EFFICIENT DATA ACQUISITION IN WIRELESS SENSOR NETWORKS WITH PARTIALLY OBSERVABLE INFORMATION

Sunisa Chobsri

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Telecommunication Engineering

Suranaree University of Technology

Academic Year 2008

การค้นข้อมูลอย่างมีประสิทธิภาพในโครงข่ายตัวตรวจรู้ไร้สาย
ที่มีข้อมูลที่สังเกตได้บางส่วน

นางสาวสุนิสา จบศรี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2551

EFFICIENT DATA ACQUISITION IN WIRELESS SENSOR NETWORKS WITH PARTIALLY OBSERVABLE INFORMATION

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master's Degree.

Thesis Examining Committee

____
(Asst. Prof. Dr.Rangsan Tongta)
Chairperson

____
(Asst. Prof. Dr.Wipawee Hattagam)
Member (Thesis Advisor)

____
(Assoc. Prof. Dr.Kittisak Kerdprasop)
Member

____
(Prof. Dr.Pairote Suttayatham)
Vice Rector for Academic Affairs

____
(Assoc. Prof. Dr.Vorapot Khompis)
Dean of Institute of Engineering

สุนิสา จบศรี : การค้นข้อมูลอย่างมีประสิทธิภาพในโครงข่ายตัวตรวจรู้ไร้สายที่มีข้อมูลที่สังเกตได้บางส่วน (EFFICIENT DATA ACQUISITION IN WIRELESS SENSOR NETWORKS WITH PARTIALLY OBSERVABLE INFORMATION) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.วิภาวี หัตถกรรม, 128 หน้า

งานวิจัยนี้นำเสนอวิธีการค้นข้อมูลอย่างมีประสิทธิภาพ โดยสามารถคงระดับความเชื่อมั่น (probabilistic confidence) ของการเก็บรวบรวมข้อมูลได้ตามต้องการในโครงข่ายตัวตรวจรู้ไร้สายที่มีความผิดพลาด (error-prone WSN)

สำหรับการประยุกต์ใช้งานโครงข่ายตัวตรวจรู้ไร้สายนั้น ประกอบไปด้วยตัวตรวจรู้จำนวนมาก ทำให้ข้อมูลเกิดความซ้ำซ้อนกัน ข้อมูลในโครงข่ายมีสหสัมพันธ์ทั้งเชิงเวลาและเชิงตำแหน่งอย่างมาก (spatial and temporal correlation) ซึ่งข้อมูลส่วนมากอาจจะไม่มีประโยชน์ต่อผู้ใช้เลย นอกจากนี้ ตัวตรวจรู้อาศัยพลังงานแบตเตอรี่ในการติดต่อสื่อสารภายในโครงข่าย ดังนั้นวิธีการค้นข้อมูลอย่างมีประสิทธิภาพที่ใช้พลังงานอย่างประหยัดจึงมีความจำเป็นอย่างยิ่ง งานวิจัยทางด้านนี้ ได้มีการศึกษากันอย่างแพร่หลาย โดยสามารถแบ่งออกเป็น 2 กลุ่ม คือ งานวิจัยที่ศึกษาการค้นข้อมูลโดยพิจารณาถึงคุณภาพของข้อมูลที่สถานีฐานได้รับเท่านั้น และงานวิจัยที่ศึกษาการค้นข้อมูลโดยพิจารณาถึงการบริโภคพลังงานในระยะยาว แต่ศึกษาถึงผลกระทบจากคุณภาพของข้อมูลที่สถานีฐานจะได้รับอย่างชัดเจน ดังนั้นในการวิจัยนี้จึงมีจุดประสงค์ที่จะระบุปัญหาการค้นข้อมูลอย่างมีประสิทธิภาพ ซึ่งสามารถเลือกตัวตรวจรู้ได้ดีที่สุด และยังคงจุดสมดุลระหว่างคุณภาพของข้อมูลกับผลรางวัลของการสื่อสารในระยะยาว ของโครงข่ายตัวตรวจรู้ไร้สายที่มีความผิดพลาด วิทยานิพนธ์นี้มีองค์ความรู้หลักสองประการ

องค์ความรู้ประการแรก คือ การกำหนดปัญหาการค้นข้อมูลอย่างมีประสิทธิภาพในโครงข่ายตัวตรวจรู้ไร้สายให้เป็นกระบวนการการตัดสินใจแบบมาคอฟภายใต้สภาวะที่การสังเกตได้บางส่วนแบบสถานะเต็มหน่วย (discrete-state partially observable Markov decision process) ซึ่งมีจุดมุ่งหมายเพื่อหานโยบายการเลือกตัวตรวจรู้เพื่อให้ได้ผลรางวัลเฉลี่ยในระยะยาวสูงที่สุด (average long-term reward) โดยประยุกต์ใช้วิธีวิทเนส (witness algorithm) ในการแก้ปัญหาสำหรับการเลือกตัวตรวจรู้ โดยที่สามารถคงความเชื่อมั่นของข้อมูลได้ตามต้องการ

องค์ความรู้ประการที่สอง คือ การขยายปัญหาแบบกระบวนการตัดสินใจแบบมาคอฟภายใต้สภาวะที่การสังเกตได้บางส่วนแบบสถานะเต็มหน่วยไปสู่กระบวนการการตัดสินใจแบบมาคอฟภายใต้สภาวะที่การสังเกตได้บางส่วนแบบสถานะต่อเนื่อง (continuous-state partially observable Markov decision process) เพื่อสามารถนำไปประยุกต์ใช้งานได้เหมาะสมยิ่งขึ้นในสถานการณ์จริง ซึ่งการค้นข้อมูลอย่างมีประสิทธิภาพนั้น ได้กำหนดเป็นกระบวนการการตัดสินใจ

แบบมาคอฟภายใต้สภาวะที่การสังเกตได้บางส่วนแบบพาราเมตริค (parametric partially observable Markov decision process) ซึ่งเป็นกระบวนการการตัดสินใจแบบมาคอฟภายใต้สภาวะที่การสังเกตได้บางส่วนแบบสถานะต่อเนื่องแบบหนึ่ง โดยมีจุดมุ่งหมายเพื่อหานโยบายการเลือกตัวตรวจรู้เพื่อให้ได้ผลรางวัลเฉลี่ยในระยะยาวสูงที่สุด โดยประยุกต์ใช้วิธีฟิทเททเวลู่อิเทอเรชั่น (fitted value iteration) สำหรับการแก้ปัญหาในการเลือกตัวตรวจรู้ โดยที่สามารถคงความเชื่อมั่นของข้อมูลได้ตามต้องการ

สาขาวิชาวิศวกรรมโทรคมนาคม ลายมือชื่อนักศึกษา____________________

ปีการศึกษา 2551 ลายมือชื่ออาจารย์ที่ปรึกษา______________

SUNISA CHOBSRI : EFFICIENT DATA ACQUISITION IN WIRELESS SENSOR NETWORKS WITH PARTIALLY OBSERVABLE INFORMATION. THESIS ADVISOR : ASST. PROF. WIPAWEE HATTAGAM, Ph.D. 128 PP.

WIRELESS SENSOR NETWORKS (WSNs)/ EFFICIENT DATA ACQUISITION/ PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (POMDP)/ PARAMETRIC PARTIALLY OBSERVABLE MARKOV DECISION PROCESS (PPOMDP)/ WITNESS ALGORITHM (WIT)/ FITTED VALUE ITERATION (FVI)

This research proposes an efficient data acquisition scheme which aims to satisfy probabilistic confidence requirements of the collected data in an error-prone wireless sensor network (WSN).

In WSN applications, sensor nodes generate huge amount of redundant data which exhibit high spatial and temporal correlation. Most data may have little benefit to the users' interpretation. Furthermore, these sensor nodes consume the limited on-board resources during data acquisition. Hence, an efficient data acquisition scheme which collects data with low resource consumption is needed. Most researches in the area have merely studied the quality of the collected data and the long-term resource consumption. However, the quality of the collected data has not been guaranteed explicitly. Therefore, the underlying aim of this thesis is to address the problem of efficient data acquisition which selects the best sensor nodes to acquire while maintaining a balance in the data quality against the *long-term* average communication reward in error-prone wireless sensor networks.

There are two main contributions in this thesis :

The first contribution is the formulation of the efficient data acquisition problem in WSNs as a discrete-state partially observable Markov decision process (POMDP), which aims to a find sequence of sensor selection that maximizes the average long-term reward for the system. An existing tool used for solving POMDPs called the witness algorithm is then employed to find a good sensor selection plan such that the requirements of the data quality on the collected data are still satisfied.

The second contribution is the extension from the discrete-state POMDP formulation to the continuous-state POMDP formulation which allows a more realistic approach to the data acquisition problem. A type of continuous-state POMDP formulation namely the parametric partially observable Markov decision process (PPOMDP) has been employed to formulate the efficient data acquisition problem which supports probabilistic confidence requirements in an error-prone WSN. An approximate algorithm called the fitted value iteration (FVI) is applied to find a good sensor selection scheme.

School of Telecommunication Engineering Student's Signature________________

Academic Year 2008 Advisor's Signature________________

ACKNOWLEDGEMENT

I am grateful to all those, who by their direct or indirect involvement have helped in the completion of this thesis.

First and foremost, I wish to express my sincere thanks to my thesis advisor, Asst. Prof. Dr. Wipawee Hattagam for her invaluable help and constant encouragement throughout the course of this research. She provided many insightful ideas and comments throughout my master's degree.

I am grateful for the lecturers in the School of Telecommunication Engineering, Asst. Prof. Dr. Rangsan Tongta, Asst. Prof. Dr. Rangsan Wongsan, Asst. Prof. Dr. Chanchai Tongsopa, Asst. Prof. Dr. Chutima Prommak and others for their suggestions and all their help.

I would also like to thank Assoc. Prof. Dr. Kittisak Kerdprasop for accepting to serve in my committee.

My special thanks goes to Mr. Prapol Jarataku for his full support.

My sincere appreciation goes to Ms. Maneerat Tumpong for her valuable administrative support during the course of my dissertation.

Finally I am most grateful to my parents and my friends both in both masters and doctoral degree courses for all their support throughout the period of this research.

Sunisa Chobsri

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

SYMBOLS AND ABBREVIATIONS

WSNs	Wireless sensor networks
BS	Base station
MDP	Markov decision process
POMDP	Partially observable Markov decision process
PPOMDP	Parametric Partially observable Markov decision process
PWLC	Piecewise-linear and convex
WIT	Witness algorithm
FVI	Fitted value iteration
RA	Randomized algorithm
HSA	Heuristic search algorithm
ALR	Average long-term reward
AEC	Average energy consumption
LP	Linear programming
t	Time
$\{S_t\}$	Random process at time t
S_t	State of the process at time t
S	State space
s	Current state
s'	Next state
$E[.]$	Expectation operator

SYMBOLS AND ABBREVIATIONS (Continued)

A	Action space
a	Action
Θ	Observation space
θ	Observation
$g(s,a,s')$	Expected reward given any current state s and an action a with any next state s'
π	Policy
π^*	Optimal policy
$V^{\pi}(s)$	Value function of a state (s) under policy π
$V^*(s)$	Value function of a state (s) under optimal policy π^*
γ	Discount factor
B	Belief state space
b	Belief state
$\Gamma(.)$	Belief transition function
$\tilde{\Gamma}(.)$	Approximate belief transition function
$V^{\pi}(b)$	Value function of a belief state (b) under policy π
$V^*(b)$	Value function of a belief state (b) under optimal policy π^*
β	Sufficient statistics space
Φ	Sufficient statistics

SYMBOLS AND ABBREVIATIONS (Continued)

$\tilde{V}^*(\Phi)$	Optimal value function of sufficient statistics (Φ)
Ψ	Set of belief point-value pairs
F_β	Function approximation
$\bar{\mu}$	Means vector
$\mathbb{C}$	Covariance matrix
ε	Error bound
δ	Probabilistic confidence requirement
$Conf$	Confidence level of the correlation model
E_s	Energy consumption at the sending mode
E_r	Energy consumption at the receiving mode
E_{att}	Energy consumption for a specific attribute
n	Number of sensor nodes in the network
q	Quantization level of state space

CHAPTER I

INTRODUCTION

This chapter provides a background on data acquisition problem in wireless sensor networks (WSNs) and highlights the significance of data acquisition problems using sensor selection methods. The problem is formulated as a partially observable Markov decision process (POMDP) and applies the witness algorithm (WIT) and fitted value iteration (FVI) for solving POMDP formulated data acquisition problem in a WSN.

1.1 Significance of the Problem

Wireless sensor networks (WSNs) are networks consisting of spatially distributed wireless sensors which cooperatively monitor physical information of the environment, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. These quantities, so called attributes, are monitored in order to perform certain tasks, such as obtain the values of physical variables at a given location, detect the occurrence of events of interest and estimate parameters of the detected event of events, classify a detected object, and track an object, etc. Therefore, the important characteristics of WSNs include the use of a large number of sensors, attachment of stationary sensors, low energy consumption, self-organization capability, collaborative signal processing, and querying ability (Mahalik, 2007). The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance.

In the present, wireless sensor networks are used in various applications, including environment and habitat monitoring, logistics and tracking, healthcare applications, home automation, and traffic control. These applications typically require use of a large number of redundant sensors. Particularly in environment monitoring applications, sensors are deployed to acquire physical information of the environment. The base station which collects such data must therefore deal with huge amount of redundant data. In addition, physical information can change continuously with time. Due to limited battery power, onboard processing capability and network bandwidth, acquiring data from the environment is only sampled periodically. In such wireless systems, one common problem is that the sensor readings can be uncertain, noisy and error-prone. Therefore, data received at the BS may provide incorrect information which could result in misinterpretation. In order to solve this problem, a data acquisition scheme which efficiently collects sensor readings with low resource consumption and provides quality assurance on the collected data is needed.

The related works on efficient data acquisition in error-prone WSNs can be classified into three major groups, based on the number of sensors used for sensing the environment, which include sensor sampling, sensor scheduling and sensor selection.

1.1.1 Sensor Sampling

Sensor sampling is a conventional data acquisition scheme which acquires data from all sensors in the network at periodic time intervals. In a recent work, Galmes (2006) used a sensor sampling scheme based on random sampling for vineyard monitoring. The advantage of this scheme is the easy implementation.

However, determining an appropriate periodic time interval which specifies how frequent a sensor should sense the field is an important issue in order to achieve the efficient sensor sampling scheme. Moreover, this scheme requires collecting data from as many sensors as possible to have a better view of the sensor surroundings. However, due to energy limitations, the number of active sensors should be kept to a minimum. Furthermore, some data may even provide incorrect information about the environment. So, the BS does not need to receive data from all sensors.

1.1.2 Sensor Scheduling

Sensor scheduling is a data acquisition scheme which only one sensor is allowed to acquire a data at any time. A recent work by Gupta et al. (2004) randomly sampled a sensor per unit time from a suitable probability distribution. They are only concerned with limited computation, but do not consider communication noise in the wireless system. In addition, a sensor scheduling problem is investigated in previous works by Yavuz and Jeffcoat (2007) for a single selected sensor to maintain a bounded estimate of position at multiple locations. The advantage of this scheme is the minimal resource consumption (i.e. in terms of bandwidth and energy) since both communication and energy consumption are generated from only one sensor. However, the collected data may not always be able to correctly represent all physical information of the environment. That is, data from one single sensor may be insufficient to represent the physical information of the environment in a large network.

1.1.3 Sensor Selection

Sensor selection is a data acquisition scheme which selects some appropriate sensors to acquire data in the network. The decision as to which sensor should be selected to sense the environment takes into account a variety of factors depending on the algorithm, such as residual energy, required coverage, required data quality, or the type of information required. Sensor selection schemes can be classified based on the purpose of selection into three classes as follows (Rowaihy et al., 2007).

1.1.3.1 Coverage Schemes

In coverage sensor selection schemes, sensors are selected in order to ensure full coverage for the field. In other words, every point in the field must be in the sensing range of at least one sensor. Typically, the sensors are densely deployed, resulting in redundant coverage. Therefore, sensor selection is used to select only a subset of sensors in order to achieve full coverage while conserving energy and hence extending the network lifetime. Cardei and Du (2005), Shih et al. (2006) and Lu et al. (2005) investigated sensor selection schemes in WSNs by considering a minimal subset of sensor readings while guaranteeing a desired coverage area. These schemes focus on the trade-off between the energy consumption and the desired coverage only. They do not consider the quality of the data acquired from an error-prone WSN. Hence, if faulty sensors are selected, the quality of the acquired data may be affected.

1.1.3.2 Target Tracking and Localization Schemes

Target tracking and target localization sensor selection schemes are mostly applied in military applications. The area of interest extends from general information collection, to enemy tracking, battlefield surveillance or target classification. The sensors are densely scattered to collect important information in the area. Therefore, this scheme is focused on selecting a minimal subset of sensors to track the desired target and estimate a target location. Several researches such as Isler and Bajcsy (2006), Sadaphal and Jain (2006), and Fuemmeler and Veeravalli (2008), emphasize only on the trade-off between the prediction error and the number of selected sensors. However, if faulty sensors are selected, the quality of the acquired data may be affected resulting in user misinterpretation.

1.1.3.3 Mission Assignment Schemes

In a WSN which must perform a specific mission repeatedly over time, sensors are selected such that the mission is accomplished in the most efficient manner. The objective of these sensor selection schemes is to select the sensor nodes which are most useful for the mission. The meaning of “usefulness” refers to utility or quality maximization, or entropy minimization. In a recent work by Zhang, Fang and Li (2008) improved the quality of acquired data by using a spatial correlation model to estimate the data when the error of the model’s corresponding estimation exceeds an allowable error bound. Typically, data collected from sensor field are highly correlated. There are several works which have exploited the spatial and temporal correlations to enhance sensor selection. Liu, Wu and Pei (2007) chose a subset of sensors such that the data from selected sensors are directly acquired and used to build spatial and temporal correlation models. These models are, in turn, used to calculate estimate values for the non-collected nodes while the error of value estimation is still satisfied. However, these works emphasize on the prediction error trade-off only. Moreover, sensor selection is determined independently in each time step. Therefore, the sensor selection plan could be optimal in the immediate time, but may not be optimal in the long run. On the other hand, Meliou et al. (2007) minimized the long-term sensor selection cost within a planning horizon. In particular, the energy consumption and communication costs over a finite horizon were minimized by using used spatial and temporal correlation models. Although, the long-term sensor selection has been investigated in their work, the quality of the collected data has not been explicitly guaranteed. Quality of the collected data is guaranteed in Deshpande et al. (2004), where spatial and temporal correlation models are used. Sensors are used to acquire data only when the model itself is not sufficiently rich to answer the query with acceptable confidence. An optimized sensor selection plan in terms of energy usage and the successful transmission is determined independently in each time step. Such sensor selection plan may be optimal in the short run, but may not be optimal in the long run.

Therefore, the underlying objective of this thesis is to determine a sensor selection plan which selects the best sensor readings to acquire while still guaranteeing the data quality in an error-prone WSN. In a recent work by Rezaeian (2007), the sensor selection problem formulated as an average cost (as entropy) Markov decision process (MDP) is presented. They use entropy rate to deal with inaccurate sensor readings in the network. The solution to their MDP formulated problem is a sequence of sensor selection decisions which are optimal in the long run. However, the performance and quality of the collected data have not been numerically evaluated yet. The long-term performance criterion is considered particularly advantageous over the short-term performance criterion for the data acquisition problem in error-prone WSNs. To illustrate this, consider a model-based data acquisition system where a model of sensor data is maintained at the base station (BS). Upon receiving a query, the BS can acquire data from the sensor field or estimate the query answer from such model. The best immediate decision is to select the model to the predict sensor reading as this does not consume any resource. However, the model may later become out-of-date and provide in accurate estimates. Therefore, the best immediate decision may be suboptimal in the long run. Additionally, sensor readings may be uncertain, noisy and error-prone. Therefore, data received at the BS may provide incorrect or inaccurate information to users. Motivated by the need for an optimal long-term sensor selection plan and guarantee on data quality, we therefore propose two frameworks in this thesis based on partially-observable Markov decision process (POMDP) formulation, to capture the uncertainty of the sensor readings in an error-prone WSN. Firstly, we propose a data acquisition problem as a discrete-state POMDP framework which represents the sensor readings in the form of discrete values. To demonstrate the optimality of this framework, we then solve for a good long-term sensor selection plan in finite horizon by using an existing tool called the witness algorithm (Cassandra, 1995). However, most sensor readings are real-valued (such as temperature, humidity, pH, etc.). Furthermore, the solution becomes intractable with fine discretization of sensor readings. Therefore, to capture the *uncertainty of continuous-valued* data in error-prone WSNs, we propose the second framework which formulates the data acquisition problem as a continuous-state partially observable Markov decision process (POMDP). To evaluate its performance, we then solve for a good sensor selection plan by using an existing tool called the fitted valued iteration (FVI) method. (Brooks, Makarenko, Williams and D-Whyte, 2006)

To conclude, the main contributions of this thesis are fourfold:

1) We proposed a data acquisition problem formulation as a discrete-state POMDP.
2) We applied the witness algorithm which is an existing analytical tool for solving discrete-state POMDP problems to solve for a good long-term sensor selection plan in an error-prone WSN.
3) We proposed a data acquisition problem formulation as a continuous-state POMDP. Since a parametric representation of the probabilistic distribution over continuous-states was used to simplify the problem, the formulation is referred to as a parametric POMDP (PPOMDP) formulation.
4) We applied an existing method called the fitted value iteration to solve continuous-state PPOMDP problems for a good long-term sensor selection plan in an error-prone WSN.

1.2 Research Objectives

The objectives of this research are as follows:

1.2.1 To study efficient data acquisition methods in WSNs with partially observable information.

1.2.2 To formulate the data acquisition problem as a discrete-state POMDP whose objective is to maximize some long-term performance criterion.

1.2.3 To formulate the data acquisition problem as a continuous-state POMDP to extend the data acquisition problem to a more realistic scenario.

