal{M}} k_i \cdot \mathbf{m}_i\right) + \epsilon'$$

$$avg(\mathbf{m}_p) = \sum_{\mathbf{m}_i \in \mathcal{M}} avg(k_i \cdot \mathbf{m}_i) + \epsilon'$$

$$\Rightarrow \sum_{\mathbf{m}_i \in \mathcal{M}} k_i \cdot \frac{sum(\mathbf{m}_i)}{M} \geq \rho$$

Then, there must exist at least one $k_j \cdot \frac{sum(\mathbf{m}_j)}{M}$ that is larger than $\frac{\rho}{K}$, because otherwise the above condition will not be satisfied. Then:

$$\begin{aligned} \exists j, k_j \cdot \frac{sum(\mathbf{m}_j)}{M} &\geq \frac{\rho}{K}, \\ \Rightarrow \exists j, k_j \cdot sum(\mathbf{m}_j) &\geq \frac{\rho \cdot M}{K} \end{aligned}$$

Given that the total number of nodes in the field, K , must not be greater than the data sampling period M as shown in Lemma 1 below, we have:

$$sum(\mathbf{m}_j) \geq \frac{\rho}{k_j}$$

Thus,

$$\exists sum(\mathbf{m}_j) \geq \frac{\rho}{k_j}, \text{ if } avg(\mathbf{m}_p) \geq \rho, \text{ where } j \neq p.$$

■

With the help of this theorem, it is easy for us to find out whether there is a threshold event in the field. Each node can store the information of the cluster it belongs to and its linear dependence coefficient. Then, on reading sufficient data over a sampling period, a simple calculation can lead to the detection of an extreme event at the location of sensor node n_p .

Lemma 1. Suppose that a data matrix has K node readings, expressed as $\mathbf{M} = [X_{.1}, X_{.2}, \dots, X_{.K}]$, where $\mathbf{M} \in \mathbb{R}^{M \times K}$, and the row rank is equal to the column rank [61] [62]. Then, if $K > M$, which means the data sampling period is less than the number of nodes to be analyzed, which is infeasible. Thus, we must have $M \geq K$.

In the present analysis it is assumed that each node has only one sensor. The case of multiple sensors in a node will be considered in future work.

3.2 Conclusion

This chapter presented the Linear Reduction method, which is able to determine linearly dependent sensor data, which in turn can detect redundant sensor nodes. It was shown that linearly dependent data vectors within the linear correlation set could be removed until the sampling data matrix was full rank. By doing so, the redundancy among sensor node locations could be removed. Based on the theorems presented and proved in this chapter, a sensor placement strategy can be developed which avoids redundant sensor node deployment locations thereby reducing the cost while not sacrificing the acquired information. Algorithms proposed in this chapter will be utilized in the following chapter for optimizing the sensor node deployment for monitoring the environment.

Chapter 4 : Optimal Sensor Node Deployment with Linear Reduction

In water quality monitoring, a group of mobile sensor nodes are deployed to explore a geographical area of interest (AOI) and acquire key water quality parameters at various locations and time. Since the number of mobile sensor nodes is limited, moving them to the best locations to acquire most information with least error (e.g., least estimation error) is a key objective. To determine the best sensor placement locations, a discrete number of mobile sensor nodes needs to search a mathematically infinite region subject to some optimizing criteria [63]. Sampling-based path planning for sensor scheduling over an infinite horizon may be used to achieve this objective [64] [65]. For this purpose, an optimal sensor node deployment strategy with linear reduction may be proposed as in Figure 4.1.

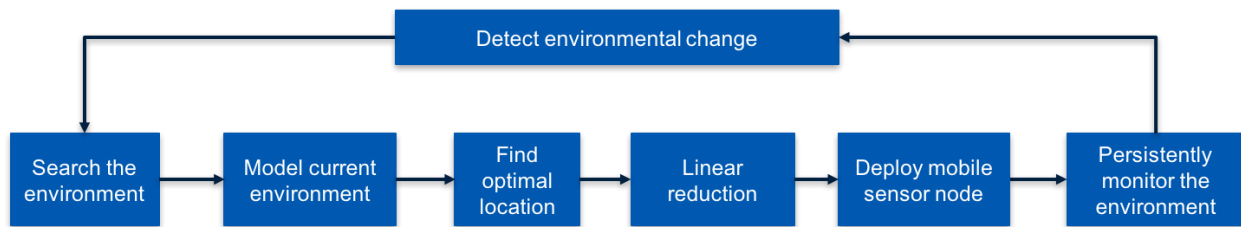


Figure 4.1: Framework of Optimal Sensor Node Deployment with Linear Reduction.

First, such technologies as satellite imaging, mobile sensor nodes or static sensor network may be utilized for monitoring and then modeling the current aquatic environment. Then, based on this model, algorithms may be developed to establish the optimal sensor node deployment locations that meet certain optimizing criteria like receiving most information or achieving least estimation error. After analysing the sensor reading at these locations based on the environment model, linear reduction may be applied to determine the redundant sensor nodes. After optimal sensor node deployment locations are determined, the sensor nodes will be deployed to persistently monitor the aquatic environment. Once the collected data is analyzed to fully estimate the current

environment with respect to the current model or if a change in the aquatic environment is detected, the above process has to be repeated to generate a new model that satisfies the new environment.