1.2.4 To apply the witness algorithm (WIT) and fitted value iteration (FVI) to solve a data acquisition problem while balancing the trade-off between information quality and data acquisition performance in WSNs.

1.2.5 To compare performance metrics of the fitted value iteration with the witness algorithm in terms of data acquisition performance such as average reward, average energy consumption and average confidence level.

1.3 Assumptions

1.3.1 The data acquisition in WSNs can be formulated as a POMDP problem.

1.3.2 The witness algorithm can achieve a good sensor selection policy which can satisfy data quality requirements in WSNs.

1.3.3 The fitted value iteration scheme can achieve a good sensor selection plan and can satisfy the requirements of data quality in WSNs.

1.4 Scope of the Research

The experiment is separated into two parts. In the first part, the data acquisition problem which supports probabilistic data quality assurance in WSNs is formulated as POMDP and is solved with an exact method called the witness algorithm (WIT). WIT will be compared with existing data acquisition tools, such as the randomized algorithm (RAN) which makes a randomized decision, and the heuristic search algorithm (HSA) which makes a decision to optimize the immediate (short-term) performance criterion. To evaluate the performance, four metrics are compared, namely, the average long-term reward, the average energy consumption, the percentage which a_0 is selected and the percentage of queries that meet the required confidence level. We use the average long-term reward metric to demonstrate the optimality of the decision, the average energy consumption and the percentage which a_0 is selected metrics to demonstrate the ability to save energy, whereas the percentage of queries that meet the required confidence level is used to demonstrate the ability to maintain probabilistic data quality assurance in the WSNs.

The second part extends the study from the first part to a more realistic scenario by employing the continuous-state POMDP formulation for the data acquisition problem to satisfy probabilistic data quality requirements in WSNs. An approximate algorithm, called the fitted value iteration (FVI), is applied to solve for a good sensor selection plan in the long-term. We study the use of the temporal correlation model alone, and both the spatial and temporal correlation models at the base station. The experiments are conducted with the same metrics and compared with the same sensor selection schemes as in part one.

1.5 Expected Usefulness

1.5.1 To obtain an efficient sensor selection scheme for data acquisition problem in error-prone WSNs.

1.5.2 To obtain a good sensor selection scheme by using WIT which can support probabilistic data quality assurance in WSNs.

1.5.3 To obtain a good sensor selection scheme by employing FVI which can reduce the computational burden that increases rapidly with fine state discretization while still satisfying the probabilistic confidence requirements of acquired data in WSNs.

1.6 Synopsis of Thesis

The remainder of this thesis is organized as follows. **Chapter 2** presents the theoretical background which is the foundation for the contributions of this thesis. Firstly, the concept of the general POMDP formulation is reviewed. This is followed by an introduction of the continuous-state POMDP formulation which is used for capturing uncertainty in the real world data. Next, an existing tool used for solving POMDP, called witness algorithm (WIT), is introduced and followed by fitted value iteration (FVI) which is employed to solve continuous-state POMDP.

Chapter 3 studies the data acquisition problem which aims to satisfy probabilistic confidence requirements of the acquired data in wireless sensor networks (WSNs). The data acquisition problem is formulated as a partially observable Markov decision process (POMDP) and solved by using an exact analytical method called witness algorithm (WIT).

Chapter 4 extends the contribution of the previous chapter to a more realistic scenario by employing the continuous-state POMDP formulation. Since most sensor readings are real-valued, the advantage of this approach is that it caters a more realistic sensor reading scenario. The fitted value iteration (FVI) is used to solve for the sensor selection plan under such formulation.

Finally**, chapter 5** summarizes all the findings and original contribution in this thesis and points out possible future research directions.

CHAPTER II

BACKGROUND THEORY

2.1 Introduction

This thesis studies the data acquisition problem which supports probabilistic confidence requirements of the data quality in wireless sensor networks. Typically, wireless sensor networks employed for environmental monitoring purposes, require use of a large number of redundant sensor nodes. These sensor nodes are deployed to acquire physical information of the environment which exhibit high spatial and temporal correlation. Acquiring as much data as possible from the environment at a given point in time may result in high querying costs in both time and power consumption. Furthermore, most of the data may provide little benefit to the quality of query results. Wireless sensors networks (WSNs), in particular, have limited communication, computation processing ability and battery life. Hence, a data acquisition scheme which efficiently collects sensor reading with low resource consumption is needed.

Since high spatial and temporal correlation is an inherent feature of physical information of the environment, Deshpande et al. (2004) applied both spatial and temporal correlation models for data acquisition which supports probabilistic data quality assurance. In particular, by modeling the temporal correlation of sensor readings received at the base station (BS), a stochastic probabilistic model of the behavior of these sensor readings can be characterized.

Therefore, the sensor values received earlier in time should assist in estimating the sensor values later on (Deshpande et al., 2004). Such characterization of sensor readings received at the BS is assumed to hold the *Markov property*. This assumption is verified by statistical tests by generating sensor readings from various Gaussian distributions. The results from the verification study show that the probability of future states occurring in the process given the present state is equal to the probability of future states occurring in the process given the history of the process (see **Appendix II**). In addition, since the change of sensor values received at the BS depends on the (BS's) decision on which sensor(s) is selected, the data acquisition problem at the BS can be viewed as a *Markov decision processes* (MDP).

Despite being able to formulate the data acquisition problem based on sensor readings as a MDP, sensor readings do not always exhaustively represent the actual data in the real world. This may be caused by sediments attached to the sensor from its surroundings through rain or wind, the sensors themselves are faulty or transmission errors (packet corruption or loss) which occur as data is relayed from node to node. Therefore, the received data at the base station does not always accurately capture the actual state of the environment. In other words, the base station receives data that is merely a *partial observation* of the actual state of the environment.

To capture the imprecise information received at the BS, the data acquisition problem therefore can be formulated as a partially observable Markov decision process (Kaelbling, Littman and Cassandra, 1998). To solve the POMDP problem, we apply an existing tool called the witness algorithm (Cassandra, 1995) to determine a good sensor selection plan which selects the best sensor readings while maintaining a balance between the long-term communication cost and the quality of the acquired data in an error-prone sensor reading environment. The scheme formulates the problem as a *discrete-state* partially observable Markov decision process because it quantizes real-valued sensor readings into discrete intervals. However, computational burden increases rapidly with fine discretisation. Furthermore, most sensor readings are continuous-valued. Therefore, to capture the uncertainty of *continuous-valued* sensor readings in error-prone WSNs, we extend the data acquisition framework by formulating it as a parametric partially observable Markov decision process (PPOMDP) (Brooks, Makarenko, Williams and D-Whyte, 2006). Just as the POMDP framework, the aim of the PPOMDP scheme is to find a sensor selection scheme which best refines the query answer with acceptable confidence. To evaluate the performance of the PPOMDP framework, we then solve for a sensor selection plan in finite horizon by using an existing tool called the fitted valued iteration (FVI) method (Brooks, Makarenko, Williams and D-Whyte, 2006).

Therefore, this chapter serves as an introductory to the fundamental theory of POMDP which is the basis of the contribution of this thesis. The next section provides a theoretical background on Markov decision process (MDP) theory. A description of partially observable Markov decision process (POMDP) theory is given in section 2.3. Section 2.4 presents the parametric partially observable Markov decision process theory. Section 2.5 describes the witness algorithm. The fitted value iteration is given in section 2.6 and the conclusion is presented in the final section.

2.2 Markov Decision Process Theory

Markov decision processes (MDP) serves as a basis for solving the more complex partially observable problems in this thesis. A MDP is a model of a decision-maker interacting synchronously with the environment. Since the decision-maker sees the environment's true state, it is referred as a *completely observable Markov decision process*. The foundation of Markov decision process is presented as follows.

2.2.1 Markov Property

Let $\{S_t\}$ be a random process where S_t is a random variable which refers to the state of the process at arbitrary time t. The sequence of S_t is a Markov process if the future of process given the present is independent of the past, that is,

$$P\left(S_{t+1} = s_{t+1} \middle| S_t = s_t, \ldots, S_0 = s_0\right) = P\left(S_{t+1} = s_{t+1} \middle| S_t = s_t\right). \tag{2.1}$$

This equation is referred to as the Markov property.

2.2.2 Markov Decision Process (MDP)

A Markov decision process (MDP) is a discrete-time random process defined by a set of states, actions and the one-step dynamics of environment. Given any state s and action a, the probability of occurrence of each possible next state s' is

$$p\left(s' \middle| s, a\right) = P\left(S_{t+1} = s' \middle| S_t = s, a_t = a\right). \tag{2.2}$$

This equation is called transition probability. Similarly, given any current state and action, s and a, together with any next state, s', the expected value of the incurred reward is

$$g(s,a,s') = E\left[g_{t+1} \middle| S_t = s, a_t = a, S_{t+1} = s' \right], \tag{2.3}$$

where $E[.]$ is the expectation operator and g_{t+1} is the reward received at time $t+1$. Equation (2.2) and (2.3), completely specify the most important aspects of the dynamics of the MDP. A MDP model can be shown in Fig. 2.1.

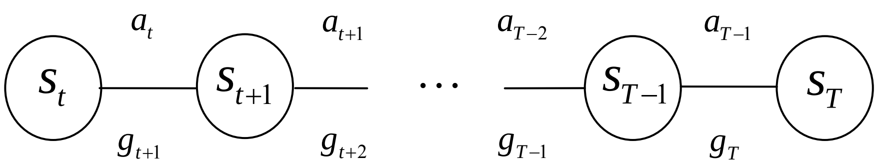

Figure 2.1 A MDP model.

2.2.3 Policy

A policy, π, is a description of the behavior of a decision-maker, or a function mapping states to actions, $\pi : S \rightarrow A$. There are two types of policies. A *stationary policy* is a situation-action mapping, i.e., it specifies an action to be taken at each state. The choice of action depends only on the state and is independent of the time step. A *non-stationary policy*, on the other hand, is a sequence of situation-action mappings, indexed by time. In this thesis, we focus on stationary policies since our data acquisition problem is based on models of sensor readings which are obtained in particular time frame, such as in the mornings, afternoons, etc. Hence, within such period, the model is stationary hence the policy is also assumed stationary.

2.2.4 Value Function

Value functions are the expected sum of rewards received from starting in state s. Value functions thus evaluate the performance of the decision which the decision-maker has taken at a given state. In other words, it quantifies how good it is to perform a given action starting in a given state. Since the rewards to be received in the future by the decision-maker depend on the actions it is willing to take, value functions are defined with respect to each particular policy. Therefore, we can define the value function of a state under a policy π, $V^{\pi}(s)$ as

$$V^{\pi}(s) = E_{\pi}\left[g_t \middle| S_t = s\right] = E_{\pi}\left[\sum_{k=1}^{\infty} \gamma^k g_{t+k} \middle| S_t = s\right], \tag{2.4}$$

where $E_{\pi}[.]$ is cthe expectation operator and γ is a discount factor[1] where $0 < \gamma < 1$.

A fundamental property of value functions is that they satisfy a particular recursive relationship. For any policy π and any state s, the following consistency condition holds between the value of s and the value of its possible successor states,

$$V^{\pi}(s) = g(s,a) + \gamma \sum_{\forall s'} p(s'|s,a) V^{\pi}(s'). \tag{2.5}$$

[1] A discount factor is used for presenting the reward in the future time step, namely a reward received k time steps in the future is worth only γ^k times in reward received immediately. If $0 < \gamma < 1$, then the total reward in finite time step has finite value on the condition that the immediate received reward is limited. If $\gamma = 0$, then the agent is myopic because agent focuses on only maximization the immediate reward (Sutton and Barto, 1998).

Equation (2.5) is called the Bellman equation for $V^{\pi}(s)$. It expresses a relationship between the value function of s and the value functions of its possible successor states s'. The Bellman equation averages over all the possibilities, weighting each successor state by its probability of occurrence. It states that the value function of the starting state (s) must equal the (discounted) value of the expected next state (s'), plus the reward expected along the way.

2.2.5 Optimal Value Function

The aim of solving a MDP is to find an optimal policy that achieves the maximum reward over the long run when starting in an arbitrary state, that is,

$$V^{*} = \max_{\pi} V^{\pi}(s), \tag{2.6}$$

for all $s \in S$.

In this thesis, we denote the optimal policy by π^{*}. Moreover, the expected rewards obtained by taking actions governed by the optimal policy is called the optimal value function, defined as

$$V^{*}(s) = \max_{a}\left[g(s,a) + \gamma \sum_{\forall s'} p(s'|s,a) V^{\pi}(s') \right], \tag{2.7}$$

for all $s \in S$.

2.3 Partially Observable Markov Decision Process Theory

Unlike the completely observable MDP, in partially observable environments, a decision-maker is no longer able to determine the state which the environment is actually in with absolute certainty. Therefore, a POMDP is a MDP in which the agent is unable to observe the actual state completely. Instead, it makes an observation based on the action and resulting state.

Consider a finite horizon POMDP with a finite set of states of the environment which is represented by $S = \{s(1), s(2)..., s(m)\}$ and whose state dynamics is governed by state transition probabilities operating under some stationary policy π, P^{π}. In a partial observability scenario, the actual state is concealed from the decision-maker. Hence, the decision-maker sees only a set of observations Θ controlled by an observation process $r : S \times A \to R$, where R is the probability distribution over Θ, and A is the set of available actions at the decision-maker. Given a current observation $\theta \in \Theta$, an action $a \in A$ is selected based on the stationary policy π. Suppose at time step t, the environment is in state s_t. However, the decision-maker sees instead an observation θ_t, and generates action a_t. As a result, the system transits to state s_{t+1}, and a new observation θ_{t+1} is observed. Then the decision-maker receives an immediate reward, $g(s, a, s', \theta)$ from the environment. For convenience, let $g(s, a)$ be the immediate reward for taking action a in state s. This term can easily be derived from the immediate reward $g(s, a, s', \theta)$ weighted by the transition and observation probabilities for the different actions and observations as follows

$$g(s,a) = \sum_{\forall s',\theta} p(s'|s,a)\, r(\theta|s',a)\, g(s,a,s',\theta). \tag{2.8}$$

In a POMDP framework, the decision-maker lacks the exact knowledge of the current state. However, the decision-maker knows the sequence of observations and actions, $H_t = \{\theta_t, a_t, \theta_{t-1}, a_{t-1}, ..., \theta_0, a_0\}$, which is called the history at time step t. Equivalently, H_t can be mapped into a belief state $b_t = [\mathrm{b}_t(1), ..., \mathrm{b}_t(m)]^T \in B$, which presents the probability that system is currently in state s at time step t by

$$\mathrm{b}_t(s) = P(s_t = s | \theta_t, ..., \theta_0; a_t, ..., a_0; b_0), \tag{2.9}$$

where $s \in S$, b_0 is the given initial distribution, and $B = \left\{ b : b \in [0,1]^m, \sum_{\forall s \in S} \mathrm{b}(s) = 1 \right\}$ is the set of all possible belief distributions over state space S. Upon taking an action $a \in A$ at each state $s \in S$, the environment transits to a new state $s' \in S$. However, in stead of a new state, the decision-maker sees a new observation, $\theta \in \Theta$. Therefore, the new belief state can be updated by

$$\mathrm{b}(s') = \frac{\sum_{\forall s} \mathrm{b}(s) p(s'|s,a) r(\theta|s',a)}{\sum_{\forall s,s'} \mathrm{b}(s) p(s'|s,a) r(\theta|s',a)}. \tag{2.10}$$

For convenience, let $\Gamma(b_t, a_t, \theta_{t+1})$ denote the short hand notation for Eq. (2.10). It is well known that, given an observable history, the sequence $\{b_t\}$ is a completely observable Markov process (Cassandra, 1995) since b_t can be recursively computed from Eq. (2.10) and relying only on b_t, a_t and θ_{t+1}. Figure 2.2 illustrates the belief changes in such MDP. The transition probability of this Markov process is given by

$$P\left(b_{t+1}\middle|b_t,a_t\right)=\sum_{\forall\theta\in\Theta|b_{t+1}=\Gamma(b_t,a_t,\theta)} r\left(\theta\middle|s',a\right)\sum_{\forall s\in S} p\left(s'\middle|s,a\right)b_t(s)$$

$$=\sum_{\forall\theta\in\Theta|b_{t+1}=\Gamma(b_t,a_t,\theta)} p\left(\theta\,|\,b_t,a_t\right). \tag{2.11}$$

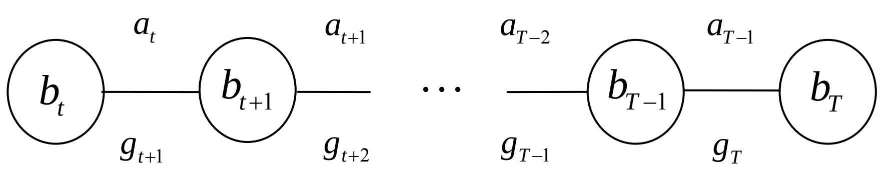

Figure 2.2 A POMDP model represented by a belief state MDP.

2.3.1 POMDP Policy

In a completely observable MDP model, a policy is a mapping from states to actions. On the other hand, for a partially observable MDP model or POMDP, the decision-maker never knows the actual state so the policies must map belief states into actions. The number of belief states is infinite even for an environment with a finite number of states. Therefore, storing the policy or value function in tables is no longer feasible. An alternative belief state representation is presented in section 2.4.

In addition, this thesis only considers formulating data acquisition in a WSN as a POMDP problem with finite horizon, or a T - horizon problem. The duration of T - time step is commonly referred to as the decision-maker's lifetime or size of the horizon. Moreover, we assume that the transition probability model is the same throughout the duration of T - time step. For example, the temperature tends to increase in the morning period and it tends to decrease during the night. Therefore, our problem is suitable for finite horizon POMDP formulation.

2.3.2 Value Function

For POMDP, a value function is the expected sum of immediate rewards, $g(b_t, a_t)$ given that the decision-maker is started off in belief state b_0 and follows policy π thereafter for a finite horizon of T steps. The POMDP value function is thus given by

$$V^{\pi}(b) = E_{\pi}\left[\sum_{t=1}^{T} g(b_t, a_t) \middle| b_0 = \mathrm{b}\right], \tag{2.12}$$

for all $\mathrm{b} \in B$, where $E_{\pi}[.]$ is the expectation operator. Similar to the MDP, the Bellman's equation for Eq. (2.12) can be written by

$$V_t^{\pi}(b) = \sum_{\forall s} \mathrm{b}(s)\left[g(s,a) + \sum_{\forall s',\theta} p(s'|s,a) r(\theta|s',a) V_{t-1}(\mathrm{b}(s'))\right]. \tag{2.13}$$

2.3.3 Optimal Value function

In order to solve the POMDP, we must find the optimal policy which maps the belief states to action as shown in Fig. 2.2. The optimal policy is a policy which generates the maximum value function for all belief states such that

$$V_t^{*}(b) = \max_a \left[\sum_{\forall s} \mathrm{b}(s)\left[g(s,a) + \sum_{\forall s',\theta} p(s'|s,a) r(\theta|s',a) V_{t-1}^{*}(\mathrm{b}(s'))\right]\right]. \tag{2.14}$$

In addition, Eq. (2.14) can be rewritten to show that the value function is can be defined as a combination of linear segments. Let $\alpha_t(s)$ be the optimal value function at time step t given that the starting state is s, which is defined as follows

$$\alpha_t(s) = g(s,a) + \sum_{\forall s',\theta} p(s' \mid s,a) r(\theta \mid s',a) \alpha_{t-1}^{l(b,a,\theta)}(s'), \tag{2.15}$$

where $l(b,a,\theta)$ is an indcexing function which identifies b,a,θ of the α-vector that maximizes $V_{t-1}^{*}(\mathrm{b}(s'))$. That is,

$$l(b,a,\theta) = \arg\max_{k} \left[\sum_{\forall s,s'} \mathrm{b}(s) p(s' | s,a) r(\theta | s',a) \alpha_{t-1}^{k}(s') \right]. \tag{2.16}$$

Therefore, Eq. (2.13) can be simplified as

$$V_t^{*}(b) = \max_{a} \sum_{\forall s} \mathrm{b}(s) \alpha_t(s), \tag{2.17}$$

where $\alpha_t(s)$ is a set of vectors $\alpha_t(s) = \{\alpha^0(s), \alpha^1(s), ..., \alpha^k(s)\}$. Note that $\alpha^k(s)$ can be viewed as one of the *linear pieces* of $V_t^{*}(b)$. According to Eq. (2.17), it can be seen that the optimal value function is still piecewise-linear and convex (PWLC) as shown in Fig. 2.3.

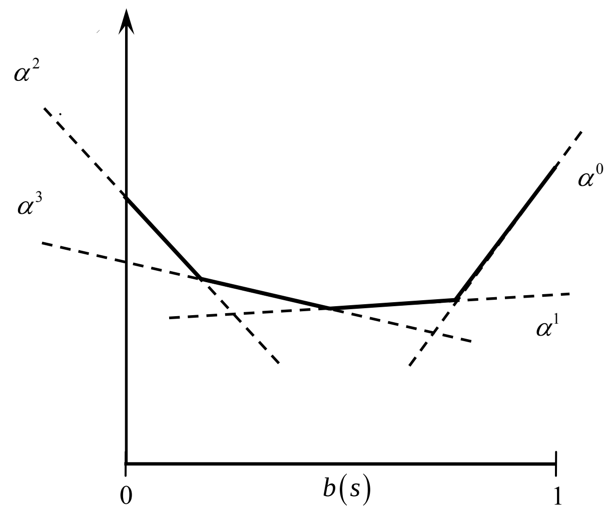

Figure 2.3 Simple piecewise linear and convex the optimal value function for $|S| = 2$.

Figure 2.3 shows the optimal value function for a two-dimensional belief state POMDP ($|S| = 2$). The optimal value function is defined as four in a plane. That is, it consists of four vectors. Since the optimal value function takes a maximum dot product of this vector and the belief state as shown in Eq. (2.17), the optimal value function is hence the upper-most line at each belief state. To solve for the optimal policy, we employ the witness algorithm (Cassandra, 1995) which is presented in section 2.5.

2.4 Parametric Partially Observable Markov Decision Process Theory

A parametric partially observable Markov decision process is a POMDP which represents a belief state in a *parametric form* (Brooks, Makarenko, Williams and D-Whyte, 2006). As described in section 2.3, the state space in a general POMDP is discretized. By partitioning the continuous state space into a finite set of discrete states, the belief states can be represented by a finite dimension probability vectors, $b \in [0,1]^{|S|}$, where $|S|$ is the cardinality of the discretized state space. Note that $\mathrm{b}(s)$ is an element of vector b which represents the decision-maker's belief that the environment is in state s. In a realistic scenario, however, most sensor readings are real-valued. The main drawback of discrete-state POMDP formulation of continuous state space is that the computational burden increases rapidly with the number of fine discretization.

Intuitively, the difficulty of solving POMDP exactly is due to the "curse of dimensionality". For a discrete POMDP, the belief state $b \in B$ has dimensionality equal to the size of state space, $|S|$. Therefore, $|b|$ grows exponentially with $|S|$ and as a consequence, the POMDP computational complexity increases rapidly with the dimensionality of the belief space.

An alternative is to assume a parametric form to represent belief state with a relatively small numbers of parameters. These parameters are called *sufficient statistics*. A requirement for the use of a parametric form representation for the distribution b_t is that the parametric form must be able to accurately approximate the belief transition function $\Gamma(b_t, a_t, \theta_{t+1})$, or at least a close approximation to it.

Therefore, an efficient analytic belief transition function which can be evaluated directly in the space of sufficient statistics should be employed.

To incorporate the parametric representation into the framework, the probability distribution over a continuous state space, b_t, is written in terms of a vector of sufficient statistics Φ_t which can be updated by a belief update function

$$\Phi_{t+1} = \tilde{\Gamma}\left(\Phi_t, a_t, \theta_{t+1}\right), \tag{2.18}$$

where $\tilde{\Gamma}$ is an approximate belief update function.

The immediate reward and the distribution over subsequent observations are also functions of the sufficient statistics, respectively defined by

$$\tilde{g}\left(\Phi_t, a_t\right) = \int g\left(s_t, a_t\right) P\left(s_t \middle| \Phi_t\right) ds_t, \tag{2.19}$$

$$P\left(\theta_{t+1} \middle| \Phi_t, a_t\right) = \int P\left(\theta_{t+1} \middle| s_{t+1}\right) P\left(s_{t+1} \middle| \Phi_t, a_t\right) ds_{t+1}. \tag{2.20}$$

2.4.1 Optimal Value Function for PPOMDP

In PPOMDP, the definition of value function is similar to value function for POMDP. However, unlike the general POMDP, the parametric representation of a belief state in PPOMDP is based on a vector of sufficient statistics, Φ_t. Therefore, given an appropriate belief update function $\tilde{\Gamma}$, the value function in terms of sufficient statistics Φ_t can then be written as

$$\tilde{V}^*\left(\Phi_t\right) = \max_{a_t}\left[\tilde{g}\left(\Phi_t, a_t\right) + \gamma \underset{\theta_{t+1}}{E}\left[\tilde{V}^*\left(\tilde{\Gamma}\left(\Phi_t, a_t, \theta_{t+1}\right)\right)\right]\right], \tag{2.21}$$

where the $\underset{\theta_{t+1}}{E}[.]$ the expectation over all observations which can be calculated using Eq. (2.20).

To solve for the optimal policy, we employ the fitted value iteration (Brooks, Makarenko, Williams and D-Whyte, 2006) which is presented in section 2.6.

2.5 Witness Algorithm for Solving POMDP

This section presents an exact algorithm, called the witness algorithm (WIT) (Cassandra, 1995) for performing value iteration[2] (Sutton and Barto, 1998) in POMDP problems described in section 2.3.

The basic concept of all algorithms for solving POMDP starts with a set of vectors, $\nu_{t-1}^{*} = \left\{\alpha_{t-1}^{0}, \alpha_{t-1}^{1}, ..., \alpha_{t-1}^{M-1}\right\}$, where M is a number of piecewise linear vectors in the set ν_{t-1}^{*}. Such set includes the piecewise linear representation for the optimal value function at the $t-1$ th time step where

$$V_{t-1}^{*}(b) = \max_{k} \; b \cdot \alpha_{t-1}^{k}, \tag{2.22}$$

and α_{t-1}^{k} is the k^{th} component of the set of piecewise linear vectors, ν_{t-1}^{*}, at the $(t-1)^{\text{th}}$ time step.

---

[2] Value Iteration is basic algorithm used to find an optimal value function through a simple iterative process that can be shown to converge to the correct V^{*} values.

The output of the algorithm will be a set of vectors $v_t^* = \{\alpha_t^0, \alpha_t^1, ..., \alpha_t^{N-1}\}$, representing the optimal value function, $V_t^*(b)$, for the t^{th} time step where N is a possible number of piecewise linear vectors in set v_t^*. Note that N can be found by determining the linear segments which satisfy Eq. (2.22).

Define an alternate set of vectors, $\Omega_t^a = \{\alpha_t^0(a), \alpha_t^1(a), ..., \alpha_t^{N-1}(a)\}$. This set of vectors represent the piecewise linear action-value function, $Q_t^a(b)$, for performing action a at time t and performing optimally thereafter. The value function $V_{t-1}^*(b)$ gives us the value for performing optimally thereafter. We will need a separate Ω_t^a set of vectors for each action, a. This value function can be specified in terms of immediate rewards and previous $\alpha_{t-1}^k \in v_{t-1}^*$ vector as

$$Q_t^a(b) = \sum_{\forall s} \mathrm{b}(s) g(s,a) + \gamma \sum_{\forall s,s',\theta} \mathrm{b}(s) p(s'|s,a) r(\theta|s',a) \alpha_{t-1}^{l(b,a,\theta)}(s'), \tag{2.23}$$

where $l(b,a,\theta) = \arg\max_k \left[\sum_{\forall s,s'} \mathrm{b}(s) p(s'|s,a) r(\theta|s',a) \alpha_{t-1}^k(s') \right]$.

According to these equations, we can present the s^{th} component of a vector in Ω_t^a for given a belief b by

$$\alpha_t^k(s,a) = g(s,a) + \gamma \sum_{\forall s',\theta} p(s'|s,a) r(\theta|s',a) \alpha_{t-1}^{l(b,a,\theta)}(s'). \tag{2.24}$$

This is exactly analogous to generating a vector for v_t^* given a particular belief state except that we do not need to maximize over the actions since a is fixed for each set Ω_t^a.

The witness algorithm will first construct the Ω_t^a sets and then determine the desired v_t^* set of vectors from Ω_t^a. In constructing a particular set of Ω_t^a, we will incrementally build up to the full set by successive approximations, $\tilde{\Omega}_t^a$. The approximations are such that $\tilde{\Omega}_t^a \subseteq \Omega_t^a$ at all times. Note also that $v_t^* \subseteq \bigcup_a \Omega_t^a$, since the value function $V_t^*(b)$ is the same as the action-value function $Q_t^a(b)$ at a belief state when the action a is optimal at that point. Since we have accounted for all possible actions, no matter what the optimal action for a point is, the corresponding vector will be in $\bigcup_a \Omega_t^a$.

The pseudo-code for constructing Ω_t^a of the witness algorithm is presented in algorithm1.

Algorithm1 Witness algorithm

---

(1) Select a belief state, $b \in B$ and an action, $a \in A$.

(2) Initialize $\tilde{\Omega}_t^a$ with a vector generated from any belief point and mark it.

(3) Select a marked vector, $\tilde{\alpha}_t(a)$ from $\tilde{\Omega}_t^a$. If there are none, then we are done and $\tilde{\Omega}_t^a = \Omega_t^a$.

(4) For each possible combination of $\alpha_{t-1}^k \in v_{t-1}^*$ and $\theta \in \Theta$

(i) Construct and solve an LP with $\tilde{\alpha}_t(a)$, α_{t-1}^k and θ (LP is shown below).

(ii) If $\lambda > 0$, then construct a vector from solution point using equation (2.24), add it to $\tilde{\Omega}_t^a$, mark it, and repeat (i) for this same combination.

(iii) If $\lambda \le 0$, then move on the next combination of α_{t-1}^k and θ.

(5) Unmark $\tilde{\alpha}_t(a)$.

(6) Go to (3).

Each LP will have the following form:

$$\max : \lambda$$

$$s.t. \quad \sum_{\forall s} \mathrm{b}(s)\tilde{\alpha}_t(s,a) \ge \sum_{\forall s} \mathrm{b}(s)\tilde{\alpha}_t^l(s,a), \quad \forall \alpha_t^l(a) \in \tilde{Q}_t^a$$

$$\sum_{\forall s,s'} \mathrm{b}(s)\, p(s'|s,a)\, r(\theta|s',a)\left(\alpha_{t-1}^k - \alpha_{t-1}^{i(\theta)}\right) \ge \lambda$$

$$\sum_{\forall s} \mathrm{b}(s) = 1$$

$$\mathrm{b}(s) \ge 0, \quad \forall s$$

2.6 Fitted Value Iteration for Solving Parametric POMDP

This section provides an approximate algorithm, called fitted value iteration (FVI) (Brooks, Makarenko, Williams and D-Whyte, 2006), which is used to solve for a solution to the PPOMDP described in section 2.4. FVI is an approach for solving MDPs with very large or infinite numbers of states. A POMDP can be viewed as a continuous belief-state MDP. Therefore, FVI can be applied to this resultant (infinite-state) MDP. The basic concept of FVI is to store value functions explicitly at only a relatively small number of belief points, using a function approximator to approximate the value function for all belief points in between (see **Appendix I**). In principle, the value at one state may give no information about the value at another state. However, if the value function is sufficiently smooth and enough values are stored explicitly, FVI is likely to provide a good approximation. At each subsequent time-step, a new set of explicit stored values can be estimated from the approximate value function of the previous time-step.

More formally, let β be a predefined set of belief points in sufficient statistics space of size $|\beta|$. In particular, let $\beta = \left\{\Phi_1, \Phi_2, ..., \Phi_{|\beta|}\right\}$ and Ψ be the set of explicit belief point-value pairs

$$\Psi = \left\{\left(\Phi_1, \psi\left(\Phi_1\right)\right), \left(\Phi_2, \psi\left(\Phi_2\right)\right), ..., \left(\Phi_{|\beta|}, \psi\left(\Phi_{|\beta|}\right)\right)\right\}, \tag{2.25}$$

where $\psi(\Phi_i)$ is the estimated value function of the i^{th} belief point in set β. Let $\hat{V}_t(\Phi)$ denote the current estimated value function of belief point $\Phi \in \beta$. Note that $\hat{V}_t(\Phi)$ can be estimated using a function approximator[3], F_β, based on the set Ψ.

$$\hat{V}_t(\Phi) = F_\beta(\Phi, \Psi_t). \tag{2.26}$$

Equation (2.26) and the value function of the i^{th} belief point in β are used to estimate the value function at any point in sufficient statistics space. The value functions associated with each belief point in β can therefore be updated by using

$$\psi_t(\Phi_i) = \max_{a_t}\left[\tilde{g}(\Phi_i, a_t) + \gamma \underset{\theta_{t+1}}{E}\left[\hat{V}_{t+1}\left(\tilde{\Gamma}(\Phi, a_t, \theta_{t+1})\right)\right]\right]. \tag{2.27}$$

A pseudo-code of the FVI method is presented in Algorithm2. Fitted value iteration for PPOMDP is guaranteed to converge provided that the function approximator is not an expansion in the max (Gordon, 1995). Note that the function approximator is one which estimates the value of a point as a weighted sum of the values of nearby points. The convergence criterion in step 4 of algorithm2 is defined by one acceptable criterion, that is, when the maximum change in the value of a belief point in β is less than a predefined threshold.

---

[3] Freudenthal triangulation is the function approximator scheme applied in this thesis. More details can be found in **Appendix I**.

Algorithm2 Fitted value iteration

Select a set of belief points β

$t \leftarrow MAX\_ITERATIONS$

initialize $\psi_t(\Phi_i)$ to zero, $\forall i \in 1,...,|\beta|$

while not converged

    $t \leftarrow t-1$

    for each $\Phi \in \beta$

        for each $a_t \in A$

            calculate $\rho_{a_t} \leftarrow \tilde{g}(\Phi, a_t)$

            $\nu_{a_t} \leftarrow 0$

            for each $\theta_{t+1} \in \Theta$

                calculate the likelihood $l \leftarrow P(\theta_{t+1} | \Phi, a_t)$

                calculate $\Phi' \leftarrow \tilde{\Gamma}(\Phi, a_t, \theta_{t+1})$

                apply the function approximator F_β to find $\nu \leftarrow \hat{V}_{t+1}(\Phi')$

                $\nu_{a_t} \leftarrow \nu_{a_t} + l\nu$

            end for each θ_{t+1}

        end for each a_t

        $\psi_t(\Phi) \leftarrow \max_{a_t}\left[\rho_{a_t} + \gamma \nu_{a_t}\right]$

    end for each Φ

end while

2.7 Conclusions

In this chapter, an overview of the discrete-state partially observable Markov decision process (POMDP) concept is given. The concept of continuous-state partially observable Markov decision process called parametric POMDP (PPOMDP) is also introduced. The POMDP framework has been used to formulate data acquisition problem in error prone wireless sensor networks. To solve the discrete-state POMDP framework, the witness algorithm (WIT) has been introduced to provide an exact analytical solution. However, as most sensor readings are real-valued, their computational burden increases rapidly with fine state discretization. Therefore, the fitted value iteration has been presented to provide a good solution for a continuous-state POMDP.

In the next chapter, a discrete-state POMDP formulation of the data acquisition problem while guaranteeing collected data quality in error-prone wireless sensor networks is presented. The performance of the solution obtained by means of the witness algorithm is also evaluated.

CHAPTER III

A POMDP FRAMEWORK FOR DATA ACQUISITION IN WSNs : WITNESS ALGORITHM

3.1 Introduction

This chapter studies the data acquisition problem which supports probabilistic confidence requirements of the acquired data in wireless sensor networks (WSNs). A wireless sensor network (WSN) consists of a large number of redundant wireless sensor nodes which cooperatively monitor physical information of the environment (such as temperature, pressure or motion, at different locations) which typically exhibit high spatial and temporal correlation. Acquiring as much data as possible from the environment at a given point in time may result in high communication costs. Furthermore, most of the data may provide little benefit to the quality of query results. WSNs, in particular, have limited communication, computation processing power and battery life. Hence, a data acquisition scheme which efficiently collects sensor readings with low resource consumption is needed.

Recently, there have been several researches which have investigated efficient data acquisition in an error-prone wireless sensor network to provide good quality query results as well as save energy consumption (Deshpande et al. (2004), Liu et al. (2007), Meliou et al. (2007)). Deshpande et al. (2004) and Liu et al. (2007) studied the quality of the collected data and resource consumption in the short term by using spatial and temporal correlation models. However, data acquisition is determined independently in each time step (i.e. short-term). Therefore, the data acquisition plan may be optimal in the immediate time, but may suboptimal in the long run. On the other hand, Meliou et al. (2007) minimized the long-term data acquisition cost by using spatial and temporal correlation models. In particular, their proposed scheme outperforms the short-term scheme. However, the quality of the collected data has not been explicitly guaranteed in their work.

Therefore, the underlying aim of this chapter is to optimize the long-term performance criterion in data collection while maintaining acceptable quality of data under error-prone WSNs. We formulate the problem as a partially observable Markov decision process (POMDP) (Chobsri and Usaha, 2008).

A fundamental assumption is that the Markov property of the sensor attribute readings must hold. The assumption has been verified by statistical tests performed on sequences of attribute sensor readings generated by various Gaussian distributions (see **Appendix II**). Based on our testing, the probability of occurrence of future states in the process given the present state is equal to the probability of occurrence of future states in the process given all previous states. Hence, Markov property is assumed to hold.

The other general assumptions of the POMDP framework in this chapter are as follows.

1) The base station (BS) maintains a spatial and temporal correlation model of the physical information of the environment.

2) The sensor readings received at the BS have both spatial and temporal correlation.

3) The decision to acquire sensor readings (or not) occurs only at the BS, and as a result, the spatial and temporal correlation model change.

4) WSN has limited resource and is error-prone.

To solve this problem, an existing tool called the witness algorithm is applied to find a good sensor selection plan to acquire data such that the quality assurance requirements on the sensor readings are still satisfied.

The contribution in this chapter is therefore twofold, i) the mapping of the data acquisition problem into a POMDP framework, ii) applying the witness algorithm to solve it.

This chapter is organized as follows. The overview of the network architecture and data acquisition process will be described in section 3.2. Section 3.3 provides the formulation of data acquisition problem in WSNs as a POMDP. In section 3.4, the numerical results will be presented and, finally, section 3.5 summarizes the entire chapter.

3.2 System Overview

The data acquisition system consists of a base station which includes declarative query processing engine and a number of sensor nodes that acquire information about the surroundings of the wireless sensor network.

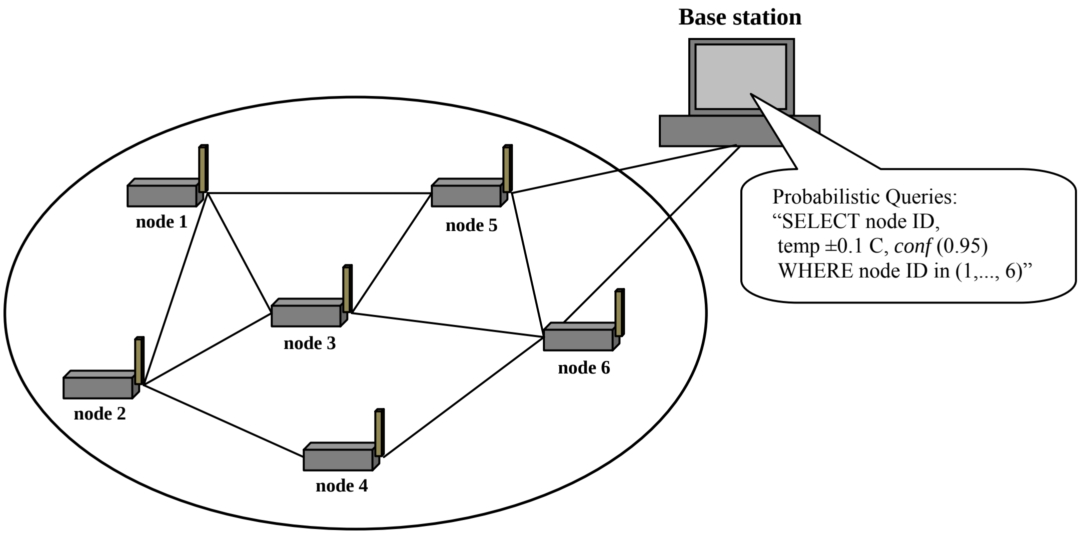

Figure 3.1 Structure for data acquisition in WSNs.

Fig. 3.1 illustrates the data acquisition process in a wireless sensor network of 6 nodes. At first, a user sends a probabilistic query[4] which specifies a required confidence bound on the query result. For example, a user sends a query to the BS for the temperature value of the environment in a WSN. The queries include an error tolerance of 0.1°C and a confidence bound of 95% may be submitted to a group of sensors. This bound specifies how much uncertainty the user is willing to take. Based on the knowledge of the network, the base station must then decide

[4] A query and query answer which include a probability based on a probabilistic model in the database (Wuthrich, 1994 and Deshpande et al., 2004).

how to answer this query. It could collect data from most sensors which could result in high energy and processing requirements. Furthermore, most data may even be redundant and have little benefit to the query result. Alternatively, based on the knowledge of the sensor network, the base station may choose data selectively from certain sensors with high quality data to reduce resource consumption. As onboard resource on a sensor node is highly scarce, most of the recent research works place emphasis on the latter approach.

Deshpande et al. (2004) proposed to use the knowledge of the sensor network in the terms of the spatial and temporal correlations of the sensor readings. An example of the relation between sensor readings in a WSN is shown in Fig. 3.2 which presents a trace of voltage and temperature readings over a 2-day period from two sensors. Notice the close correlations between the two attributes Such behavior can be described by correlation models. An important usage of the correlation model is that the BS can exploit correlations between sensor readings to estimate a required sensor reading when the BS is sufficiently confident that its estimation can satisfy the confidence requirement of the sensor reading. As a result, the BS may be able to answer the user's query with high information quality without resorting to actually querying any sensor at all. Thus, sensor nodes can reduce resource consumption. In this thesis, we therefore apply the correlation model to solve the data acquisition problem to support the quality of query result and also reduce resource consumption.

3.2.1 The Correlation Models

To describe the relation between the received attributes at the BS, let X_i, $i = 1,...,n$, be a random variable for attribute i in the WSNs. Each X_i is an attribute at a particular sensor, e.g., temperature on sensing node, etc. There is typically one attribute per sensing type per sensor node. Therefore, X_i is referred to a random variable for an attribute at sensor i in this thesis. We denote a spatial correlation model by a probability density function (pdf), $p(X_1, X_2,..., X_n)$, which assigns a probability value for each joint value $x_1, x_2,..., x_n$ for attributes $X_1, X_2,..., X_n$. This joint probability density function quantifies the relation between each attribute.