In the present work, a common model that is used in geological and environmental studies, which is generated from a fixed number of spatial basis function is used for modeling the environment. It has been proved that this model is a linear-Gaussian dynamic system, which presents both spatial and temporal correlation in the geographical AOI [13]. In water quality monitoring, passive sensors are used to monitoring the water quality. The mobile sensor nodes will capture the water quality information plus noise (assumed to be Gaussian white) noise at their deployed locations. Then, infinite horizon cost is used as an evaluation criterion for this problem. By minimizing the infinite horizon cost, optimal sensor deployment locations can be determined [63]. The linear Kalman filter, which minimizes the mean square estimation error (MSE) for a linear system is used to optimize the sensor node deployment [13] [45].

4.1 Modeling of Environment

Modeling of the aquatic environment within the geographical AOI is presented now. A common environmental model that is used for geological and environmental systems [66] is chosen for this purpose. The environment model can be represented as the output value at a desired location and at a given time. It is a function of the temporal information matrix A , spatial information matrix (x, y) , location (x, y) in the field and the time t , and may be expressed as $v_t(x, y) = F(A, C(x, y), x, y, t)$. The output value in the field $v_t(x, y)$ may be decomposed as the dot product of the spatial information matrix $C(x, y) = [c_1(x, y) \quad \dots \quad c_n(x, y)]$, which is in fact a static row vector consisting of n spatial basis functions, and the time varying coefficient vector $a_t =$

$[a_t^1 \dots a_t^n]^T$. Each basis function is represented by a Gaussian basis function as $c_i(x, y) = K e^{-\frac{\|(x,y)-(x_i,y_i)\|^2}{2\sigma_c^2}}$, where (x_i, y_i) represents the location of the i th basis center. The time-varying coefficient vector can be represented as the state of a linear discrete-time system:

$$\mathbf{a}_{t+1} = \mathbf{A}\mathbf{a}_t + \boldsymbol{\omega}_t$$

where $\boldsymbol{\omega}_t \sim N(0, \mathbf{Q})$ is Gaussian white noise and \mathbf{Q} is the covariance matrix, which is positive semidefinite.

It follows that the aquatic model output can be represented as

$$v_t(x, y) = \mathbf{C}(x, y)\mathbf{a}_t + \mathbf{v}_t$$

where $\mathbf{v}_t \sim N(0, \mathbf{R})$ is Gaussian white noise and \mathbf{R} is the covariance matrix, which is positive semidefinite. In this manner, both spatial and temporal information is captured. Matrices \mathbf{A} and \mathbf{Q} are learned from the data set that was collected during our field testing at Yosef Wosk Reflecting Pool, UBC, Vancouver, Canada, using the subspace estimation method for a state space model [67]. To illustrate this environment model, the estimated environment at times 1, 100, 200, 300 is plotted as in Figure 4.2. the z-axis represents the estimated temperature. These figures are generated by using $t_1 = 1$ as the initial time, \mathbf{a}_1 as the initial state vector, and by calculating $\mathbf{C}(x, y)$. Thus the state-space output $v_t(x, y)$ at each location at any time, can be calculated.

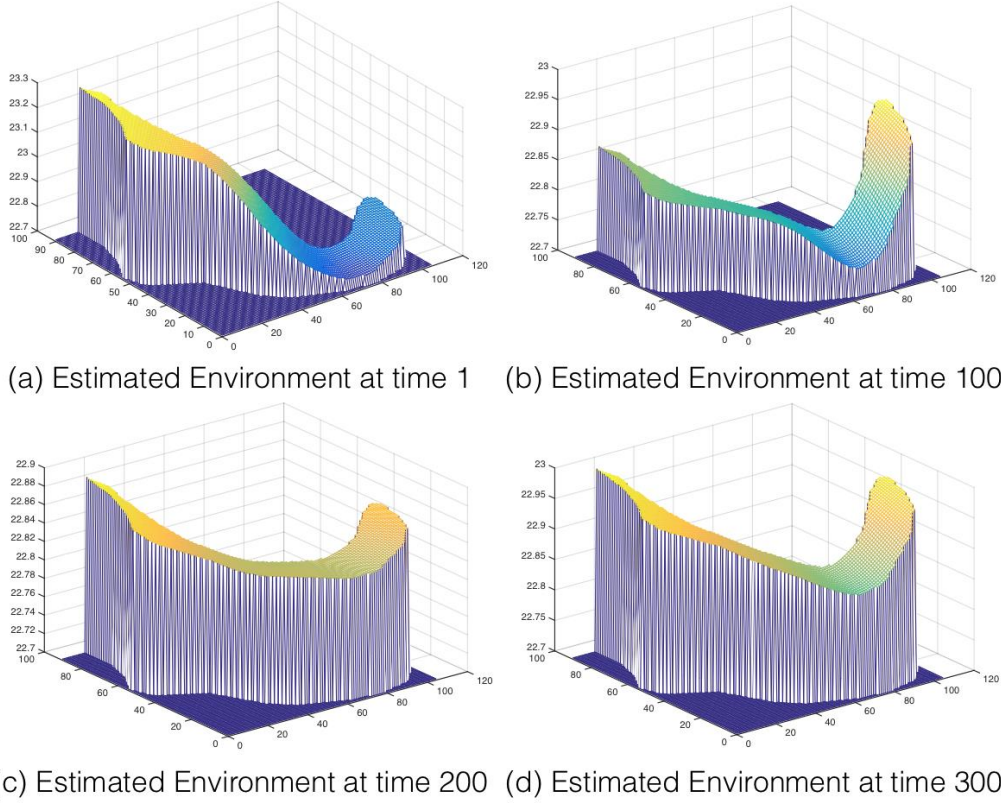


Figure 4.2: Example of Spatiotemporal Field for Different Times.