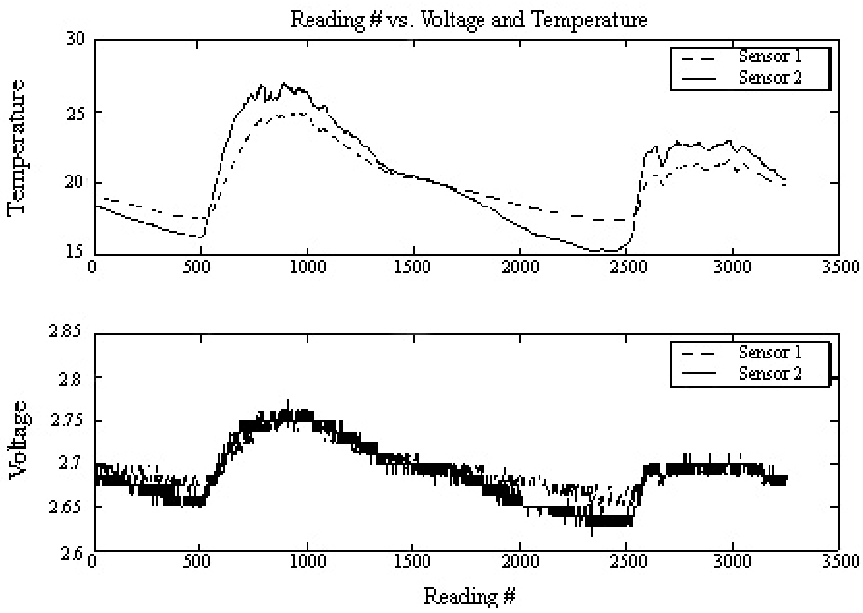

Figure 3.2 Trace of voltage and temperature readings over a two day period from Mica2 sensor motes (Deshpande et al., 2004).

In this thesis, we use a specific model based on *a time-varying multivariate*[5] Gaussian distribution to capture the spatial and temporal correlation model. A Gaussian pdf over attributes from *n* sensors, $X_1, X_2, ..., X_n$ can be viewed as a function of two parameters, that is a *n*-dimensional vector of means $(\vec{\mu})$ and a $n \times n$ covariance matrix $(\mathbb{C})$ as follows

$$p(X_1, X_2, ..., X_n) = \frac{\exp\left\{-\frac{1}{2}(\vec{X} - \vec{\mu})^T \mathbb{C}^{-1} (\vec{X} - \vec{\mu})\right\}}{(2\pi)^{n/2} |\mathbb{C}|^{1/2}}, \quad (3.1)$$

where $\vec{X} = [x_1, x_2, ..., x_n]$, $\vec{\mu} = [\mu_1, \mu_2, ..., \mu_n]$ and

$$\mathbb{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{n1} & \cdots & & c_{nn} \end{bmatrix}. \quad (3.2)$$

The mean vector $(\vec{\mu})$ is the center of this probability distribution and the covariance matrix $(\mathbb{C})$ presents the spread of the distribution which is a measure of correlation between attributes.

---

[5] The word of "time-varying" is used to demonstrate the relation in term of time. In addition, the word of "multivariate" is used to present the relation in term of space.

3.2.2 Probabilistic Queries

Based on a correlation model described above, the BS can use this model to estimate answers to queries when the query probability exceeds the user's specified confidence threshold. This section reviews the methodology for using a correlation model to answer the user's query.

Suppose that a user generates a range query for a sensor reading with an error bound of ε to sensor *i*, X_i with confidence at least δ. A user sends such query to determine whether the probability that X_i is in a certain range meets the confidence requirement, i.e. $P\left(X_i \in \left[x_i - \varepsilon, x_i + \varepsilon\right]\right) \geq \delta$. Hence, the probability that X_i is in an arbitrary range $P\left(X_i \in \left[L_i, U_i\right]\right)$ can be computed by

$$P\left(X_i \in \left[L_i, U_i\right]\right) = \int_{L_i}^{U_i} p\left(x_i\right) dx_i, \tag{3.3}$$

where $p\left(x_i\right) = \int p\left(x_1, \ldots, x_n\right) dx_1 \ldots dx_{i-1} dx_{i+1} \ldots dx_n$ is the marginal probability of attribute X_i being in the arbitrary range $\left[L_i, U_i\right]$. Hence, the correlation model can be used to compute the confidence of the sensor reading. The higher the probability, the higher the confidence of the attribute being in that range.

In addition to range queries, a correlation model can be used to answer the other types of queries. For example, value queries ask the value of a particular attribute on sensor *i*. Such query can be answered by computing the mean value of such attribute on sensor *i* as follows

$$\mu_i = \int x_i p\left(x_i \middle| \theta\right) dx_i, \tag{3.4}$$

where $p(x_i|\theta)$ is the posterior probability of sensor i given observation θ. The confidence level is given by $P(X_i \in [x_i - \varepsilon, x_i + \varepsilon]|\theta)$ which can be determined from the correlation model conditioned by the occurrence of observation θ.

3.2.3 Energy Consumption Model

The energy consumption model on each sensor is obtained from the data sheets of sensors and the radio used on Mica2 Motes with a Crossbow MTS400 environmental sensor board as shown in table 3.1. For our system, we assume that the sender and receiver are well synchronized so that the sender and receiver simultaneously turn on the radio[6]. The Mica2 Motes consume about 0.4 mJ of energy in both sending and receiving modes. In the data acquisition process, a user submits a query for a specific attribute from the environment. The total energy consumption for acquiring data that transits a number of n_t nodes and acquires a number of sensor readings from n_a nodes can be calculated as follows

$$\text{total energy consumption per attribute} = (E_s \times n_t) + (E_r \times n_t) + (E_{att} \times n_a), \qquad (3.5)$$

where E_s is the energy consumed in the sending mode, E_r is the energy consumed in the receiving mode and E_{att} is the consumed energy for sampling a specific attribute.

[6] In practice, the receiver turns on periodically to sample the radio channel and detect a sender's signal to begin transmission (Polastre, 2003). Though this periodic radio sampling uses some energy, it is small because the sampling duty cycle can be 1% or less.

In this thesis, we primarily focus on the temperature readings since the temperatures have both temporal and spatial correlations. The temporal correlation of temperature characterizes a dynamic correlation model. Therefore, the temperature values observed earlier in time should assist in estimating the temperature values later in time. This temporal characterization is referred to the *Markov property*. However, our framework can be applied to other readings such as pressure, humidity, etc., which exhibit a certain degree of Markov property.

Table 3.1 Summary of Energy Requirement of Crossbow MTS400 Sensor Board at 3 Volts of Battery Level (Deshpande et al., 2004).

Sensor	Energy per sample (@3V),mJ
Solar Radiation	0.525
Barometric Pressure	0.003
Humidity and Temperature	0.5
Voltage	0.00009

3.3 Partially Observable Markov Decision Process Formulation

This section presents the formulation of data acquisition problem in an error-prone WSN as a POMDP. The general assumptions of the POMDP framework in this chapter are as follows.

1) The sensor readings of the attributes received at the base station (BS), $X_1, X_2, ..., X_n$, have both spatial and temporal correlation.

2) The BS maintains a spatial and temporal correlation model, $p\left(X_1^t, X_2^t, ..., X_n^t\right)$, of the physical information of the environment.

3) The decision for data acquisition occurs only at the BS. As a result, the spatial and temporal correlation model changes to $p\left(X_1^{t+1}, X_2^{t+1}, \ldots, X_n^{t+1}\right)$, according to new sensor readings $X_1^{t+1}, X_2^{t+1}, \ldots, X_n^{t+1}$ received at the BS.

4) WSN has limited resource and is error-prone.

The components of the POMDP framework in this chapter are :

3.3.1 State Space

Let S be a finite set of states which specifies all possible attribute values received at the BS from all sensor nodes in the network. Let $s \in S$ represent a vector of received attribute values at the BS. For example, consider a WSN of 3 nodes, where each sensor node can collect a single attribute (i.e. temperature). At the BS, the temperature values are 24.8, 26.7 and 28.2 °C on sensor node 1, 2, and 3 respectively. Therefore, the state of the BS is [24.8, 26.7, 28.2]. Notice that most attribute values are continuous-valued. For a discrete-state POMDP framework, quantization is needed to discretize the state space.

Let q be the number of the quantization levels. For this example, suppose that temperature values have possible values ranging from 0-100 °C. Suppose that the range is quantized into 2 subintervals, i.e. [0,50) and [50,100] °C which can be represented by integers 0 and 1, respectively. Thus, the state of the BS is transformed to [0, 0, 0] and the size of state space is 8. For an arbitrary WSN with n nodes and q quantization levels, the size of the state space of all the possible attribute values at the BS is q^n. Define an arbitrary state $s(j)$ by

$$s(j) = \left(X_1 \in [c_1, d_1], X_2 \in [c_2, d_2], \ldots, X_n \in [c_n, d_n]\right), \tag{3.6}$$

where $j = 1,2,...,q^n$, c_j is the lower bound of the j^{th} state at sensor node i and d_j is the upper bound of the j^{th} state at sensor node i.

3.3.2 Belief State Space

Let $\mathrm{b}\left(s\left(j\right)\right)$ be the belief state which represents the probability that the attribute values received at the BS are currently in state $s\left(j\right)$. The belief state, $\mathrm{b}\left(s\left(j\right)\right)$ where $j = 1,2,...,q^n$, can be computed from the correlation model as follows

$$\mathrm{b}\left(s\left(j\right)\right) = P\left(X_1 \in \left[c_1, d_1\right], X_2 \in \left[c_2, d_2\right], ..., X_n \in \left[c_n, d_n\right]\right)$$

$$= \int_{c_n}^{d_n} \int_{c_{n-1}}^{d_{n-1}} ... \int_{c_1}^{d_1} p\left(x_1, x_2, ..., x_n\right) dx_1 dx_2 ... dx_n. \tag{3.7}$$

Therefore, a belief state vector represents the probability distribution over all possible states which can be written by

$$b = \left[\mathrm{b}\left(s(1)\right), ..., \mathrm{b}\left(s\left(q^n\right)\right)\right], \tag{3.8}$$

where $b \in B$ and $B = \left\{ b : b = \left[0,1\right]^{q^n}, \sum_{j=1}^{q^n} \mathrm{b}\left(s\left(j\right)\right) = 1 \right\}$ is a finite set of belief states.

3.3.3 Action Space

For each user query, a decision is made at the BS whether it is to estimate data from a correlation model or acquire data from a number of specific sensor node(s) in a WSN. The decision is referred to an action. The action space, denoted by A, is the set of all possible actions which is defined as follows

$$A = \{a : a = [\mathrm{a}_1, \mathrm{a}_2, ..., \mathrm{a}_n], \mathrm{a}_i \in \{0,1\}, i = 1,2,...,n\}, \tag{3.9}$$

where a_i refers to the action related to the attribute from sensor node i such that

$$\mathrm{a}_i = \begin{cases} 1 & \text{, if the BS acquires an attribute from sensor node } i, \\ 0 & \text{, otherwise.} \end{cases} \tag{3.10}$$

In other words, action $\mathrm{a}_i = 0$ is the action which the BS estimates data from the correlation model; $\mathrm{a}_i = 1$ refers to the action of the BS acquiring data by submitting a query to sensor node i.

3.3.4 Observation Space

Due to the possibility that a sensor may give faulty values, the BS may be unable to see actual state or receive actual state of the environment. This could be caused by faulty communication between sensors, sediments from the surroundings which may have been caught on the sensors through rain, wind, or animals passing by, or faulty sensors themselves, etc. Therefore, the BS sees just an observation, $\theta = s \pm \vec{e}$, instead the actual state (s) of the environment, where $\theta \in \Theta$, $\vec{e}$ is the q^n-dimensional vector of the observation error and Θ is the set of all possible observations. In particular,

$$\Theta=\left\{\theta:\theta=\left[\theta_1,\theta_2,...,\theta_{q^n}\right],\theta_j\in\left[c_j,d_j\right],j=1,2,...,q^n\right\}. \tag{3.11}$$

where $j=1,2,...,q^n$, c_j is the lower bound in the j^{th} state at sensor node *i*, d_j is the upper bound in the j^{th} state at sensor node *i*.

3.3.5 Transition Probability Model

Upon taking an action $a\in A$ in state $s\in S$, the actual received data at the BS transits to a new state $s'\in S$ according to the transition probability matrix *P*. Note that $P=\left[p\left(s'|s,a\right)\right]$ defines the Markov process that the BS operates within, where $p\left(s'|s,a\right)$ is the probability that the system transits to state s' given that it is currently in state *s* and action *a* was just taken. More specifically, let *s* be the current attribute values received at the BS. The BS then decides to take an action, $a\in A$, which could be estimating attribute values from the correlation model, or acquiring new sensor readings from a particular set of sensors. As a result, the BS receives a new actual attribute$\left(s'\right)$. Notice that the one-step state transition probabilities are fixed by an action and do not change with time[7]. Such transition probability is given by

$$p\left(s'|s,a\right)=P\left(X_1^{t+1},...,X_n^{t+1}\middle|X_1^t,...,X_n^t,a_t\right). \tag{3.12}$$

---

[7] This is referred to the homogeneous transition probability.

These quantities can be found from historical data in two simple steps (Deshpande et al., 2004). First, we estimate a mean vector $(\bar{\mu})$ and a covariance matrix $(\mathbb{C})$ for the joint density $P\left(X_1^{t+1},...,X_n^{t+1},X_1^t,...,X_n^t,a_t\right)$. That is, we form tuples $\left\langle X_1^{t+1},...,X_n^{t+1},X_1^t,...,X_n^t,a_t\right\rangle$ for the sensors at every consecutive time step $t+1$ and t. We then use these tuples to compute the joint mean vector and covariance matrix, i.e. at time step t+1,

$$\bar{\mu}_{t+1}=\left[\mu_1^{t+1},\mu_2^{t+1},...,\mu_n^{t+1}\right],\ \mathbb{C}_{t+1}=\begin{bmatrix} c_{11}^{t+1} & c_{12}^{t+1} & \cdots & c_{1n}^{t+1} \\ c_{21}^{t+1} & c_{22}^{t+1} & \cdots & c_{2n}^{t+1} \\ \vdots & \vdots & & \vdots \\ c_{n1}^{t+1} & \cdots & & c_{nn}^{t+1} \end{bmatrix},$$

and at time step t,

$$\bar{\mu}_{t}=\left[\mu_1^{t},\mu_2^{t},...,\mu_n^{t}\right],\ \mathbb{C}_{t}=\begin{bmatrix} c_{11}^{t} & c_{12}^{t} & \cdots & c_{1n}^{t} \\ c_{21}^{t} & c_{22}^{t} & \cdots & c_{2n}^{t} \\ \vdots & \vdots & & \vdots \\ c_{n1}^{t} & \cdots & & c_{nn}^{t} \end{bmatrix}.$$

We then use the conditional probability[8] to compute the transition model:

$$P\left(X_1^{t+1},...,X_n^{t+1}\middle|X_1^t,...,X_n^t,a_t\right)=\frac{P\left(X_1^{t+1},...,X_n^{t+1},X_1^t,...,X_n^t,a_t\right)}{P\left(X_1^t,...,X_n^t,a_t\right)}. \tag{3.13}$$

[8] The conditional probability is defined by $P\left(A\middle|B\right)=\dfrac{P\left(A\cap B\right)}{P\left(B\right)}$ for $P\left(B\right)>0$.

3.3.6 Observation Model

As the sensors may be error prone and give inaccurate readings, the BS may not have access to the core process in Eq. (3.13). Therefore, to model such inaccuracy, let $R = \left[r\left(\theta \middle| s',a \right) \right]$ be the observation model where $r\left(\theta \middle| s',a \right)$ denotes the probability that θ is observed at the BS when, in fact, the actual attributes are in state $s \in S$ at time t and the action just taken (at time $t-1$) was $a \in A$. Note that $r\left(\theta \middle| s',a \right)$ can be found by forming the tuple $\left\langle \theta_1^t,...,\theta_n^t, X_1^t,...,X_n^t, a_{t\text{-}1} \right\rangle$ and then use the conditional probability to compute the observation probability $r\left(\theta \middle| s',a \right)$ by

$$P\left(\theta_1^t,...,\theta_n^t \middle| X_1^t,...,X_n^t, a_{t-1} \right) = \frac{P\left(\theta_1^t,...,\theta_n^t, X_1^t,...,X_n^t, a_{t-1} \right)}{P\left(X_1^t,...,X_n^t, a_{t-1} \right)}. \tag{3.14}$$

3.3.7 Reward Model

Let $G = \left[g\left(s,a,s',\theta \right) \right]$ denote the reward model where $g\left(s,a,s',\theta \right)$ is the immediate reward received when taking action $a \in A$ in state $s \in \mathrm{S}$ at time t, and moving to state $s' \in S$ at time $t+1$ and obtaining observation $\theta \in \Theta$. Note that the reward consists of two parts, namely, the data acquisition reward, g_{model}, which represents the reward for estimating these attribute values from the correlation model, and g_{sensor} which represents the reward for acquiring the attributes from the sensor network. Typically, reward values should be assigned such that $g_{\text{model}} > g_{\text{sensor}}$ when there is high confidence in the predicate of the attribute values so that the use of the correlation model estimation is favorable. On the other hand, if there is low confidence in the attribute values, we may need to acquire data from the sensor network in order to collect more information to answer such query with sufficient confidence to satisfy the user's confidence requirement. In such case, $g_{\text{model}} < g_{\text{sensor}}$ should be assigned.

3.3.8 Policy

A sensor selection policy (or plan) is a rule that maps a belief state to an action or sequence of actions. Suppose that at any given belief state, b, an action which decides whether to estimate a data from the correlation model or acquire data from the sensor network, is selected according to a specified policy π. A sensor selection policy is defined by

$$\Pi = \left\{ \pi : B \to A \middle| \pi(b) \in A(b), \forall b \in B \right\}, \tag{3.15}$$

where $\pi(b)$ refers to the action taken at belief state b under policy π.

Therefore, the goal is to find a sensor selection policy that optimizes some performance criterion.

3.3.9 Performance Criterion

The immediate reward for performing action $a \in A$ in state $s \in S$, $g(s,a)$, can be defined as

$$g(s,a) = \sum_{\forall s',\theta} p(s'|s,a) r(\theta|s',a) g(s,a,s',\theta). \tag{3.16}$$

Note that for a given belief state b, we have that

$$g(b,a) = \sum_{\forall s} b(s) g(s,a), \tag{3.17}$$

is the immediate reward for taking action $a \in A$ when the belief state is b. The value function of starting in belief state b and following policy π thereafter for a finite horizon of T steps is therefore

$$V^{\pi}(b) = E_{\pi}\left[\sum_{t=1}^{T} g(b_t, a_t) \middle| b_0 = \mathrm{b}\right], \tag{3.18}$$

where $E_{\pi}[.]$ is the expectation operator.

The objective is, hence, to find a policy such that

$$\pi^*(b) = \underset{\forall \pi}{\arg\max}\ V^{\pi}(b). \tag{3.19}$$

In other words, the objective is to find a sensor selection policy π that maximizes the average reward performance criterion while satisfying the user's confidence requirement on the data acquired from the network.

To achieve this goal, we apply an existing tool, called the witness algorithm to find a good sensor selection policy in Eq. (3.19).

3.4 Numerical Results

To evaluate the POMDP framework concept, we consider a simple network consisting of a base station (BS) which maintains a correlation model, and a sensor node that takes temperature reading as a single attribute (X_1). Since there is only one random variable, X_1, the BS needs to maintain only a one-dimensional Gaussian pdf, $p(X_1) = N(\mu_1, \sigma_1^2)$, for the correlation model. In addition, the

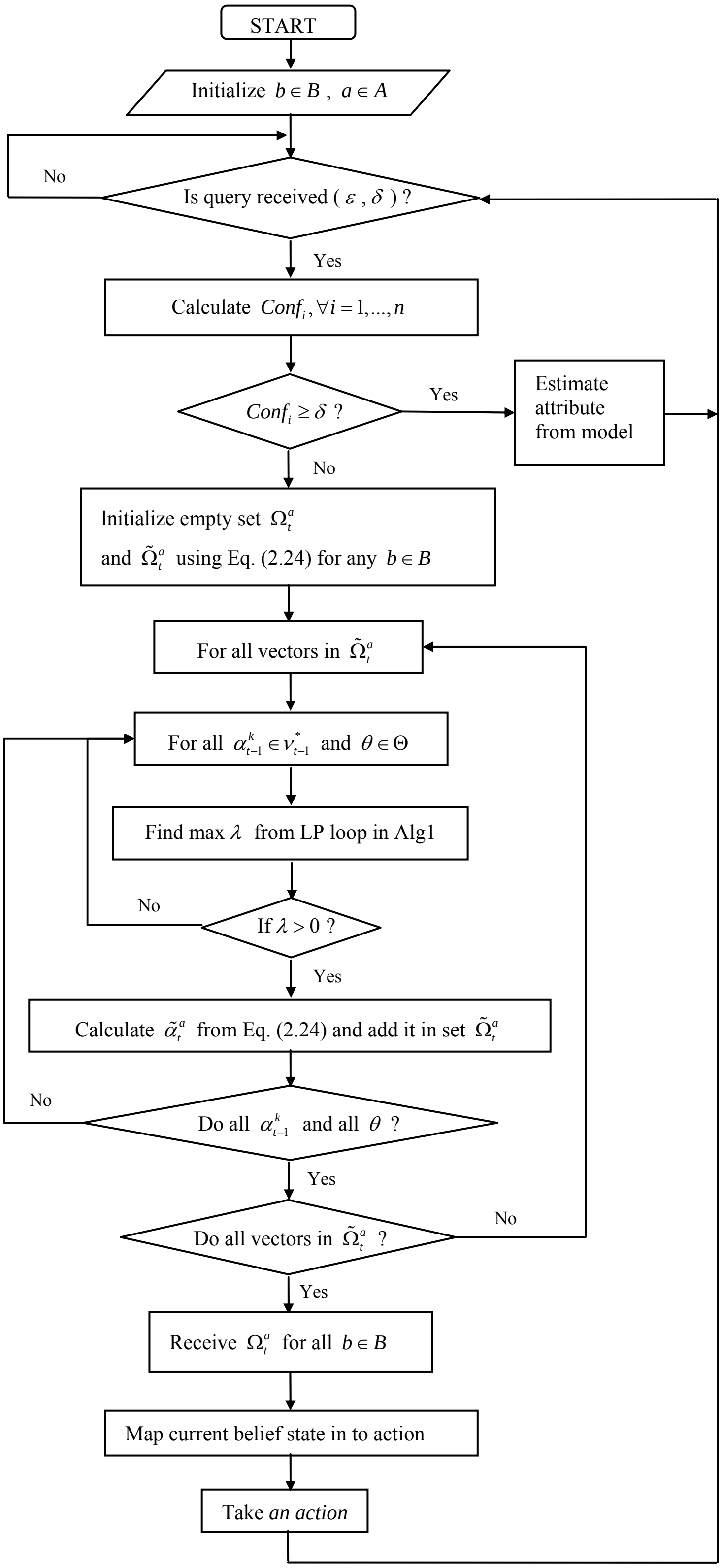

Figure 3.3 Block diagram of WIT conducted at the base station.

temperature values are inherently continuous-valued. Thus, the temperature values are quantized into 2 subintervals, [0, 20] and (20, 40] °C which are represented by integers 0 and 1, respectively. Therefore, the state space is $S=\{0,1\}$. The data acquisition process in the network starts when a user generates a query at the BS. The query includes an error tolerance (ε) of 4 °C and a target confidence bound (δ) of 95% that specifies how much uncertainty the user is willing to tolerate. The confidence level of sensor 1, $Conf$, is computed from Eq. (3.3) such that $Conf = P\left(X_1 \in \left[\mu_1 - \varepsilon, \mu_1 + \varepsilon\right]\right)$ where μ_1 is the sample mean of attribute X_1. If $Conf \geq \delta$, then the BS is sufficiently confident of its estimation and can directly estimate a temperature value from the correlation model to answer the user's query. If $Conf < \delta$, then the BS is not sufficiently confident and it must make a decision by selecting an action from a set of all possible actions $A=\{a_0, a_1\}=\{0,1\}$. If the BS decides to estimate the temperature value from the correlation model to answer the user's query (a_0), then the sensor node does not consume any energy. However, it is at the risk of obtaining an inaccurate estimate since the correlation model may be obsolete. If the BS decides to acquire a temperature value from the sensor (a_1), then the sensor node incurs energy costs as tradeoff for the additional temperature reading. The energy consumption for acquiring the temperature value can be computed by Eq. (3.5).