4.2 Estimation Quality of Deployment Locations

As shown by Lan et al. [13], the linear Kalman filter can be used for estimating the time-varying coefficients \mathbf{a}_t , and the prior error covariance matrix of this estimation can be used as a quality measure for the criteria [13]. Linear Kalman filter is an optimal estimator, which uses state-space models and minimizes the mean square error (MSE) [68]. The Riccati equation of the linear Kalman filter is used for updating the prior error covariance matrix as:

$$\boldsymbol{\Sigma}_{t+1} = \mathbf{A}\boldsymbol{\Sigma}_t\mathbf{A}^T - \mathbf{A}\boldsymbol{\Sigma}_t\mathbf{C}(x_i, y_i)^T \times (\mathbf{C}(x_i, y_i)\boldsymbol{\Sigma}_t\mathbf{C}(x_i, y_i)^T + \mathbf{R})^{-1}\mathbf{C}(x_i, y_i)\boldsymbol{\Sigma}_t\mathbf{A}^T + \mathbf{Q} \quad (4-1)$$

Thus, the maximum eigenvalue of this prior covariance matrix, $\rho(\boldsymbol{\Sigma}_t)$, can be used as the objective function for optimization since the largest eigenvalue represents the worst estimation accuracy in the field, which is more general for evaluating the goodness of estimation over the entire

geographical AOI. Furthermore, to remove the effect of transient high error of estimation in the beginning, the prior error should be evaluated as time goes infinite. Then Eqn. (4-1) becomes:

$$\boldsymbol{\Sigma}_{\infty}^i = \mathbf{A}\boldsymbol{\Sigma}_{\infty}^i\mathbf{A}^T - \mathbf{A}\boldsymbol{\Sigma}_{\infty}^i\mathbf{C}(x_i, y_i)^T \times (\mathbf{C}(x_i, y_i)\boldsymbol{\Sigma}_{\infty}^i\mathbf{C}(x_i, y_i)^T + \mathbf{R})^{-1} \mathbf{C}(x_i, y_i)\boldsymbol{\Sigma}_{\infty}^i\mathbf{A}^T + \mathbf{Q}$$

The minimal largest eigenvalue in each possible deployment strategy is the cost:

$$J(\tilde{\sigma}) = \rho(\boldsymbol{\Sigma}_{\infty}^{\tilde{\sigma}}) = \max_{i=1,2,\dots,K} \rho(\boldsymbol{\Sigma}_{\infty}^i)$$

where $\tilde{\sigma}$ is the deployment strategy selected by the algorithm, $\boldsymbol{\Sigma}_{\infty}^i$ indicates the prior error covariance for the i th deployment location of strategy $\tilde{\sigma}$ as time goes infinite, $\rho(\boldsymbol{\Sigma}_{\infty}^{\tilde{\sigma}})$ stands for the largest eigenvalue of the prior error covariance matrix for strategy $\tilde{\sigma}$ as time goes infinite. Now that the estimation quality of a selected deployment location can be determined, the optimization method for finding the optimal deploying locations based on the linear dependence of sensor readings is developed.

4.2 Optimization Scheme for Sensor Node Deployment

Rapid-exploring random tree (RRT) is an efficient method for searching within a nonconvex geographical AOI by randomly building space-filling trees [69] [70]. RRT builds up a graph $G = (V, E)$, where V represents the vertices of the graph and E represents the edges of the graph. Graph G contains many spanning trees, which indicate the locations traveled within the field. The graph is in fact the state-space model, which can be used for calculating the prior estimation error when the state value is known. Thus, by expanding the trees using the strategy of RRT while sampling, we are able to determine the infinite cost of each spanning tree inside the graph generated by the RRT algorithm, and then choose the one with the smallest estimation error. The nodes on each spanning tree indicates the deployment location, where the collected data is assumed to be stationary (i.e., does not undergo significant change during the tree expansion period). Thus, the

linear dependence of sensor readings at nodes on each spanning tree can be captured and eliminated by applying the Linear Reduction method while expanding the spanning tree. The algorithm of Rapid-exploring Random tree with infinite horizon cost and Linear Reduction (RRLR) is presented in in Algorithm 2.

Algorithm 2: Rapid-exploring Random tree with Linear Reduction

Input: $V \leftarrow (x_0, y_0)$, $E \leftarrow \emptyset$, $G = (V, E)$, $iter$, $\eta \leftarrow \infty$, $\mathcal{T} \leftarrow \emptyset$, \mathcal{A} , ρ , \mathbf{A} , \mathbf{C}
Output: η , \mathcal{T}

```

1 Initialization;
2 for  $i = 1$  to  $iter$  do
3   while True do
4      $v_{rand} = \text{RAND\_NODE}()$ ;
5      $v_{nearest} = \text{NEAREST\_VERTEX}(v_{rand}, G)$ ;
6     if  $\|v_{rand} - v_{nearest}\| \leq \rho$  and  $\text{collision}(v_{new}, \mathcal{A}) = 0$  then
7        $v_{new} = v_{rand}$ ;
8       break;
9     else
10      continue;
11    end
12  end
13   $v_{near} = \text{Cluster}(G, x_{new}, r)$ ;
14  if  $|v_{near}| == 1$  then
15     $V \leftarrow V \cup v_{new}$ ;
16     $E \leftarrow E \cup e(v_{nearest}, v_{new})$ ;
17  else
18     $V \leftarrow V \cup v_{new}$ ;
19     $(v_{min}, \eta, \mathcal{T}) = \text{vertexPick}(G, v_{near}, v_{new})$ ;
20     $E \leftarrow E \cup \text{edge}(v_{min}, v_{new})$ ;
21     $V_{new} = \text{genTree}(G, v_{new})$ ;
22     $\mathbf{D} = \text{genSample}(\mathbf{A}, \mathbf{C}, V_{new}, i, M)$ ;
23     $\Theta = \text{eye}(K)$ ;
24     $\mathbf{D}, \Theta = \text{LinearReduction}(\mathbf{D}, \Theta, K)$ ;
25     $G, \mathcal{T} = \text{IndextoEliminate}(\mathbf{D}, \Theta, G, V_{new}, \mathcal{T})$ ;
26  end
27 end

```

Algorithm 2: Rapid-exploring Random tree with Linear Reduction.