The results obtained from the simulations are compared with the randomized algorithm (RA), the heuristic search algorithm (HSA) (Deshpande et al., 2004) and the witness algorithm (WIT) (Cassandra, 1995). RA is a method that makes a randomized decision. RA is simple and requires low computational complexity. HSA is a method that makes a decision to *optimize the immediate* performance metric and guarantees the quality of collected data. This method is selected to demonstrate that the best immediate decision may not be optimal in the long run. WIT is an analytical POMDP solution method that makes a decision to *optimize the long-term* performance criterion according to the proposed POMDP framework.

We simulate our system on MATLAB. Each trial is carried out over a finite horizon of 1000 time steps, meaning that BS observes a number of 1000 queries. We average the results over 100 trials to attain a precision of 95% for all algorithms. We also use a specific probability transition model which presents changing states of temperature values at received the BS according to table 3.2. It is obtained by quantizing a sequence of temperature values into subintervals [0,20] and (20,40] °C. The temperature sequences used to obtain table 3.2 entries for a_0 and a_1 are generated from $N\left(18,1.44\right)$ and $N\left(18,4\right)$, respectively. To track the actual states of the WSN, the BS maintains a belief state of the WSN. The initial belief states of the WIT are uniform distribution, $b=\left[0.5,0.5\right]$. This belief state is used to represent the situation with highest uncertainty (i.e. the system is equally likely to be in any state) where there is no a previous knowledge about the actual temperature values.

Table 3.3 presents the reward structure which consists of the immediate rewards that the BS could obtain. If the BS observes a correct observation, then the BS receives -2 for choosing model estimation when $Conf_1 < \delta$; -0.9, which is the amount of energy consumed (in mJ), for choosing to acquire another temperature value from the WSN. If the BS observes an incorrect observation, then the BS receives -100 for both decisions.

Table 3.2 Probability Transition Matrices

$p(s' \mid s, a_0)$	$s'=0$	$s'=1$	$p(s' \mid s, a_1)$	$s'=0$	$s'=1$
$s=0$	0.95	0.05	$s=0$	0.85	0.15
$s=1$	0.95	0.05	$s=1$	0.85	0.15

Table 3.3 Reward Structures

$g(s, s', a_0, \theta)$	$\theta=0$	$\theta=1$	$g(s, s', a_1, \theta)$	$\theta=0$	$\theta=1$
$s=\{0,1\}, s'=0$	-2	-100	$s=\{0,1\}, s'=0$	-0.9	-100
$s=\{0,1\}, s'=1$	-100	-2	$s=\{0,1\}, s'=1$	-100	-0.9

The observation matrices shown in table 3.4, models the imprecise information ranging from the least to the highest imprecise information (i.e. the least to the highest state uncertainty) for 9 different cases. Each entry represents the probability that the BS observes any temperature values from a particular action. The higher the probability of the diagonal elements in the (a_1) matrix the less error-prone the WSN becomes. The higher the probability of the diagonal elements in the (a_0) matrix, the better (more precise) the correlation estimation becomes.

Table 3.4 Observation Matrices for Each Case

Case	$r(\theta \mid s', a_0)$	$\theta = 0$	$\theta = 1$	$r(\theta \mid s', a_1)$	$\theta = 0$	$\theta = 1$
1	$s' = 0$	1	0	$s' = 0$	1	0
	$s' = 1$	0	1	$s' = 1$	0	1
2	$s' = 0$	0.75	0.25	$s' = 0$	1	0
	$s' = 1$	0.5	0.5	$s' = 1$	0	1
3	$s' = 0$	0.5	0.5	$s' = 0$	1	0
	$s' = 1$	0.5	0.5	$s' = 1$	0	1
4	$s' = 0$	1	0	$s' = 0$	0.75	0.25
	$s' = 1$	0	1	$s' = 1$	0.5	0.5
5	$s' = 0$	0.75	0.25	$s' = 0$	0.75	0.25
	$s' = 1$	0.5	0.5	$s' = 1$	0.5	0.5
6	$s' = 0$	0.5	0.5	$s' = 0$	0.75	0.25
	$s' = 1$	0.5	0.5	$s' = 1$	0.5	0.5
7	$s' = 0$	1	0	$s' = 0$	0.5	0.5
	$s' = 1$	0	1	$s' = 1$	0.5	0.5
8	$s' = 0$	0.75	0.25	$s' = 0$	0.5	0.5
	$s' = 1$	0.5	0.5	$s' = 1$	0.5	0.5
9	$s' = 0$	0.5	0.5	$s' = 0$	0.5	0.5
	$s' = 1$	0.5	0.5	$s' = 1$	0.5	0.5

To evaluate the performance, all algorithms are compared using the following metrics:

1) The average long-term reward (*ALR*), which is the average accumulated immediate reward, $g(s, s', a, \theta)$, over a finite horizon of 1000 time steps. The immediate reward in each time step is obtained from table 3.3,

$$ALR = \frac{\sum_n accumulated\ immediate\ reward\ in\ trial\ n}{total\ number\ of\ trials} + \left| \min.\ average\ long\_term\ reward \right|. \tag{3.20}$$

2) The percentage which action 0 is selected ($\%A_0$), which is the average percentage that action "a_0" is selected over a finite horizon of 1000 time steps,

$$\%A_0 = \frac{\sum_n number\ of\ times\ that\ a_0\ is\ selected\ in\ trial\ n}{total\ number\ of\ horizons \times total\ number\ of\ trials} \times 100. \tag{3.21}$$

3) The average energy consumption (*AEC*), which is the average of the accumulated energy consumption over a finite horizon of 1000 time steps. The energy consumption in each time step is calculated from Eq. (3.5),

$$AEC = \frac{\sum_n accumulated\ energy\ consumption\ in\ trial\ n}{total\ number\ of\ trials}. \tag{3.22}$$

4) The average confidence level (%*C*) which is the average of the accumulated confidence level over a finite horizon of 1000 time steps. The confidence level (*Conf*) in each time step is calculated from Eq. (3.3).

$$\%C = \frac{\sum_n accumulated\ confidence\ levels\ in\ trial\ n}{total\ number\ of\ horizons \times total\ number\ of\ trials} \times 100. \tag{3.23}$$

3.4.1 The Average Long-Term Reward

Fig.3.4 compares the average long-term reward of the three algorithms as the imprecise information in network changes. Case 1 depicts the best case scenario when the BS has a precise correlation model and the WSN is error-free. It is seen that the three algorithms produce the highest average long-term reward. The reason is because with complete observability, the use of the correlation model *alone* can meet the user specified confidence level. As imprecise information in network increases in other cases, HSA and WIT produce similar average long-term rewards and these two schemes outperform RA. In addition, WIT outperforms other algorithms especially when confidence' correlation model is sufficient (case 4, 7 and 8). In worst case scenario with the least imprecise information as shown in case 9, it can be seen that, once again, all algorithms produce similar average long-term reward. The reason is because none of the algorithms could satisfy the user's confidence requirements in presence of high imprecise information in the network, so all algorithms are highly penalized. For all cases, WIT receives the highest total average long-term reward over all the other algorithms. Note that as the discount factor γ is varied from 0.1 to 0.9, WIT still outperforms other schemes, however with a marginal increase in average long-term reward of no more than 1%. Therefore, the value of $\gamma = 0.9$ was used throughout this study. In addition, the variance of data was also increased from 1.44 to 200 to evaluate how WIT performs against data with weak Markov Property. It is found that WIT still outperforms other algorithms, however with decrease in average long-term reward of no more than 2%. This is because the Markov property of data decreases with increasing data variance.

3.4.2 The Percentage which Action 0 is Selected

Fig.3.5 compares the percentage which action 0 is selected as the imprecise information in network changes. For the high network certainty scenario as shown in case 1, it is seen that WIT decides to select action 0 more often than other schemes. This is because the use of the model alone can meet the user's specified confidence bound. For cases 4, 5, 7 and 8, WIT uses the correlation model alone, therefore WIT does not consume any energy at all. This is either due to the precise model (case 4 and 7) or reasonably precise correlation model (case 5 and 8) of the sensor reading at the BS in presence of an error prone WSN. For cases 2, 3, 6 and 9, WIT rarely uses the model to estimate sensor reading and favors acquiring sensor reading from the WSN instead in order to increase confidence level of the answer. This is either due to the imprecise correlation model (case 9) or the fact that the sensor readings acquired from the WSN are more precise than the correlation model maintained at the BS.

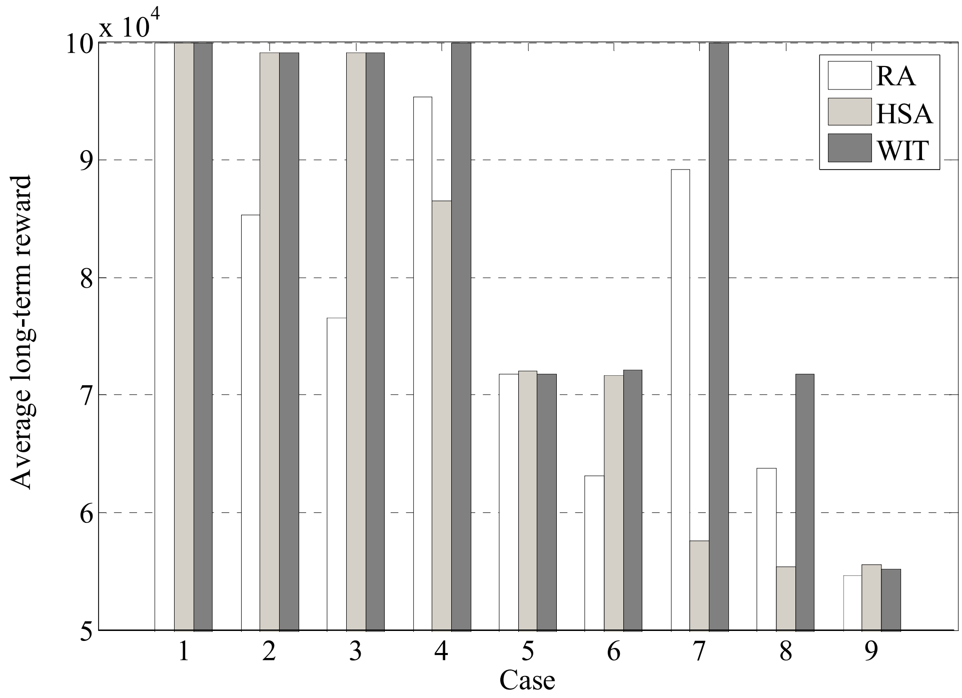

Figure 3.4 Average long-term reward as a function of imprecise information as shown in table 3.4.

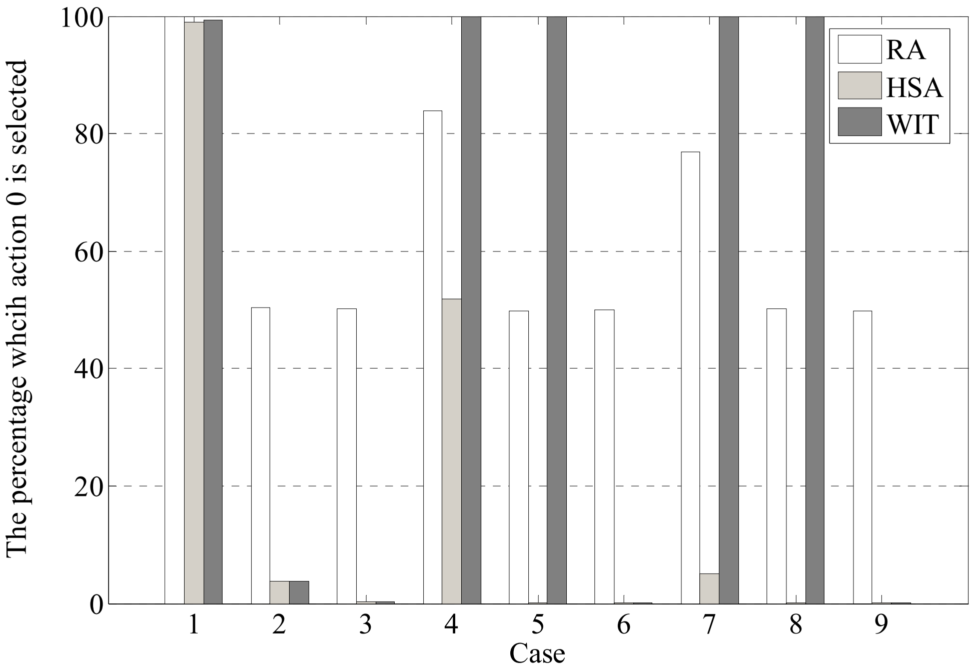

Figure 3.5 The percentage of average action 0 selection as function of imprecise information as shown in table 3.4.

3.4.3 The Average Energy Consumption

Fig. 3.6 compares the average energy consumption of the three algorithms as the imprecise information in network changes. Results show that the energy consumption of WIT is less than other schemes when the correlation model is good (i.e. high prediction accuracy) as shown in case 1, 4, 7 and 8, or when the WSN is highly erroneous and the BS still maintains a moderate model as shown in case 5 and 8. This is because the sensor selection in WIT prefers the use of model over querying the temperature values when appropriate, resulting in energy consumption savings. For case 2, 3, 6 and 9, energy consumption is increased for WIT as the BS acquires additional temperature values as a result of increased inaccuracy in the correlation model. However, both WIT and HSA still outperform RA.

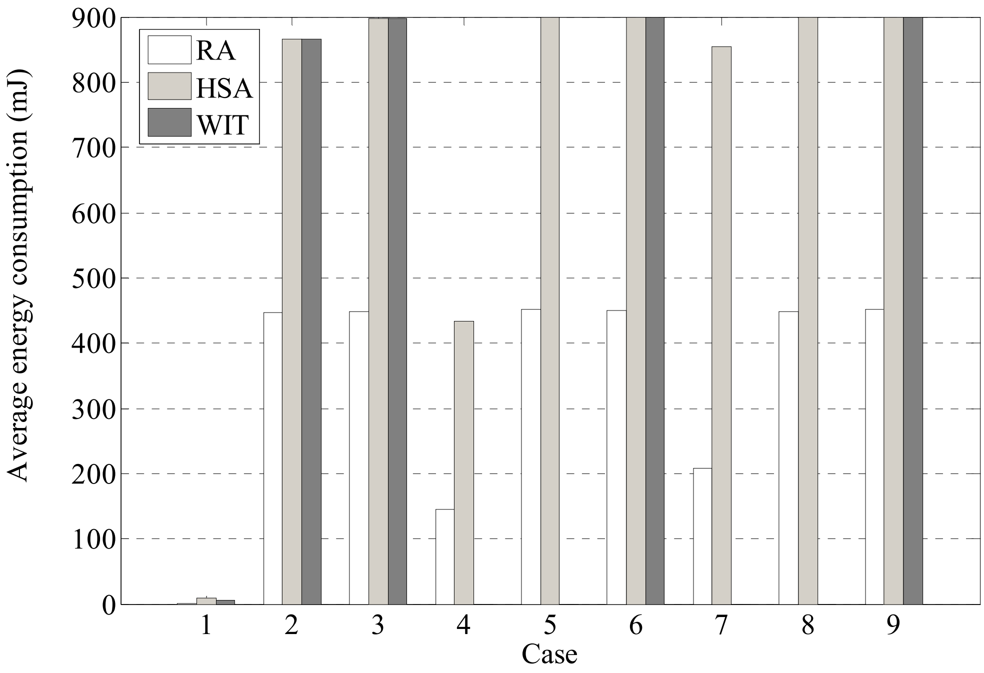

Figure 3.6 Average energy consumption as function of imprecise information as shown in table 3.4.

3.4.4 The Average Confidence Level

Fig. 3.7 compares the average confidence level of the three algorithms as the imprecise information in network changes. In the presence of highest network certainty in case 1, all algorithms produce the highest average confidence level of about 100%. This is because the use of a correlation model alone can meet the user's specified confidence bound. Note that WIT gives higher average confidence level than other schemes when the WSN becomes increasingly erroneous and the BS still maintains a precise correlation model as shown in case 4,5,7 and 8. This is because WIT prefers the use of model over acquiring temperature values to answer query. This is either due to the precise correlation model at the BS (case 4 and 7) or the fact that the WSN is error-prone (case 5 and 8). For the error-free WSN scenario as shown in case 2, 3, and reasonably low WSN error in case 6, WIT and HSA produce similar average confidence level and these two algorithms outperform RA. The reason is because acquiring additional temperature values is more precise than estimating the temperature value from the correlation model at the BS. In case 9 when the information imprecision is the highest, all algorithms give the least confidence level of about 22%. For all cases, it can be seen that WIT is able to attain higher confidence level than all other algorithms.

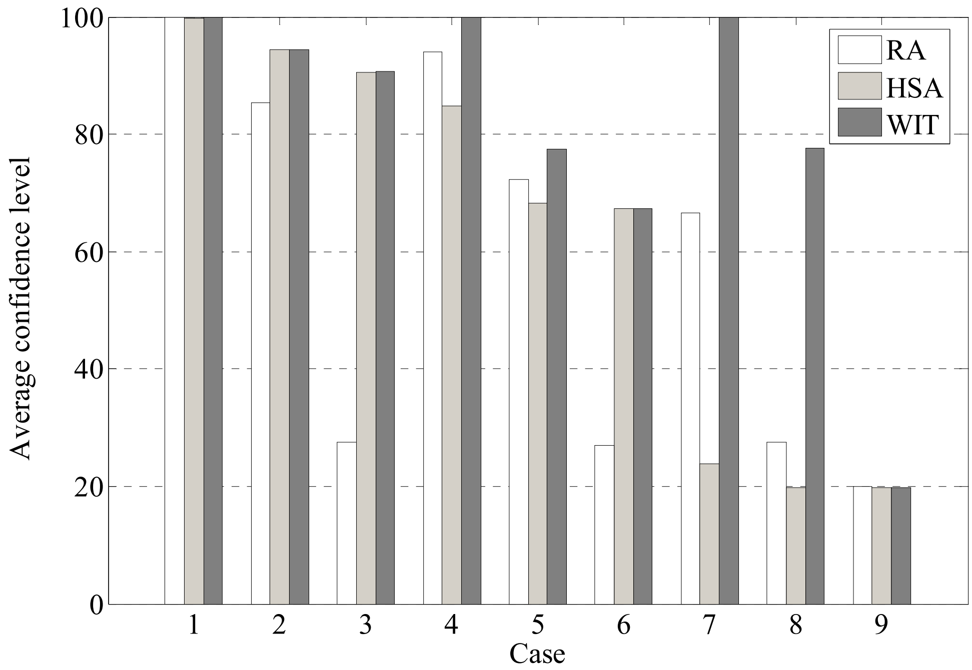

Figure 3.7 Average confidence level as function of imprecise information as shown in table 3.4.

3.5 Conclusions

In this chapter, we study the data acquisition problem which can satisfy probabilistic confidence requirements of the acquired data in an error-prone wireless sensor network (WSN). The objective of the problem is to optimize the long-term performance criterion while maintaining quality of data under error-prone WSNs. We propose a discrete-state partially observable Markov decision process (POMDP) framework for the data acquisition problem. We applied the witness algorithm to solve for a good sensor selection policy. Numerical results show that the proposed scheme can select good sensor readings by maximizing the average long-term reward while balancing the tradeoff between the long-term average communication costs and the information quality in an error-prone wireless sensor network (WSN). Although, discrete-state POMDP formulation and WIT may appear for the data acquisition problem with continuous-valued data, our framework can still be employed in data acquisition schemes with discrete-states such as object tracking in a grid of sensors (Fuemmeler and Veeravalli, 2008).

CHAPTER IV

A PARAMETRIC POMDP FRAMEWORK FOR DATA ACQUISITION IN WSNs : FITTED VALUE ITERATION

4.1 Introduction

In **chapter 3**, we study the long-term sensor selection plan to provide the quality of the answer query and reduce resource consumption. The scheme formulates the problem as discrete-state partially observable Markov decision process (POMDP). However, most sensor readings are inherently continuous-valued such as temperature, humidity, target distance, etc. Except for certain applications, discrete-state POMDP formulation is inappropriate for the data acquisition problem in many realistic situations. To extend the problem to a more realistic scenario, this chapter formulates the data acquisition problem as continuous-state partially observable Markov decision process. It is worth noting that the reason we initially used the discrete-state POMDP formulation in the previous chapter is to demonstrate that the data acquisition problem can be mapped in to discrete-state POMDP framework and to determine the best long-term sensor selection plan by using an exact analytical solution such as the witness algorithm. However, the drawback of discrete-state POMDP solution method is that the computational burden increases rapidly with fine state discretization. On the other hand, the continuous-state POMDP formulation employed in this chapter avoids fine continuous state space discretization, and instead represents the belief state in a parametric form. Such formulation is referred to a parametric POMDP (PPOMDP) (Brooks, Makarenko, Williams and D-Whyte, 2006). PPOMDP is a continuous-state partially-observable Markov decision process which represents probability distributions over a continuous state space, or belief states, in a parametric from using low-dimensional vectors of sufficient statistics. The sufficient statistics refers to the use of a low-dimensional vector (or a small number of parameters) to represent the belief state and closely approximate the belief state transition function. We focus on the use of Gaussian distributions for representing the parametric form. The reason is because the dimension of the parametric form using a multivariate Gaussian distribution will be significantly lower than that obtained by directly discretizing the continuous state space.

To demonstrate the performance of the framework, we then solve for a good sensor selection plan in finite horizon by using an existing method for solving the PPOMDP called the fitted valued iteration (FVI) method. We then compare its performance with other existing data acquisition algorithms. The contribution in this chapter is therefore twofold, i) the mapping of the data acquisition problem into parametric POMDP framework to cater real-value data and ii) the application of the fitted value iteration to solve for a good sensor selection plan.

This chapter is organized as follows. The overview of the wireless sensor network architecture and the data acquisition process is described in section 4.2. Section 4.3 provides the formulation of the data acquisition problem in WSNs as a PPOMDP. In section 4.4, the numerical results will be presented. Finally, section 4.5 summarizes the entire chapter.

4.2 Data Acquisition Structure

This section provides an overview of our basic system and data acquisition process. As shown in Fig. 4.1, the system consists of a base station (BS) which includes declarative query processing engine and a number of sensor nodes that acquire data from the environment in wireless sensor network.

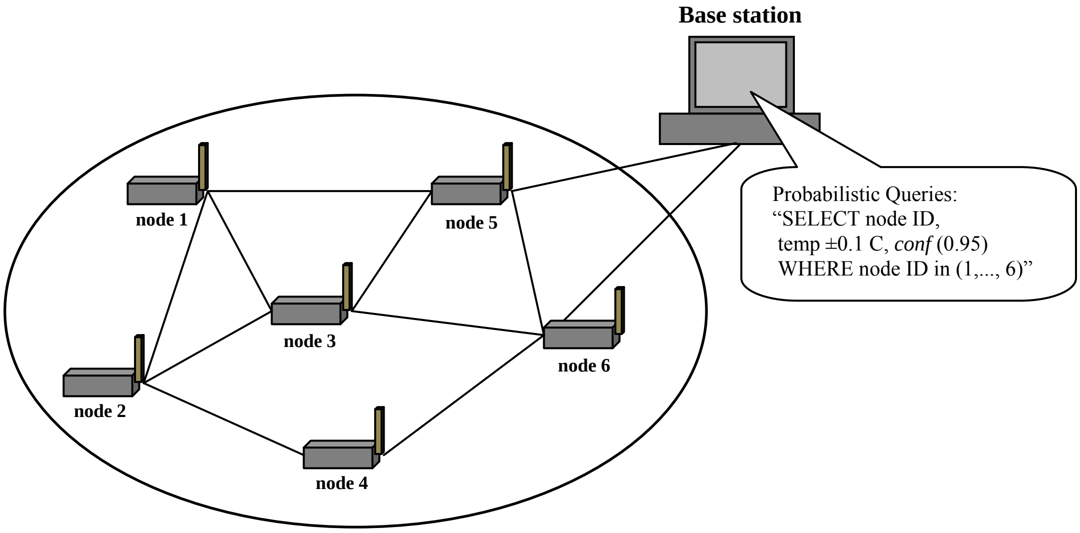

Figure 4.1 Data acquisition structure in a WSN of 6 nodes.

In environment monitoring, the requirements for data quality assurance are usually determined by the user, who specifies a required confidence threshold on the query result. For example, a user may generate queries at the BS for temperature values of the environment in a WSN. The query includes an error tolerance (ε) and a target confidence bound (δ) that specifies how much uncertainty the user is willing to tolerate. The BS must then decide how to answer the query. Based on the knowledge of the sensor network, the BS may decide to collect data from a specific sensor(s) with high data quality to reduce resource consumption instead of collecting data from all sensors in the network.

In this thesis, the knowledge of the sensor network has been summarized into a correlation model (Deshpande et al., 2004). The correlation model captures the spatial and temporal correlations of the sensor readings. Fig. 4.2 presents a trace of voltage and temperature readings over a 2-day period from two sensors. Notice the close correlations between the two attributes. Based on a correlation model of the network, the BS may decide to collect other attributes such as voltage (0.00009 mJ per sample) which consume less energy instead of a temperature value (0.5 mJ per sample) to answer such query. As aforementioned in **chapter 3**, we can apply the correlation model for data acquisition problem to support the quality of query result and also reduce resource consumption.

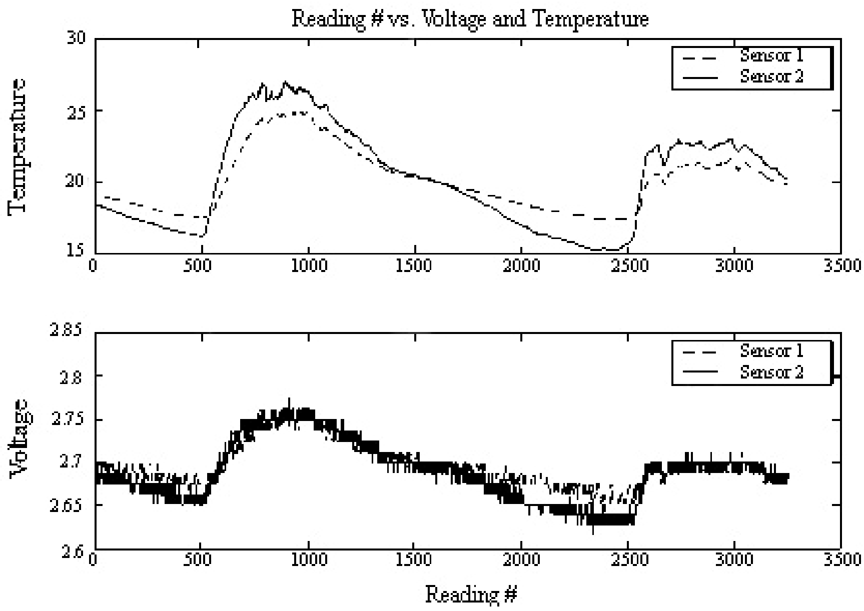

Figure 4.2 Trace of voltage and temperature readings over a two-day period from Mica2 motes sensors (Deshpande et al., 2004).

4.2.1 The Correlation Model

This section presents a correlation model to describe the relation between the attributes stored at the BS. Let X_i, $i = 1,...,n$, be a random variable representing attribute i, e.g., temperature on an arbitrary sensing node in a WSN. There is typically one attribute per sensor type per sensor node. Therefore, X_i is referred to a random variable for an attribute at sensor i in this thesis. As in the previous chapter, we describe the correlation model as a joint probability density function (pdf), $p(X_1, X_2,..., X_n)$, which assigns a probability to each joint value $x_1, x_2,..., x_n \in \mathbb{R}$ for sensors $(1, 2,..., n)$, where $\mathbb{R}$ is the set of real numbers. The joint probability density function describes the likelihood of occurrence and the relation between stored attributes.

In this thesis, we propose a specific correlation model based on a multivariate Gaussian distribution. A Gaussian pdf over a set of attributes from *n* sensors, $X_1, X_2,..., X_n$ can be viewed as a function of two parameters, i.e. a *n*-dimensional vector of means $(\bar{\mu})$ and a $n \times n$ matrix of covariances $(\mathbb{C})$. The mean vector $(\bar{\mu})$ is the center of this probability distribution and the covariance $(\mathbb{C})$ indicates the spread of the distribution which is a measure of correlation between attributes as shown in Eq. (3.1) and. (3.2), respectively.

4.2.2 The Probabilistic Queries

Based on the correlation model described above, the BS can use this model to estimate values to answer queries when its probability exceeds the user's specified confidence threshold. Such probability can be computed from the correlation model and a specific query type such as range query, value query[9].

Suppose that a user generates a query with an error bound of ε confidence at least δ for the value of sensor *i*, X_i. In this thesis, we are interested in considering range queries, which ask whether the value of sensor *i*, X_i, is in an arbitrary range $[L_i, U_i]$. Let L_i represent $x_i - \varepsilon$ and U_i represent $x_i + \varepsilon$. Hence, a user submits a range query if he wants to know whether the probability that X_i is within a specified range, $P(X_i \in [x_i - \varepsilon, x_i + \varepsilon])$, exceeds or meets the confidence requirement, δ. The probability $P(X_i \in [L_i, U_i])$ can be computed by

$$P(X_i \in [L_i, U_i]) = \int_{L_i}^{U_i} p(x_i)\, dx_i, \tag{4.1}$$

where $p(x_i) = \int p(x_1, ..., x_n)\, dx_1 ... dx_{i-1} dx_{i+1} ... dx_n$ is the marginal probability of the attribute in sensor *i*.

---

[9] For value queries, where a value of a particular attribute X_i is of interest, the mean value of X_i can be computed from $\mu_i = \int x_i p(x_i | \theta) dx_i$, where $p(x_i | \theta)$ is the posterior probability of attribute X_i given observation θ.

4.2.3 Energy Consumption Model

The energy consumption model on each sensor is obtained from the data sheets of sensors and the radio used on Mica2 Motes with a Crossbow MTS400 environmental sensor board as shown in table 3.1. We assume that the sender and receiver are well synchronized. The Mica2 Motes consume about 0.4 mJ of energy for both sending and receiving modes. When a user sends a query for a specific attribute from the environment, the total energy consumption for acquiring data that transits through a number of n_t nodes and acquires a number of sensor readings from n_a nodes can be calculated as follows

$$\text{total energy consumption per attribute} = \left(E_s \times n_t\right) + \left(E_r \times n_t\right) + \left(E_{att} \times n_a\right), \tag{4.2}$$

where E_s is the consumed energy in the sending mode, E_r is the consumed energy in the receiving mode and E_{att} is the energy consumed from sampling a specific attribute.

In this thesis, we primarily focus on the temperature readings since temperature has both temporal and spatial correlations. The temporal correlation of temperature characterizes a dynamic correlation model. Therefore, the temperature values observed earlier in time should assist in estimating the temperature values later in time. We assume that the temporal characterization of temperature exhibits *Markov property* as verified by statistical tests (see section 3.1).

4.3 Parametric Partially Observable Markov Decision Process Formulation

This section presents the formulation of the data acquisition problem in WSNs as a PPOMDP. A PPOMDP formulation is adopted to simplify the continuous-state POMDP framework. In particular, PPOMDP uses a parametric form with low dimensional vectors of sufficient statistics, Φ_t, to represent the *belief state*. The general components of the PPOMDP framework are similar to the POMDP framework in section 3.3, however, with the replacement of belief states by a simpler (parametric) form of sufficient statistics. For the sake of completeness, all PPOMDP components are presented as follows.

4.3.1 State Space

Let $[x_1, x_2, ..., x_n]$ be a vector of the attribute values of *n* sensors. The state space is a set of all possible attribute values of *n* sensors in the network which can be defined by

$$S = \{s : s = [x_1, x_2, ..., x_n]\}, x_i \in \mathbb{R}, i = 1, ..., n, \tag{4.3}$$

where $\mathbb{R}$ is the set of real numbers.

4.3.2 Action Space

When a user submits a query, the BS must then make a decision to answer the user's query by either deciding to estimate the data from the BS's correlation model, or to acquire data from a specific sensor node(s) in the network.

The decision at the BS is referred to actions. The action space, denoted by *A*, is a set of all possible actions which can be taken by the BS defined by

$$A=\{a:a=[\mathrm{a}_1,\mathrm{a}_2,...,\mathrm{a}_n],\mathrm{a}_i\in\{0,1\},i=1,2,...,n\}, \tag{4.4}$$

where a_i is the action for sensor node i such that

$$\mathrm{a}_i=\begin{cases}1 & \text{, if the attribute is acquired from the sensor } i\\ 0 & \text{, if the attribute is estimated from the model.}\end{cases} \tag{4.5}$$

In other words, $\mathrm{a}_i = 1$ is the BS's action to answer the query by actually acquiring the attribute from the sensor i in the network, whereas $\mathrm{a}_i = 0$ is the BS's action to answer the query by estimating the attribute value of sensor i from the correlation model, $p(X_1,X_2,...,X_n)$ at the BS.

4.3.3 Observation Space

Due to the possibility that sensor may give faulty values, the BS may not see or receive the actual state of the environment. This could be caused by faulty communication between sensors, sediments from the surroundings which may have been caught on the sensors through rain, wind, or animals passing by, or faulty sensors themselves, etc. Therefore, the BS sees merely an observation, $\theta = s \pm \vec{e}$, instead of the actual state (s) of the environment, where $\theta\in\Theta$, $\vec{e}$ is the n-dimensional vector of the observation error and Θ is the set of all possible observations. In particular,

$$\Theta=\{\theta:\theta=[\theta_1,\theta_2,...,\theta_n],\theta_i\in\mathbb{R},i=1,2,...n\}. \tag{4.6}$$

4.3.4 Parametric Form of Sufficient Statistics

A parametric form of sufficient statistics is a representation of a continuous distribution over continuous state space, or belief state (b_t), by a relatively small number of sufficient statistics resulting in a relatively low-dimensional representation of belief space.

In this chapter, we focus on the use of a Gaussian parametric form to represent the belief state over a continuous state space. The reason is because the number of parameters used for representing belief state by a multivariate Gaussian distribution will be significantly lower than that obtained by discretisation. To illustrate this fact, consider a simple data acquisition problem with a single temperature sensing node. Discretising the continuous state space into $|S|$ bins requires computation of $|S|$ value functions. This computation becomes enormous with fine discretization of continuous state space. Alternatively, one could represent the distribution as a Gaussian distribution with two parameters, (μ, σ^2), thereby converting to the problem of computing a value function over a two-dimensional continuous space.

Let Φ_t be a vector of sufficient statistics representing a belief state which specifies the probability of the attribute values seen at the BS being in arbitrary state $[x_1, x_2, ..., x_n]$, $p(X_1, X_2, ..., X_n)$. For example, consider the aforementioned simple data acquisition problem with a single temperature sensor. The vector $\Phi_t = [\mu, \sigma^2]$ is a vector of sufficient statistics representing the probability that the temperature value seen at the BS is in arbitrary state $\left(x_1\right)$ as $p\left(X_1\right)=\exp\left(-\left(x_1-\mu\right)^2/2\sigma^2\right)\Big/\sqrt{2\pi}\sigma$.

After taking an action a_t when the sufficient statistics is Φ_t at time step *t*, the BS receives a new observation, θ_{t+1}, at time step $t+1$ and the new sufficient statistics at time step $t+1$ is also updated by a belief update function to

$$\Phi_{t+1}=\tilde{\Gamma}\left(\Phi_t,a_t,\theta_{t+1}\right), \tag{4.7}$$

where $\tilde{\Gamma}$ is an approximate belief update function which is used to update the sufficient statistics in order to track the belief state at each time step. Note that the exact belief update function is given by Eq.(2.9) in **chapter 2**. Since the PPOMDP framework replaces the belief state with sufficient statistics, the transition probability model, the observation model, and immediate rewards must be derived in terms of Φ_t as follows.

4.3.5 Sufficient Statistics Transition Model

At time step *t*, the BS takes action $a_t\in A$ when the sufficient statistics is Φ_t. The values of the *n* attributes at the BS change to a new state, $s_{t+1}=s'\in S$, at time $t+1$ according to the sufficient statistics transition probability model. Define $P\left(s_{t+1}\middle|\Phi_t,a_t\right)$ as the probability that the values of the *n* attributes at the BS transits to a new state s_{t+1} given that it is currently in sufficient statistics Φ_t and action a_t was just taken. Hence, the sufficient statistics transition probability can be written by

$$P\left(s_{t+1}\middle|\Phi_t,a_t\right)=P\left(X_1^{t+1},\ldots,X_n^{t+1}\middle|\Phi_t,a_t\right). \tag{4.8}$$

4.3.6 Sufficient Statistics Observation Model

As mentioned in section 4.3.3, the data acquired at the BS may not always represent the actual attribute value of the environment. As a result, the direct interpretation of the sensor reading may not be a reliable representation of the real world. Hence, the BS may not have access to the actual state transition process as referred to in Eq. (4.8). Let us now define $P\left(\theta_{t+1}\middle|\Phi_t, a_t\right)$ as the sufficient statistics observation model which denotes the probability that the BS observes θ_{t+1} when the current sufficient statistics is Φ_t and the action just taken was $a_t \in A$. Note that $P\left(\theta_{t+1}\middle|\Phi_t, a_t\right)$ can be determined by weighing the probability of observing observation θ_{t+1} given that the transited state is $s_{t+1} \in S$ with the probability of state $s_{t+1} \in S$ occurring given that the current sufficient statistics is Φ_t and the action just taken was $a_t \in A$. That is,

$$P\left(\theta_{t+1}\middle|\Phi_t, a_t\right) = \int P\left(\theta_{t+1}\middle|s_{t+1}\right) P\left(s_{t+1}\middle|\Phi_t, a_t\right) ds_{t+1}, \tag{4.9}$$

where $P\left(\theta_{t+1}\middle|s_{t+1}\right)$ is the observation likelihood function.

4.3.7 Sufficient Statistics Reward Model

Let $G = \left[g\left(s, a, s', \theta\right)\right]$ denote the reward matrix where $g\left(s, a, s', \theta\right)$ denotes the immediate reward received by the BS at time $t+1$ when taking action $a \in A$ in state $s \in \mathrm{S}$ at time t, and moving to state $s' \in S$ at time $t+1$ and seeing an observation $\theta \in \Theta$. Note that the reward structure consists of two parts, namely, the data acquisition reward, g_{model}, which represents the reward obtained from taking an action to estimate an attribute value of specific sensor(s) from the correlation model, and g_{sensor} which represents the reward from taking an action to acquire an attribute value from specific sensor(s) in the sensor network. Usually, reward values should be assigned such that $g_{\text{model}} > g_{\text{sensor}}$ when there is high confidence in the data quality. Such reward regime encourages the BS to use model estimation whenever the quality of the correlation model at the BS is reasonably good, which can give high quality data estimation. On the other hand, if there is low confidence in the data quality, the BS may need to acquire data from the sensor network in order to collect more data to answer a data query with sufficient confidence. In such case, $g_{\text{model}} < g_{\text{sensor}}$ should be assigned. Furthermore, let $g(s,a)$ be the immediate reward for performing action $a \in A$ in state $s \in S$ such that

$$g(s,a) = \sum_{\forall s',\theta} p(s'|s,a)\, r(\theta|s',a)\, g(s,a,s',\theta). \tag{4.10}$$

Since the actual state, *s*, is unknown or hidden from the BS, we must resort to representing the state by sufficient statistics, Φ_t.The sufficient statistics immediate reward can then be written in terms of Φ_t by

$$\tilde{g}(\Phi_t, a_t) = \int g(s_t, a_t)\, p(s_t|\Phi_t) ds_t, \tag{4.11}$$

where $p(s_t|\Phi_t)$ is the sufficient statistics for the entire history.

4.3.8 Policy

For a generic POMDP problem, a policy is a rule that maps a belief state to an action. In the other words, for a parametric POMDP problem, a policy is a rule that maps a sufficient statistics to an action. Consider the data acquisition framework. Suppose that at any given sufficient statistics, Φ, an action which decides whether to estimate a data from the correlation model or acquire data from the sensor network, is selected according to a specified policy π. A sensor selection policy is defined by

$$\Pi = \left\{ \pi : \beta \to A \middle| \pi(\Phi) \in A(\Phi), \forall \Phi \in \beta \right\}, \tag{4.12}$$

where $\pi(\Phi)$ refers to the action taken at belief state Φ under policy π.

Therefore, the goal is to find a sensor selection policy that is optimal for some performance criterion.

4.3.9 Performance Criterion

The aim of this chapter is to find a sensor selection policy that maximizes the average long-term reward while still satisfying the user's confidence requirement on the data quality. To achieve the sensor selection policy, we define the value function in terms of sufficient statistics Φ_t by

$$\tilde{V}^{\pi}(\Phi_t) = \tilde{g}(\Phi_t, a_t) + \gamma \underset{\theta_{t+1}}{E}\left[\tilde{V}^{\pi}\left(\tilde{\Gamma}(\Phi_t, a_t, \theta_{t+1})\right)\right], \tag{4.13}$$

where the $\underset{\theta_{t+1}}{E}[.]$ is the expectation over all observations which can be calculated by Eq. (4.9). Hence, the goal is to find a policy such that

$$\pi^*(\Phi) = \arg\max_{\forall \pi} \tilde{V}^{\pi}(\Phi). \tag{4.14}$$

In other words, the goal is to find a sensor selection policy π that maximizes the average long-term reward performance criterion while satisfying the user's confidence requirement of the data acquired from the network. Since the value function (see Eq. (4.13)) is not piecewise-linear and convex (PWLC) in sufficient-statistics space, we apply an existing tool, called the fitted value iteration (FVI) to determine the best sensor selection policy in Eq. (4.14). The pseudo-code of the FVI method is shown in algorithm2. Because the fitted value iteration is an approximate algorithm, its solution can only provide an approximate sensor selection policy. On the other hand, the witness algorithm in **chapter 3** is an analytical scheme which is employed to find a good sensor selection policy. However, the applicability of the witness algorithm is restricted for only the piecewise-linear and convex (PWLC) value function and discrete-state POMDP problems.