Algorithm 2 has several inputs. In it G denotes the tree graph generated by the rapid-exploring random tree, and V and E stand for its vertices and edges, respectively. Also, \mathcal{T} denotes the set of

computed deployment locations, which should be empty in the beginning, and η records the infinite horizon cost calculated through prior estimation error on locations in set \mathcal{T} , which is an infinite set. Furthermore, ρ is the step size that adjusts the exploring speed, and matrices \mathbf{A} and \mathbf{C} contain the parameters of the environmental model for calculating the estimation error and the sampling value. The information about the boundaries of the geographical AOI is provided in \mathcal{A} .

The algorithm starts with the initialization in line 1 and iterates until the optimal deployment location is determined. A loop is continuously executed from line 3 to line 12 until the newly generated vertex meets several criteria. Line 4 generates the new random node of the tree and is used for finding the nearest vertex in the tree graph G in line 5. Then, line 6 examines whether the distance between the newly generated vertex and its closest vertex in the graph are within the step size and whether it is inside the boundary of the geographical AOI. Next, in line 13, the method of cluster detection is used for finding the vertices in graph G within a certain distance of the newly generated vertex. If only one vertex can be found in graph G , this provides the only option to connect the new random vertex, and it must be the nearest vertex that was just found in line 5. Thus, the graph is expanded as indicated in lines 15 and 16. Otherwise, there will be several options, and the one that minimizes the infinite horizon cost as indicated in line 19, must be chosen. The details are given in Algorithm 3. After a new edge to the spanning tree in graph G has been added with minimal infinite horizon cost, redundant sampling data should be checked. In the spanning tree with the new edge and vertex, the sensor readings for a time period can be generated according to the environmental model using matrices \mathbf{A} and \mathbf{C} in line 22. Then, line 23 generates an identity matrix and the linear reduction method is used to determine if there are linearly dependent sampling readings in line 24. The strategy that minimizes the locations for

sensor node deployment is given in line 25, which returns the reduced set \mathcal{T} . Its details are given in Algorithm 4.

Algorithm 3 aims to minimize the infinite horizon cost by comparing each possible spanning in the tree graph [13] [44]. For each vertex in the set that contains nodes near the new vertex, the spanning tree that connects to the new node is calculated as in line 6. Then its infinite horizon cost is computed in line 7. After this, the vertex and the computed deployment with the minimal cost are recorded from line 8 to line 12.

Algorithm 3: vertexPick($G, \mathbf{v}_{near}, v_{new}$)

```

Input:  $G, \mathbf{v}_{near}, v_{new}$ 
Output:  $v_{min}, \eta, \mathcal{T}$ 
1 foreach  $v_i, v_j \in \mathbf{v}_{near}$  do
2    $v_{min} \leftarrow \emptyset;$ 
3    $\eta = \infty;$ 
4    $\mathcal{T} \leftarrow \emptyset;$ 
5   if  $collision(v_i, v_{new}) == 0$  and  $collision(v_j, v_{new}) == 0$  then
6      $\tilde{\sigma} = \text{findSpanningTree}(v_i, v_j, v_{new}, G);$ 
7      $J(\tilde{\sigma}) = \text{InfiniteHorizonCost}(\tilde{\sigma});$ 
8     if  $J(\tilde{\sigma}) < \eta$  then
9        $\eta \leftarrow J(\tilde{\sigma});$ 
10       $v_{min} \leftarrow v_i;$ 
11       $\mathcal{T} \leftarrow \tilde{\sigma};$ 
12    end
13  end
14 end

```

Algorithm 3: Method of vertexPick to Minimize the Infinite Horizon Cost.

Algorithm 4 generates the graph that will reduce the size of the deployment set using linear dependence among the collected sampling data in different nodes. It first utilizes matrices \mathbf{M} and $\mathbf{\Theta}$ to find the set that stores the redundant sampling locations according to theorem 2 in line 2 to line 6. Then, from line 7 to line 18, redundant deployment locations are removed according to several strategies. If the dimension of the redundant sampling location set is not larger than the

difference between the matrix size and its rank, the locations in the set can be directly removed, and will not result in information loss in the matrix. However, if that dimension exceeds that difference, which means there are multiple choices for removing selected sampling locations, then we can only remove some of them to avoid information loss in the matrix. Hence, the size of the redundant sampling location set should be reduced to $|\mathbf{M}| - \text{rank}(\mathbf{M})$, as shown in line 13.

Algorithm 4: $\text{IndextoEliminate}(\mathbf{D}, \Theta, G, V_{new}, \mathcal{T})$

Input: $\mathbf{M}, \Theta, G, V_{new}, \mathcal{T}$
Output: G, \mathcal{T}

```

1  $\mathcal{I} \leftarrow \emptyset;$ 
2 foreach  $\text{avg}(\mathbf{M}_{.i}) \leq \epsilon$  do
3   | foreach  $\Theta_{ij} \geq \epsilon$  do
4   |   |  $\mathcal{I} \leftarrow \mathcal{I} \cup j;$ 
5   |   end
6   end
7 if  $|\mathcal{I}| \leq (|\mathbf{M}| - \text{rank}(\mathbf{M}))$  then
8   | foreach  $i \in \mathcal{I}$  do
9   |   |  $\mathcal{T} \setminus V_{new}(i);$ 
10  |   |  $G(V_{new}(i)) = \emptyset;$ 
11  |   end
12 else
13  |  $\mathcal{K} \leftarrow$  reduce the size of  $\mathcal{I}$  to  $|\mathbf{M}| - \text{rank}(\mathbf{M});$ 
14  | foreach  $k \in \mathcal{K}$  do
15  |   |  $\mathcal{T} \setminus V_{new}(k);$ 
16  |   |  $G(V_{new}(k)) = \emptyset;$ 
17  |   end
18 end

```

Algorithm 4: Method of IndextoEliminate.

4.3 Experimental Results

In this section, numerical simulation results are presented using the methodology developed in the thesis. The RRLR algorithm is applied on the dataset collected from the field tests at Yosef Wosk Reflecting Pool, UBC, Vancouver, Canada. RRLR produces several locations of deployment from the collected data, as shown in Figure 4.3, where the yellow rectangular blocks indicate the

deployment locations while the blue lines represent the exploring trees in the tree graph. Also, geographical AOI is generated based on the pool's GPS location given in meters. After that, a comparison between RRLR and the benchmark is presented.

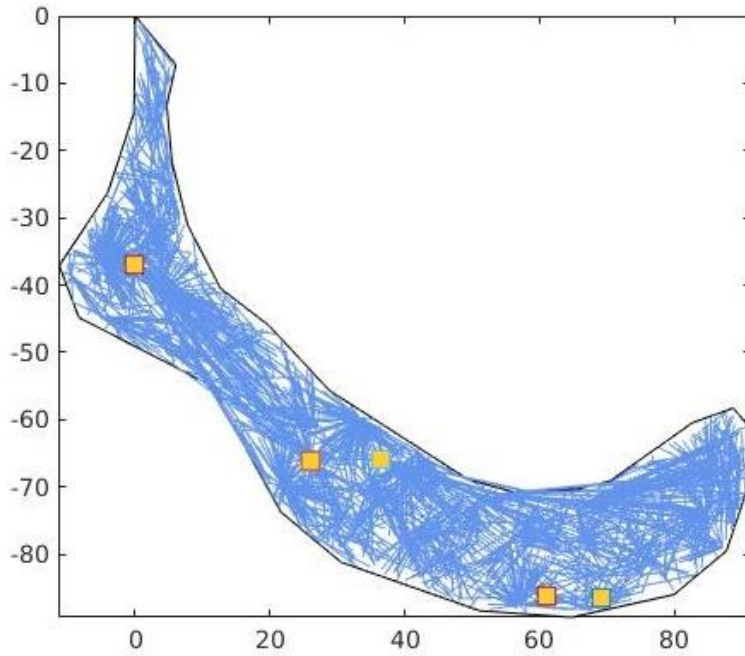


Figure 4.3: Deployment Locations Produced by RRLR.

Matrices \mathbf{A} , \mathbf{C} , \mathbf{Q} , and \mathbf{R} , which present the spatiotemporal correlation of the environment, are studied from the data sampled from five sampling locations as shown in Figure 2.10. The number

of basis functions is set to five. The parameter values for $c_i(x, y) = Ke^{-\frac{\|(x,y)-(x_i,y_i)\|^2}{2\sigma_c^2}}$ are chosen as: $\sigma_c^2 =$ square sum of the basis function variance, and $K =$ average of the entire vector. The comparison algorithm, RRC [13] is also executed using the same parameter values. All experiments are run on a PC with 4.0 GHz Quad-Core CPU and 16 G memory.

Figure 4.4 presents the deployment result generated by the algorithm Rapidly-exploring Random Cycles (RRC), which requires 12 deployment locations to achieve the overall lowest prior estimation error. Although the environment is estimated with low estimation error, that requires too many sensor nodes since it does not consider the correlation among the deployment locations when comparing to the results generated by RRLR in Figure 4.3. The method developed in the present work, RRLR, only requires five sensor nodes to persistently monitor this environment, which is close to 60% less than the solution provided by RRC. This significant reduction is due to the comparison and redundant elimination among sampling data during the tree expansion using the Linear Reduction algorithm. The result of RRLR also shows that the sensor nodes are sparsely deployed, which is practical since there is no significantly temperature change in the tested pool.

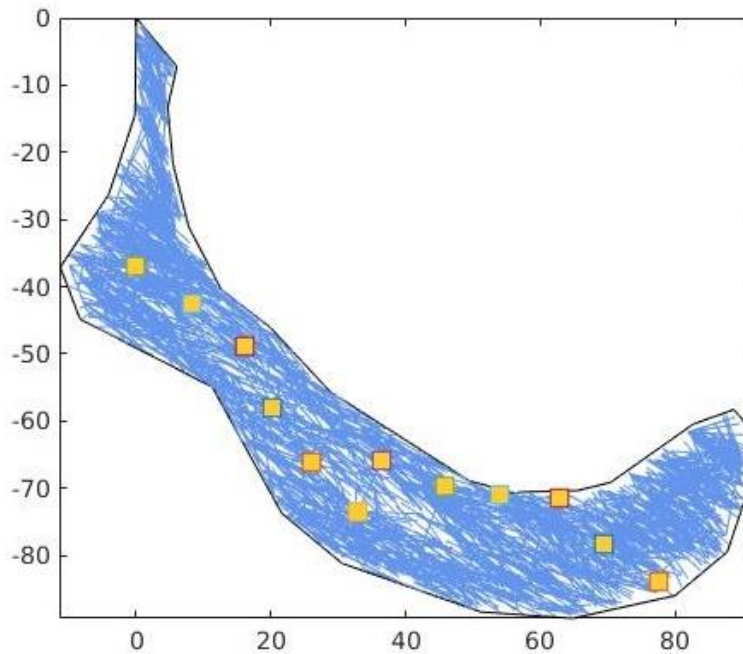


Figure 4.4: Deployment Results Generated by RRC.