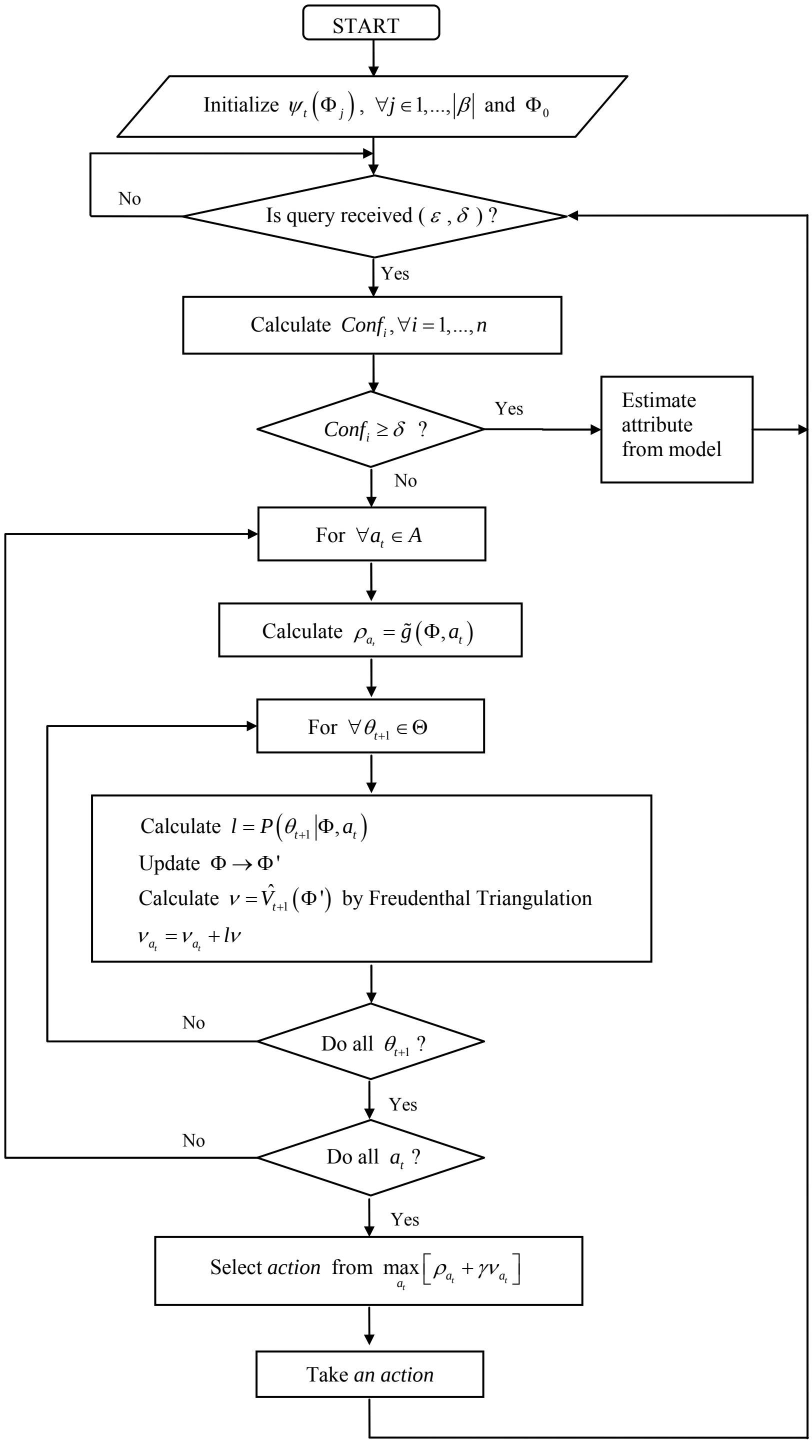

Figure 4.3 Block diagram of FVI conducted at the base station.

4.4 Numerical Results

To evaluate the PPOMDP framework concept, we consider two simple networks:

1) *A WSN with only temporal correlation.* This network consists of a base station (BS) and a single sensor node that takes temperature reading as a single attribute. We study this simple scenario in order to compare the data acquisition problem formulated as a continuous-state POMDP (solved by the FVI) and the discrete-state POMDP (solved by WIT from the previous chapter). The purpose of this scenario is to demonstrate that the sensor selection policy obtained by FVI is nearly as good as that obtained from WIT in the long run. Note that due to scalability limitations (see section 4.3.4), WIT is only used in this scenario to demonstrate the long-term optimality of the exact POMDP solution. The correlation model used in this scenario is $p(X_1) = \exp\left(-(x_1-\mu)^2 / 2\sigma^2\right) \big/ \sqrt{2\pi}\sigma$.

2) *A WSN with both spatial and temporal correlation.* This network consists of a base station (BS) and two temperatures sensor nodes located near each other which are highly correlated. The correlation coefficient is equal to 0.9, meaning that the temperature reading from two sensor nodes are (almost) directly proportional with each other. The objective of studying this scenario is to compare the performance of the sensor selection policy obtained from formulating the data acquisition problem as a continuous-state POMDP (solved by the FVI) with other existing sensor selection schemes, when both the spatial and temporal

correlation models are used. As shown in section 4.4.1, FVI is near-optimal (i.e. it is nearly as good as WIT). Therefore, WIT is no longer used for comparison in this scenario. The correlation model used in this scenario is $p(X_1, X_2) = \exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu})^{'}\mathbb{C}^{-1}(\vec{x}-\vec{\mu})\right\} \Big/ 2\pi|\mathbb{C}|^{1/2}$.

4.4.1 The Temporal Correlation Network Study

Firstly, we consider the temporal correlation WSN which consists of a base station (BS) and one sensor node (i.e. a single sensor exhibits no spatial correlation) that takes a temperature reading as a single attribute(X_1). Since temperature is inherently continuous-valued, the temperature values are quantized into 2 subintervals, [0,20] and (20,40] °C which are represented by integers 0 and 1, respectively. Therefore, the state space is $S = \{0,1\}$. The data acquisition process starts when a user generates a query at the BS for temperature values of the environment. The queries include an error tolerance (ε) of 4 °C and a target confidence bound (δ) of 95% that specifies how much uncertainty the user is willing to tolerate. The confidence level of sensor 1, $Conf = P(X_1 \in [\mu_1 - \varepsilon, \mu_1 + \varepsilon])$, can be calculated from Eq. (4.1) where μ_1 is the sample mean of the attribute X_1. Since there is only a single random variable, X_1, the BS needs to maintain only a one-dimensional Gaussian pdf model, $p(X_1) = N(\mu_1, \sigma_1^2)$. If $Conf \geq \delta$, then the BS is sufficiently confident of its estimation and it can directly compute a temperature estimate from the pdf model, $p(X_1) = N(\mu_1, \sigma_1^2)$, to answer the user's query. If $Conf < \delta$, then the BS is not sufficiently confident and it must make a decision by selecting an action from a set of all possible actions $A=\{a_0,a_1\}=\{0,1\}$. If the BS decides to compute the temperature estimate from the correlation model to answer the user's query (a_0), then the sensor node does not consume any energy. However, it is at the risk of obtaining an inaccurate estimate. If the BS decides to acquire a reading from a sensor (a_1), then the sensor incurs energy costs to tradeoff increased accuracy from the additional temperature reading. The energy consumption for acquiring the temperature reading can be computed by Eq. (4.2).

To quantify the performance, we compare four algorithms : 1) randomized algorithm (RA) is a method that makes a randomized decision, 2) heuristic search algorithm (HSA) (Deshpande et al., 2004) is a method that makes a decision to *optimize the immediate* reward, 3) witness algorithm (WIT) (Cassandra, 1995) is an analytical POMDP solution method that makes a decision to *optimize the average long-term* reward, and 4) fitted value iteration (FVI) (Brooks, Makarenko, Williams and D-Whyte, 2006), which is the proposed method we applied for obtaining a simplified PPOMDP solution, that makes a decision to attain *near-optimal average long-term* reward.

The numerical results are obtained by simulation using Microsoft Visual C++. Each trial is carried out over a finite horizon of 1000 time steps, meaning that 1000 user's queries are generated to the BS. The trials are repeated for 100 runs to achieve 95% precision. Sensor readings received at the BS change states according to the transition probability model in table 4.1 which is obtained by quantizing a sequence of temperatures into subintervals [0,20] and (20,40] °C. The temperature sequences used to obtain table 4.1 entries for a_0 and a_1 are generated from

$N(18,1.44)$ and $N(18,4)$, respectively. To track the actual states of the WSN, the BS maintains a belief state of the WSN. The initial belief states are [0.5,0.5] for the WIT and $N(20,10)$ for FVI (which roughly equals to [0.5,0.5] of WIT because the center of this probability distribution is 20 which same value that is used to quantize the state).

Table 4.1 Probability Transition Matrices in the Temporal Correlation Network

$p(s'\|s,a_0)$	$s'=0$	$s'=1$	$p(s'\|s,a_1)$	$s'=0$	$s'=1$
$s=0$	0.95	0.05	$s=0$	0.85	0.15
$s=1$	0.95	0.05	$s=1$	0.85	0.15

The immediate rewards which the BS could obtain are shown in table 4.2. The BS receives -2 for choosing model estimation when $Conf_1 < \delta$; -0.9, which is the amount of energy consumed (in mJ), for choosing to acquire more data from the WSN; and -100 if the BS observes an incorrect observation.

Table 4.2 Reward Structure in the Temporal Correlation Network

$g(s,s',a_0,\theta)$	$\theta=0$	$\theta=1$	$g(s,s',a_1,\theta)$	$\theta=0$	$\theta=1$
$s=\{0,1\},s'=0$	-2	-100	$s=\{0,1\},s'=0$	-0.9	-100
$s=\{0,1\},s'=1$	-100	-2	$s=\{0,1\},s'=1$	-100	-0.9

The observation matrices shown in table 4.3, model the data uncertainty ranging from the least to the highest data uncertainty for 9 different cases. Each entry represents the probability that the BS observes a temperature value (observation) after having taken a particular action. Consider the observation matrices on the right hand side of the table which give observation probabilities if the BS decides to acquire additional sensor reading (a_1). For each matrix, the higher the diagonal probabilities, the more reliable the WSN becomes (i.e. the lower the WSN error). Similarly, the observation matrices on the left hand side of the table give the observation probabilities if the BS decides to estimate the sensor reading from the correlation model (a_0). The higher the diagonal probabilities in these matrices, the better the correlation estimation becomes (i.e. the more precise the correlation model).

Table 4.3 Observation Matrices in the Temporal Correlation Network

Case	$r(\theta \mid s', a_0)$	$\theta = 0$	$\theta = 1$	$r(\theta \mid s', a_1)$	$\theta = 0$	$\theta = 1$
1	$s' = 0$	1	0	$s' = 0$	1	0
	$s' = 1$	0	1	$s' = 1$	0	1
2	$s' = 0$	0.75	0.25	$s' = 0$	1	0
	$s' = 1$	0.5	0.5	$s' = 1$	0	1
3	$s' = 0$	0.5	0.5	$s' = 0$	1	0
	$s' = 1$	0.5	0.5	$s' = 1$	0	1
4	$s' = 0$	1	0	$s' = 0$	0.75	0.25
	$s' = 1$	0	1	$s' = 1$	0.5	0.5
5	$s' = 0$	0.75	0.25	$s' = 0$	0.75	0.25
	$s' = 1$	0.5	0.5	$s' = 1$	0.5	0.5
6	$s' = 0$	0.5	0.5	$s' = 0$	0.75	0.25
	$s' = 1$	0.5	0.5	$s' = 1$	0.5	0.5
7	$s' = 0$	1	0	$s' = 0$	0.5	0.5
	$s' = 1$	0	1	$s' = 1$	0.5	0.5
8	$s' = 0$	0.75	0.25	$s' = 0$	0.5	0.5
	$s' = 1$	0.5	0.5	$s' = 1$	0.5	0.5
9	$s' = 0$	0.5	0.5	$s' = 0$	0.5	0.5
	$s' = 1$	0.5	0.5	$s' = 1$	0.5	0.5

All algorithms are compared using the following metrics: the average long-term reward (*ALR*), the percentage which a_0 is selected (%A_0), the average energy consumption (*AEC*) and the percentage of queries that meet the required confidence level (*%M*)[11] :

4.4.1.1 The Average Long-Term Reward

Fig. 4.4 compares the average long-term reward for each data uncertainty case. Case 1 depicts the best case scenario with complete observability, i.e. when the correlation model is most precise and error-free sensor readings are acquired from the WSN. It is seen that all algorithms produce high average long-term rewards. The reason is because with complete observability, the use of the model alone can meet the user's specified confidence bound. As sensor data uncertainty increases in other cases, WIT and FVI produce similar average long-term rewards and these two schemes outperform RA, HSA. In the case of setting 5 and the worst case of setting 9 (the least observability), all algorithms produce similar average long-term rewards. The reason is because none of the algorithms could satisfy the user's confidence requirements in presence of high data uncertainty, so all algorithms are highly penalized. Note that as the discount factor γ is varied from 0.1 to 0.9, WIT and FVI still outperform other schemes, however with a marginal increase in average long-term reward of no more than 1%. Therefore, the value of $\gamma = 0.9$ was used throughout this study. In addition, note that as the variance of the data increases, WIT and FVI still outperform other algorithms, however with decrease in average

[11] $\%M = \frac{\sum_n \text{accumulated queries that meet the confidence requirement in trial } n}{\text{total number of horizons} \times \text{total number of trials}} \times 100.$

long-term reward of no more than 2%. This is because the Markov property of data decreases with increase in variance of data.

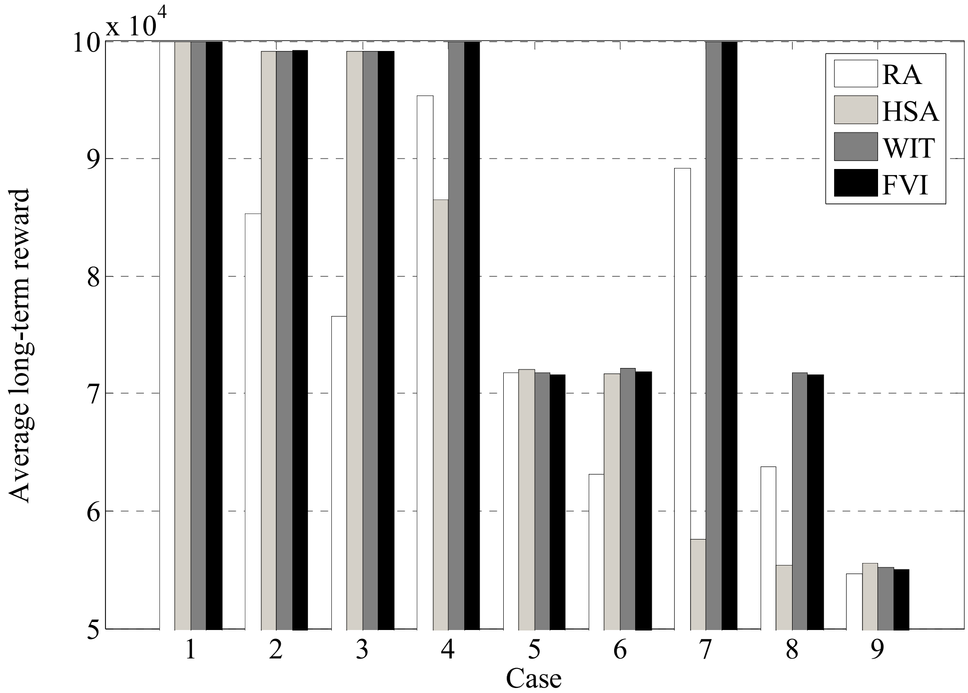

Figure 4.4 Average long-term reward for different cases of imprecise sensor data as shown in table 4.3.

4.4.1.2 The Percentage which Action 0 is Selected

Fig. 4.5 compares the percentage which a_0 is selected for different cases of data uncertainty. For the highest sensor data certainty case of setting 1, the BS observes the actual sensor readings and has a precise correlation model. Therefore, the model's confidence exceeds the user's specified confidence threshold. As a result, all algorithms are able to use the model alone to answer the user's query. In cases 3, 6 and 9, WIT and FVI favor acquiring sensor readings (a_1) from the WSN in order to increase confidence level in answering the user's query.

This is because the BS has an imprecise correlation model (cases 3, 6 and 9) and/or the sensor readings acquired from the WSN are more imprecise than the correlation model (cases 6 and 9). Note that in cases 2, 4, 5, 7 and 8, WIT and FVI selectively use the correlation model more often than HSA and RA, therefore attaining higher average long-term reward over HSA and RA.

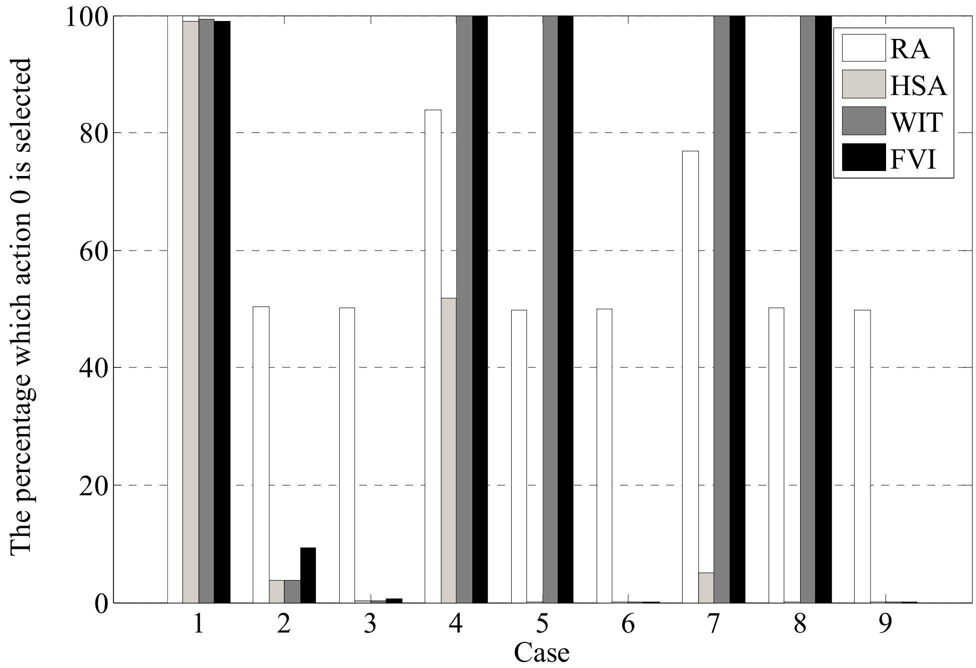

Figure 4.5 The percentage of average *a*0 is selected for different cases of imprecise sensor data as shown in table 4.3.

4.4.1.3 The Average Energy Consumption

Fig. 4.6 shows the average energy consumption in each data uncertainty case. In the best case scenario (case 1) where the BS has a precise correlation model and the WSN is error-free, it is seen that all algorithms do not consume any energy because the use of the correlation model alone can meet the user's specified confidence bound. For cases 4, 5, 7 and 8, where the WSN error increases and the BS still maintains a precise correlation model, results show that WIT and FVI consume less energy than other algorithms. The reason is because these algorithms prefer to use the correlation model than to acquire sensor readings to answer the user's query. This is either due to the accurate correlation model (cases 4 and 7) or a good correlation model (cases 5 and 8) in presence of an error-prone WSN. Note that for cases 2, 3, 6 and 9, the BS favors acquiring sensor readings from the WSN. It is seen that HSA, WIT and FVI consume a similar amount of energy, which is less than RA. This is either due to the high sensor data uncertainty (case 9) or the fact that the sensor reading is more accurate than the correlation model (cases 2, 3 and 6).

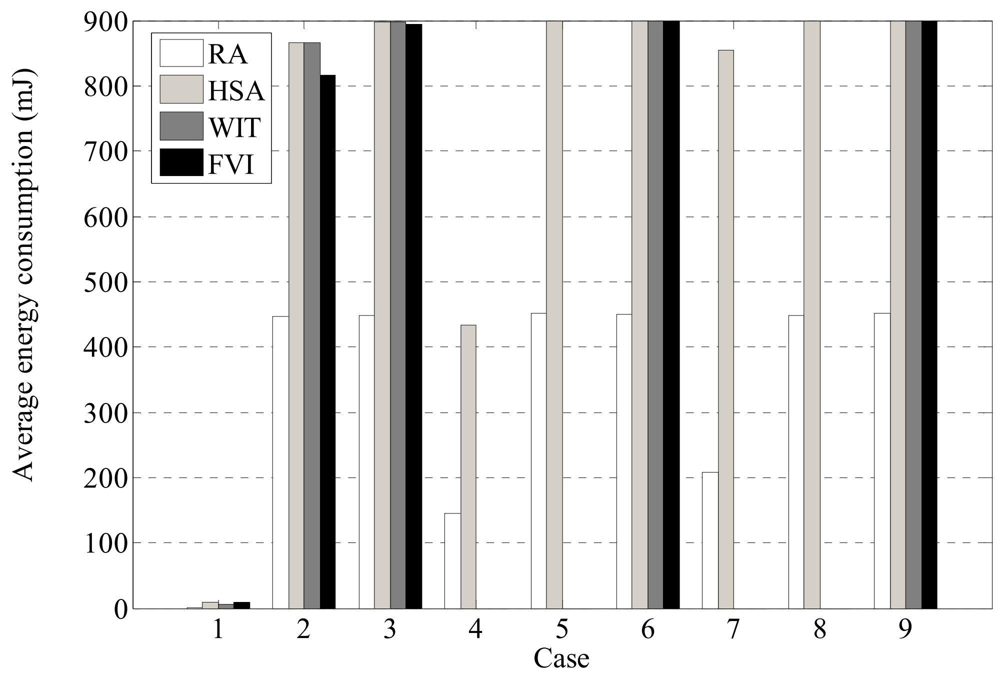

Figure 4.6 Average energy consumption for different cases of imprecise sensor data as shown in table 4.3.