Finally, the error levels in estimating the environment by the two algorithms are compared. As shown in Figure 4.5, the minimum estimation error (infinite horizon cost) for RRLR is close to

but less than the result provided by RRC, which indicates that RRLR requires fewer sensor nodes to carry out persistent monitoring within the geographical AOI. Also, RRLR can achieve low prior estimation error while RRC requires 140% more sensor nodes to achieve the same error level.

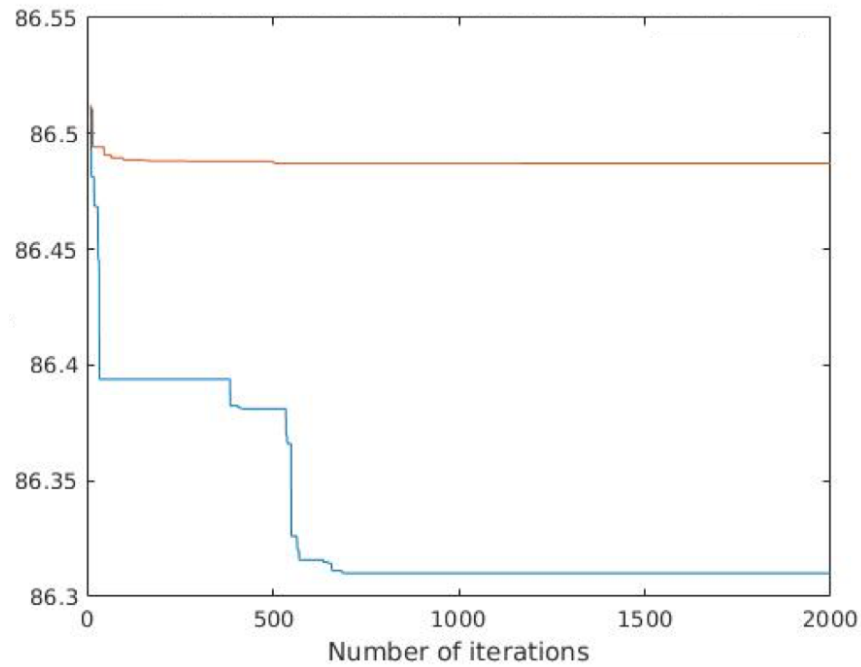


Figure 4.5: Cost Comparison between RRLR and RRC.

Chapter 5 : Conclusions and Future Work

5.1 Contributions

This thesis developed a practical platform for real time monitoring of water quality using a network of mobile sensors, which provides efficient data acquisition and transmission and less costly and easy-to-upgrade modular components. A novel strategy for optimal sensor deployment was developed, which is based on environmental estimation error and the linear dependence of sampling, so as to minimize the number of sensor nodes for the same acquired information content, through the minimization of the infinite-horizon cost.

Test results show that the developed mobile sensor network platform exhibits good performance in transmitting data and mobility in aquatic environments with strong water flow. Also, the proposed algorithm, RRLR, performs well in both estimating an aquatic environment at low prior estimation error and low linear dependence among the deployed sensor nodes.

5.2 Possible Future Work

Extensive field testing of networked multiple mobile sensor nodes will be carried in Canada and India, for monitoring active aquatic environments. Optimal collaboration among mobile sensor nodes should be further investigated, subject to various criteria of cost (e.g., energy, distance travelled, cost, and accuracy). Also, the information sharing among sensor nodes has to be improved to save energy. Besides, an improved strategy for planning the next steps of monitoring should be introduced. In the present approach, the spanning trees expend randomly without considering captured historical data, which can result in unnecessary search steps. By examining

the historical sensor data in the locations that the spanning tree has searched, we may find more beneficial places for searching. The size of random exploring tree may be reduced in this manner, and the search of the overall environment will be accelerated.

Modeling of an unknown environment should be explored. Some of water quality attributes like temperature may be modeled as prior knowledge using satellite or aerial thermal images. Utilizing historical data in this manner, at locations that are monitored, can be quite beneficial.

In the theorems that are proved in the present work, it was assumed that each sensor node contained just one sensor. The theorems will be generalized for the case of multiple sensors in each node. The developed algorithms will be implemented in the prototype multi-node mobile sensor platform, and extensive field testing will be carried out. Noise, disturbances, and unknown/unmodeled factors will enter during practical implementation and testing. Measures to mitigate such effects should be explored. In general, the developed methodologies and algorithms should be improved for accommodating such more challenging and practical issues.

Bibliography

- [1] H. A. Mona, "Elevated blood lead levels in children associated with the Flint drinking water crisis: a spatial analysis of risk and public health response.," *American journal of public health*, vol. 106, no. 2, pp. 283-290, 2016.
- [2] X. Zhang, "Emergency drinking water treatment during source water pollution accidents in China: origin analysis, framework and technologies," *Environ. Sci. Technol.*, vol. 45, no. 1, pp. 161-167., 2010.
- [3] J. Xie, *Addressing China's water scarcity: recommendations for selected water resource management issues.*, World Bank Publications, 2009.
- [4] A. Singh, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707-755, 2009.
- [5] C. Alippi, "A robust, adaptive, solar-powered WSN framework for aquatic environmental monitoring," *IEEE Sensors Journal* , vol. 11, no. 1, pp. 45-55.
- [6] P. Corke, "Environmental wireless sensor networks," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1903-1917, 2010.
- [7] M. Di Francesco, S. K. Das and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 1, p. 7, 2011.
- [8] W. Du, X. Zikun, L. Mo, H. Bingsheng, H. C. C. Lloyd and H. Miao, "Optimal sensor placement and measurement of wind for water quality studies in urban reservoirs," in

Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium, 2014.

- [9] C. Guestrin, K. Andreas and S. Ajit Paul, "Near-optimal sensor placements in gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, 2005.
- [10] H. Gupta, N. Vishnu, D. Samir and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 1, p. 4, 2008.
- [11] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 1, pp. 78-92, 2005.
- [12] M. Michaelides and C. G. Panayiotou, "Improved coverage in WSNs by exploiting spatial correlation: the two sensor case," *EURASIP Journal on Advances in Signal Processing*, vol. 1, p. 11, 2011.
- [13] X. Lan and M. Schwager, "Rapidly Exploring Random Cycles: Persistent Estimation of Spatiotemporal Fields With Multiple Sensing Robots," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230-1244, 2016.
- [14] B. Fateh and G. Manimaran, "Energy minimization by exploiting data redundancy in real-time wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 6, pp. 1715-1731, 2016.
- [15] L. A. Villas, A. Boukerche, D. L. Guidoni, H. A. D. Oliveira, R. B. D. Araujo and A. A. Loureiro, "An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in wireless sensor networks," *Computer Communications*, vol. 36, no. 9, pp. 1054-1066, 2013.

- [16] H. Shibo, C. Jiming, L. X. S. Xu and S. Youxian, "Leveraging prediction to improve the coverage of wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 701-712, 2012.
- [17] A. Singh, A. Krause, C. Guestrin and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707-755, 2009.
- [18] C. Alippi, R. Camplani, C. Galperti and M. Roveri, "A robust, adaptive, solar-powered WSN framework for aquatic environmental monitoring," *IEEE Sensors Journal*, vol. 11, no. 1, pp. 45-55, 2011.
- [19] Clearpath, "HERON by Clearpath Robotics," [Online]. Available: <https://www.clearpathrobotics.com/heron-bathymetry-unmanned-surface-vessel/>. [Accessed 05 2016].
- [20] "BlueRov2 by Blue Robotics," [Online]. Available: <http://docs.bluerobotics.com/brov2/>. [Accessed 05 2016].
- [21] M. Vuran, B. A. Özgür and F. A. Ian, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245-259, 2004.
- [22] E. Karasabun, I. Korpeoglu and C. Aykanat, "Active node determination for correlated data gathering in wireless sensor networks," *Computer Networks*, vol. 57, no. 5, pp. 1124-1138, 2013.
- [23] C. Liu, K. Wu and J. Pei, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *IEEE transactions on parallel and distributed systems*, vol. 18, no. 7, 2007.

- [24] B. Fateh and G. Manimaran, "Energy minimization by exploiting data redundancy in real-time wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 6, pp. 1715-1731, 2013.
- [25] L. M. Feeney, "An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 6, no. 3, pp. 239-249, 2001.
- [26] D. Ganesan, E. Deborah and H. John, "DIMENSIONS: Why do we need a new data handling architecture for sensor networks?," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 143-148, 2003.
- [27] N. Iri, L. Yu, H. Shen and G. Caulfield, "Congestion-adaptive data collection with accuracy guarantee in cyber-physical systems," in *Sensing, Communication, and Networking (SECON)*, 2015.
- [28] S. Adam, G. Filpus, K. Munagala and Y. Jun, "Data-Driven Processing in Sensor Networks," *CIDR*, pp. 10-21, 2007.
- [29] C. Lopes and H. S. Ali, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064-4077, 2007.
- [30] L. A. Villas, A. Boukerche, H. A. D. Oliveira, R. B. D. Araujo and A. A. Loureiro, "A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks," *Ad Hoc Networks*, vol. 12, pp. 69-85, 2014.
- [31] D. C. Mocanu, M. T. Vega and A. Liotta, "Redundancy reduction in wireless sensor networks via centrality metrics," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015.

- [32] N. Heo and P. K. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks.," *Wireless Communications and Networking*, vol. 3, pp. 1597-1602, 2003.
- [33] R. Ghrist and M. Abubakr, "Coverage and hole-detection in sensor networks via homology.," *Information Processing in Sensor Networks, 2005. IPSN*, pp. 254-260, 2005.
- [34] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 2, pp. 262-267, 2011.
- [35] M. Abo-Zahhad, M. A. Sabah, N. Sabor and A. F. Al-Ajlouni, "The convergence speed of single-and multi-objective immune algorithm based optimization problems," *Signal Processing: An International Journal (SPIJ)*, vol. 4, no. 5, p. 247, 2010.
- [36] M. Abo-Zahhad, S. M. Ahmed, N. Sabor and S. Sasaki, "Rearrangement of mobile wireless sensor nodes for coverage maximization based on immune node deployment algorithm," *Computers & Electrical Engineering*, vol. 43, pp. 76-89, 2015.
- [37] M. Abo-Zahhad, N. Sabor, S. Sasaki and S. M. Ahmed, "A centralized immune-Voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks," *Information Fusion*, vol. 30, pp. 36-51, 2016.
- [38] M. A. Alsheikh, L. Shaowei, H.-P. Tan and D. Niyato, "Area coverage under low sensor density," in *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2014.
- [39] A. A. Bara'a, A. K. Enan and A. Cosar, "Multi-objective evolutionary routing protocol for efficient coverage in mobile sensor networks," *Soft Computing*, vol. 19, no. 10, pp. 2983-2995, 2015.

- [40] A. Aziz, N. A. Bt, A. W. Mohemmed and B. D. Sagar, "Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization," in *Intelligent and Advanced Systems, 2007. ICIAS 2007.*, 2007.
- [41] N. Ganganath, C. Chi-Tsun and K. T. Chi, "Distributed antiflocking algorithms for dynamic coverage of mobile sensor networks.," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1795-1805, 2016.
- [42] F. Li, J. Luo, W. Wang and Y. He, "Autonomous Deployment for Load Balancing k-Surface Coverage in Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 279-293, 2015.
- [43] N. Cressie, *Statistics for spatial data*, John Wiley & Sons, 2015.
- [44] X. Lan and M. Schwager, "Planning periodic persistent monitoring trajectories for sensing robots in gaussian random fields," *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2415-2420, 2013.
- [45] X. Lan and M. Schwager, "A variational approach to trajectory planning for persistent monitoring of spatiotemporal fields," *American Control Conference (ACC)*, pp. 5627-5632, 2014.
- [46] X. Lan and M. Schwager, "Continuous curvature path planning for semi-autonomous vehicle maneuvers using RRT," *2015 European Control Conference (ECC)*, pp. 2360-2365, 2015.
- [47] L. Xiaodong and S. M, "Rapidly Exploring Random Cycles: Persistent Estimation of Spatiotemporal Fields With Multiple Sensing Robots," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230 - 1244, 2016.

- [48] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," *International Symposium on Information Processing in Sensor Networks*, p. 55, 2005.
- [49] T. P. Lambrou and C. G. Panayiotou, "A testbed for coverage control using mixed wireless sensor networks.," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 527-537, 2012.
- [50] Y. Qu and S. V. Georgakopoulos, "A centralized algorithm for prolonging the lifetime of wireless sensor networks using particle swarm optimization," in *In Wireless and Microwave Technology Conference (WAMICON)*, 2012.
- [51] C. Haas and J. Wilke, "Energy evaluations in wireless sensor networks: a reality check," in *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, 2011.
- [52] N. Kamyabpour and D. B. Hoang, "Modeling overall energy consumption in Wireless Sensor Networks," *arXiv preprint arXiv:1112.5800*, 2011.
- [53] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd annual Hawaii international conference on System sciences*, 2000. .
- [54] Z. Zhang, T. P. Jung, S. Makeig, Z. Pi and B. D. Rao., "Spatiotemporal sparse Bayesian learning with applications to compressed sensing of multichannel physiological signals," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 22, no. 6, pp. 1186-1197., 2014.
- [55] "Libelium," [Online]. Available: <http://www.libelium.com/>. [Accessed 09 2015].

- [56] "300M WIFI ADAPTER WITH ANTENNA by EDUP," [Online]. Available: <http://www.szedup.com/product-item/300m-wifi-adapter-antenna/#tab-id-2>. [Accessed 02 2017].
- [57] V. G. Cerf and R. E. Kahn, " A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications*, vol. 22, no. 5, p. 637–648, May 1974.
- [58] A. Krause, C. Guestrin, A. Gupta and J. Kleinberg, "Near-optimal sensor placements: Maximizing information while minimizing communication cost," in *In Proceedings of the 5th international conference on Information processing in sensor networks*, 2006.
- [59] "3DR Pixhawk Mini," 3D Robotics, [Online]. Available: <https://store.3dr.com/products/3dr-pixhawk>. [Accessed 09 2016].
- [60] D. J. Terry, "Gaussian Elimination with Partial Pivoting," [Online]. Available: http://web.mit.edu/10.001/Web/Course_Notes/GaussElimPivoting.html. [Accessed 06 2016].
- [61] W. Wardlaw, "Row rank equals column rank," *Mathematics Magazine*, vol. 78, no. 4, pp. 316-318, 2005.
- [62] S. Banerjee and R. Anindya, *Linear algebra and matrix analysis for statistics*, CRC Press, 2014.
- [63] S. M. LaValle, *Planning algorithms*, Cambridge University Press, 2006.
- [64] L. Zhao, Z. Wei, H. Jianghai, A. Alessandro and T. Claire, "On the optimal solutions of the infinite-horizon linear sensor scheduling problem," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2825-2830, 2014.

- [65] S. Karaman and F. Emilio, "Sampling-based algorithms for optimal motion planning with deterministic μ -calculus specifications," in *American Control Conference (ACC)*, 2012.
- [66] J. Cortés, "Distributed Kriged Kalman filter for spatial estimation," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2816-2827, 2009.
- [67] "Estimate state-space model using subspace method," [Online]. Available: <https://www.mathworks.com/help/ident/ref/n4sid.html>. [Accessed 12 2016].
- [68] K. Rudolph Emil, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35-45, 1960.
- [69] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.
- [70] S. M. LaValle and J. K. James, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378-400, 2001.
- [71] A. Liu, J. Xin, C. Guohua and C. Zhigang, "Deployment guidelines for achieving maximum lifetime and avoiding energy holes in sensor network," *Information Sciences*, vol. 230, pp. 197-226, 2013.