4.4.1.4 The Percentage of Queries that Meet the Confidence Threshold

Fig. 4.7 shows the percentage of queries that meet the confidence threshold for different cases of sensor data uncertainty. Results show that WIT and FVI can meet the user's specified confidence requirement more frequently than other algorithms.

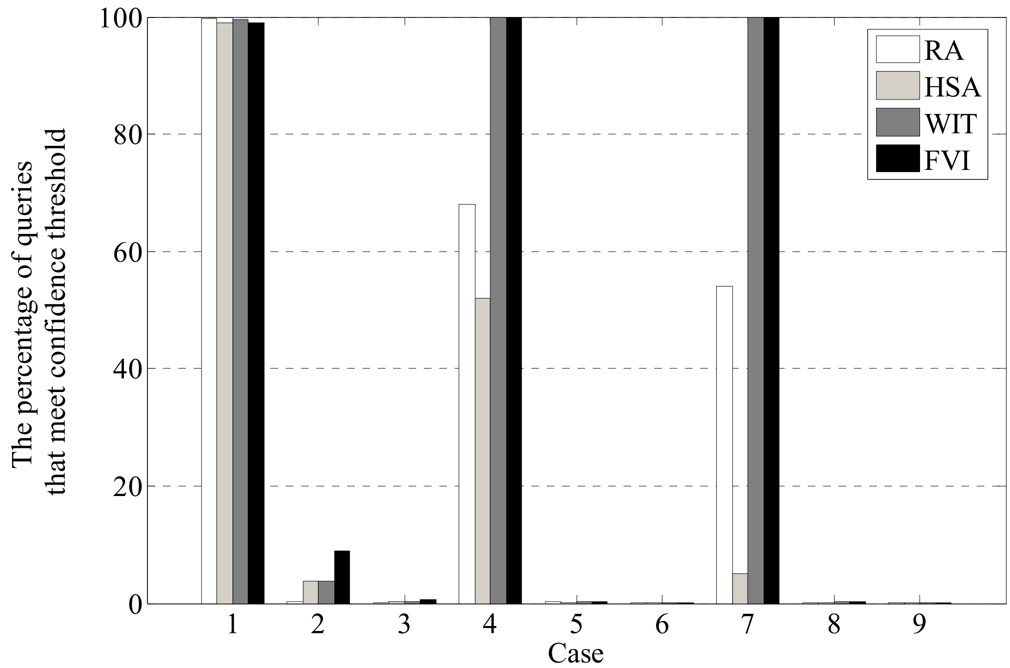

Figure 4.7 The percentage of queries that meet the required confidence level for different cases of imprecise sensor data as shown in table 4.3.

4.4.2 The Spatial and Temporal Correlation Network Study

From the previous section, it was demonstrated that the PPOMDP formulation (solved by FVI) gives a sensor selection policy close to the POMDP (solved by WIT). We then consider a network consisting of a base station (BS) and two temperature sensor nodes which are highly correlated. Note that the two sensors provide a simple conceptual framework to study the effects of spatial correlation. The correlation coefficient is equal to 0.9, meaning that the temperature readings from the two sensor nodes have a strong increasing linearly relationship. The temperature readings represent the state of the sensor readings. The temperature values are quantized into 2 subintervals, [0,20] and (20,40] °C, which are represented by integers 0 and 1, respectively. Therefore, the state space is given by $S = \{s_1, s_2, s_3, s_4\} = \{[0,0],[0,1],[1,0],[1,1]\}$. The data acquisition process starts when a user generates a query to the BS for temperature values of the environment. The query includes an error tolerance (ε) of 4 °C and a target confidence bound (δ) of 95% that specifies how much uncertainty the user is willing to tolerate. The confidence level of sensor *i*, $Conf_i = P\left(X_i \in [\mu_i - \varepsilon, \mu_i + \varepsilon]\right)$, is computed from Eq. (4.1) where μ_i is the sample mean of the attribute on sensor *i*. If $Conf_i \geq \delta$, $\forall i$, then the BS can directly estimate the temperature value from its correlation model to answer the user's query since the BS is sufficiently confident of its estimation. Since there are two random variables, X_1 and X_2, the correlation model maintained at the BS is a two-dimensional Gaussian pdf model. If $Conf_i < \delta$, then the BS is not sufficiently confident of its own estimation, and it must make a decision by selecting an action from a set of all possible actions $A=\{a_0,a_1,a_2,a_3\}=\{[0,0],[0,1],[1,0],[1,1]\}$. If the BS decides to estimate the temperature values of the two sensors to answer the user's query (a_0), then the sensors do not consume any energy. However, it is at the risk of obtaining an inaccurate estimate. If the BS decides to acquire reading(s) from a specified sensor such as sensor 1 (a_1), or sensor 2 (a_2) or both sensors (a_3), then the sensor incurs energy costs to tradeoff increased accuracy from acquiring the additional temperature reading(s). As in section 3.4, the energy cost model in this section is obtained from the datasheets of the Mica2 motes with the Crossbow MTS400 environmental sensor board.

We compared three algorithms, i.e., the randomized algorithm (RA) which makes a randomized decision, the heuristic search algorithm (HSA) which makes a decision to optimize the immediate (short-term) reward (Deshpande et al., 2004) and the fitted value iteration (FVI) which makes a decision to optimize *the average long-term* reward (Brooks, Makarenko, Williams and D-Whyte, 2006). All algorithms are compared using the four metrics described in section 4.4.1.

We simulated our network on Microsoft Visual C++. Each trial is carried out over a finite horizon of 200 time steps, meaning that 200 queries are sent to the BS. The trials were repeated for 100 runs to attain a precision of 95% for all schemes. The sensor readings change states according to the transition probability model in table 4.4, which is obtained by quantizing sequences of temperatures from the two sensors into subintervals [0,20] and (20,40] °C. The temperature sequences which are used to obtain the probability transition entries for a_0, a_1, a_2, and a_3 in table 4.4, are generated from a two-dimensional Gaussian pdf model (a correlation model), $N(\vec{\mu},\mathbb{C})$, where $\vec{\mu}=[18,18]$ is the vector of mean temperature on sensor 1 and 2, respectively, and $\mathbb{C}$ is the temperature covariance matrix. We assume that state transition from a_0 is less likely than other actions since the temperatures are estimated from the correlation model, so the variance is the least for action a_0. The covariance matrices, $\begin{bmatrix} c_{x_1x_1} & c_{x_1x_2} \\ c_{x_2x_1} & c_{x_2x_2} \end{bmatrix}$, for a_0, a_1, a_2, and a_3 are given by $\begin{bmatrix} 1 & 1 \\ 1 & 1.2 \end{bmatrix}$, $\begin{bmatrix} 1.2 & 1.4 \\ 1.4 & 2 \end{bmatrix}$, $\begin{bmatrix} 2 & 2.2 \\ 2.2 & 3 \end{bmatrix}$, and $\begin{bmatrix} 2 & 2.5 \\ 2.5 & 4 \end{bmatrix}$, respectively. These covariance matrices give a correlation coefficient[13] of about 0.9.

The immediate rewards, $g(s,a,s',\theta)$, which the BS could obtain are shown in table 4.5. If the BS obtains correct readings from both sensors, then the following reward regime is used. The BS receives -2 for choosing correlation model estimation (a_0) when $Conf_i < \delta, \forall i$, in order to discourage the use of model when the BS is not sufficiently confident of its own estimation. The BS receives -0.9, which is the amount of energy consumed (in mJ) for choosing to acquire a reading from sensor 1 (a_1) or sensor 2 (a_2). Similarly, the BS receives -1.8, which is the amount of energy consumed (in mJ) for choosing to acquire readings from both sensors (a_3). However, if the BS observes incorrect (faulty) readings from both sensors, it receives -100 in order to penalize itself for acquiring faulty sensor readings. If the BS observes a faulty reading from only one sensor, then an additional -2 is

---

[13] The correlation coefficient, ρ_{xy}, is calculated by $\rho_{xy} = c_{xy}/c_{xx}c_{yy}$.

added to the reward the BS received from correctly reading both sensors. For example, if the BS decides to acquire a reading from sensor 1, then the BS first receives -0.9. After taking this action, if the BS observes an incorrect reading from the sensor, the BS receives an additional reward of -2. So, the total reward received at the BS is $-0.9-2=-2.9$ as shown in table 4.5.

Table 4.4 Probability Transition Matrices in the Spatial and Temporal Network

$p(s'\|s,a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$s=(0,0)$	0.95	0.03	0.00	0.02
$s=(0,1)$	0.95	0.03	0.00	0.02
$s=(1,0)$	0.95	0.03	0.00	0.02
$s=(1,1)$	0.95	0.03	0.00	0.02
$p(s'\|s,a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$s=(0,0)$	0.92	0.05	0.00	0.03
$s=(0,1)$	0.92	0.05	0.00	0.03
$s=(1,0)$	0.92	0.05	0.00	0.03
$s=(1,1)$	0.92	0.05	0.00	0.03
$p(s'\|s,a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$s=(0,0)$	0.86	0.06	0.01	0.07
$s=(0,1)$	0.86	0.06	0.01	0.07
$s=(1,0)$	0.86	0.06	0.01	0.07
$s=(1,1)$	0.86	0.06	0.01	0.07
$p(s'\|s,a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$s=(0,0)$	0.83	0.09	0.01	0.07
$s=(0,1)$	0.83	0.09	0.01	0.07
$s=(1,0)$	0.83	0.09	0.01	0.07
$s=(1,1)$	0.83	0.09	0.01	0.07

Table 4.5 Reward Structure in the Spatial and Temporal Network

$g(s,a_0,s')$	$\theta=(0,0)$	$\theta=(0,1)$	$\theta=(1,0)$	$\theta=(1,1)$
$\forall s,s'=(0,0)$	-2	-4	-4	-100
$\forall s,s'=(0,1)$	-4	-2	-100	-4
$\forall s,s'=(1,0)$	-4	-100	-2	-4
$\forall s,s'=(1,1)$	-100	-4	-4	-2
$g(s,a_1,s')$ or $g(s,a_2,s')$	$\theta=(0,0)$	$\theta=(0,1)$	$\theta=(1,0)$	$\theta=(1,1)$
$\forall s,s'=(0,0)$	-0.9	-2.9	-2.9	-100
$\forall s,s'=(0,1)$	-2.9	-0.9	-100	-2.9
$\forall s,s'=(1,0)$	-2.9	-100	-0.9	-2.9
$\forall s,s'=(1,1)$	-100	-2.9	-2.9	-0.9
$g(s,a_3,s')$	$\theta=(0,0)$	$\theta=(0,1)$	$\theta=(1,0)$	$\theta=(1,1)$
$\forall s,s'=(0,0)$	-1.8	-3.8	-3.8	-100
$\forall s,s'=(0,1)$	-3.8	-1.8	-100	-3.8
$\forall s,s'=(1,0)$	-3.8	-100	-1.8	-3.8
$\forall s,s'=(1,1)$	-100	-3.8	-3.8	-1.8

Table 4.6 shows 9 different cases of data certainty. Each entry represents the average probability that the BS obtains correct temperature values from taking a particular action. The higher the probability obtained by querying the sensors (a_1, a_2, a_3), the less error the WSN has, such as in case 2, 3, 6. The higher the probability obtained by estimating from the correlation model (a_0), the better (more precise) the correlation model estimation becomes, such as in case 4, 7, 8.

Table 4.6 Probability of Data Uncertainty in the Spatial and Temporal Network (see **Appendix III**)

Case	a_0	a_1	a_2	a_3
1	1.00			
2	0.45	1.00	1.00	1.00
3	0.29			
4	1.00			
5	0.45	0.46	0.48	0.50
6	0.29			
7	1.00			
8	0.45	0.30	0.29	0.29
9	0.29			

FVI represents belief points by a two-dimensional Gaussian distribution with a diagonal covariance matrice, $N\left(\bar{\mu},\mathbb{C}\right)$. As result, a belief point is four-dimensional, i.e., it can be represented by a vector, $\Phi_i = \left[\mu_1, \mu_2, c_{11}, c_{22}\right]$, $1 \le i \le |\beta|$, consisting of two means and two variances, where B is the set of belief points, and $|\beta|$ is the cardinality of set β. The means and variances of the belief points are quantized into two levels. Therefore, the total number of belief points is 16. The freudenthal triangulation[14] (Brooks, 2006) is used for function approximation in line 12 of the FVI pseudo-code. Note that other function approximation methods such as multilinear interpolation (Davies, 1996) can be used. However, Davies method requires computation time of $O\left(2^d\right)$ whereas the computational required for freudenthal triangulation is only $O\left(d \log d\right)$ time.

[14] The details of the Freudenthal triangulation can be found in **Appendix I**.

4.4.2.1 The Average Long-Term Reward

Fig. 4.8 shows the average long-term reward for different cases of data certainty. Case 1 represents the best case scenario, i.e., when the BS has a good (precise) correlation model and the WSN is error-free. It is seen that all algorithms produce high average long-term rewards. The reason is because the use of the correlation model alone can satisfy the user's specified confidence level. As sensor data uncertainty increases in other cases, FVI outperforms other schemes. In the worst case scenario with the least data certainty as shown in case 9, all algorithms achieved the least average long-term reward. This is because none of the algorithms could satisfy the user's confidence requirements in presence of high data uncertainty, so all algorithms are highly penalized.

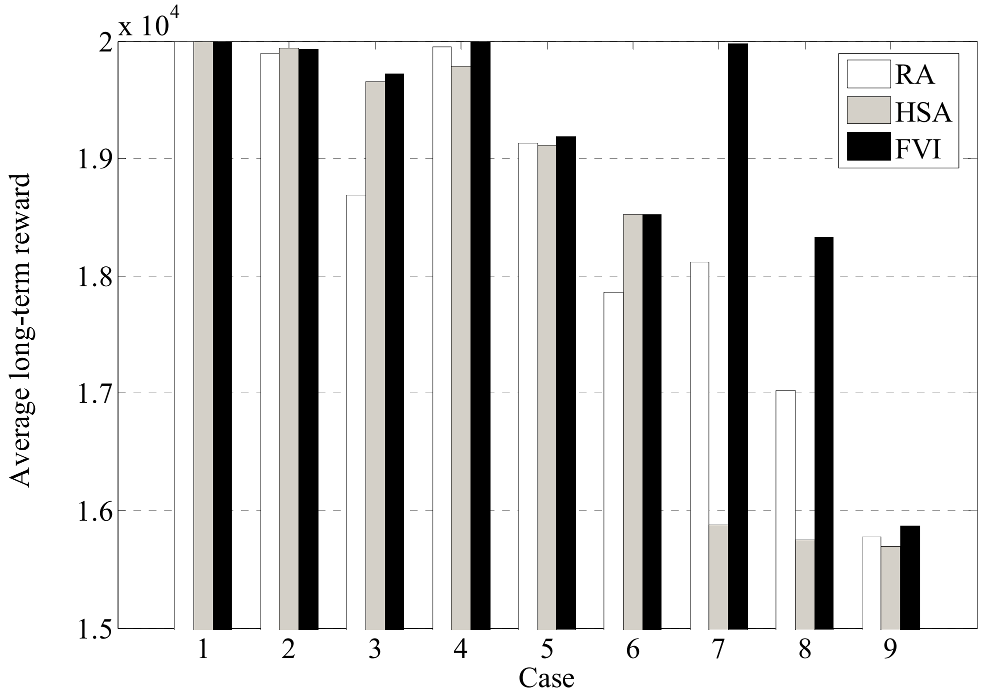

Figure 4.8 Average long-term reward for different cases of data uncertainty as shown in table 4.6.

4.4.2.2 The Average Energy Consumption

Fig. 4.9 shows the average energy consumption for each case. For case 1 and 2 where the WSN contains no error and the BS always obtains correct sensor readings, the BS is highly confident in estimating the answer to the query. As a result, all algorithms use the correlation model alone when the model is good (case 1), or acquire sensor readings occasionally to increase confidence level when the correlation model is reasonably good (case 2). In case 4, 5, 7 and 8, where the WSN becomes increasingly erroneous, FVI consumes less energy than other algorithms. The reason is because the FVI algorithm prefers the use of model over acquiring sensor readings to answer the query, resulting in energy consumption savings. This is due to the good (case 4 and 7) and reasonably good correlation models (case 5 and 8) of the sensor data maintained at the BS in presence of an error prone WSN. Note that for cases 3, 6 and 9, FVI consumes slightly more energy than the RA. This is either due to the fact that querying sensors is more accurate than estimating from the correlation model (case 3 and 6), or use of a poor (imprecise) correlation model in a highly error-prone WSN (case 9).

4.4.2.3 The Percentage which Action 0 is Selected

Fig. 4.10 shows the percentage which a_0 is selected for different cases of data certainty. For the highest sensor data certainty scenario (case 1), the all algorithms use the correlation model alone to answer the user’s query. The reason is because the model’s confidence exceeds the user’s specified confidence requirement. For case 2, 4, 5, 7 and 8, FVI selectively uses the correlation model more frequently, therefore attaining low energy consumption and higher average long-term reward than HSA and RA. For cases 3, 6, and 9, FVI rarely uses the correlation model to estimate the query answer and favors acquiring sensor readings (a_1, a_2, a_3) from the WSN instead in order to increase the confidence level at the BS. This is due to the inaccurate correlation model (case 9) or the fact that the sensor readings acquired from the WSN are more accurate than the correlation model maintained at the BS (case 3 and 6).

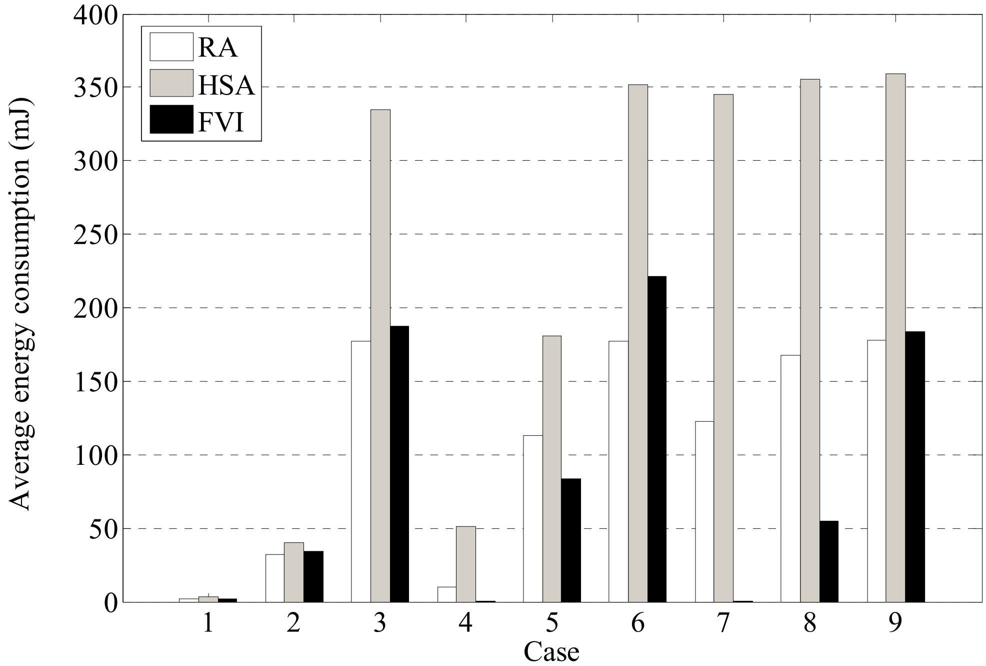

Figure 4.9 Average energy consumption for different cases of data uncertainty as shown in table 4.6.

4.4.2.4 The Percentage of Queries that Meet the Confidence Threshold

Fig. 4.11 shows the percentage of queries that meet the confidence threshold for each data certainty case. Results show that FVI meets the user's specified confidence bound more frequently than other algorithms.

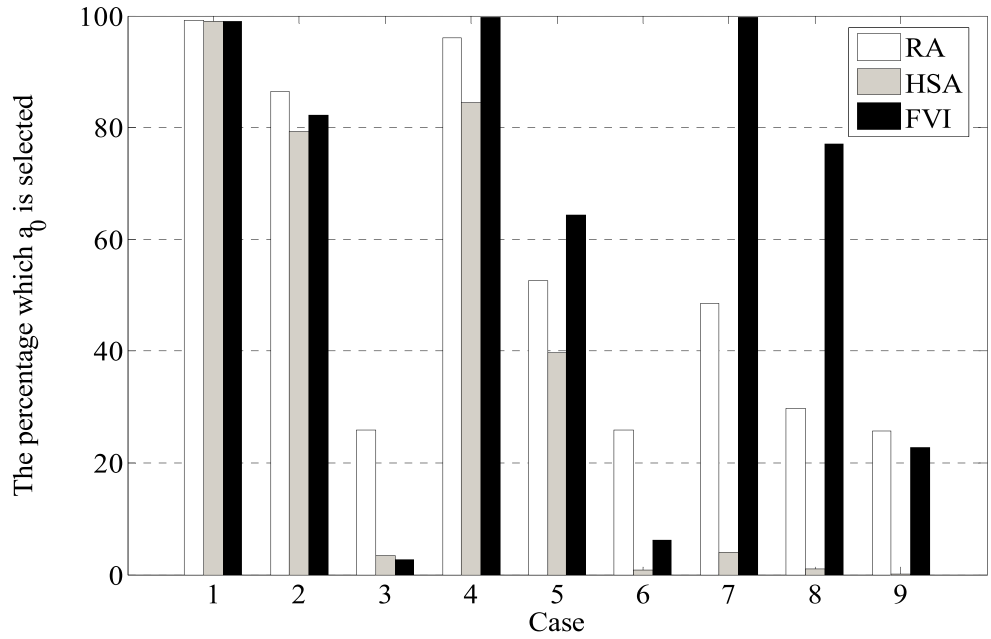

Figure 4.10 The percentage which a0 is selected for different cases of data uncertainty as shown in table 4.6.

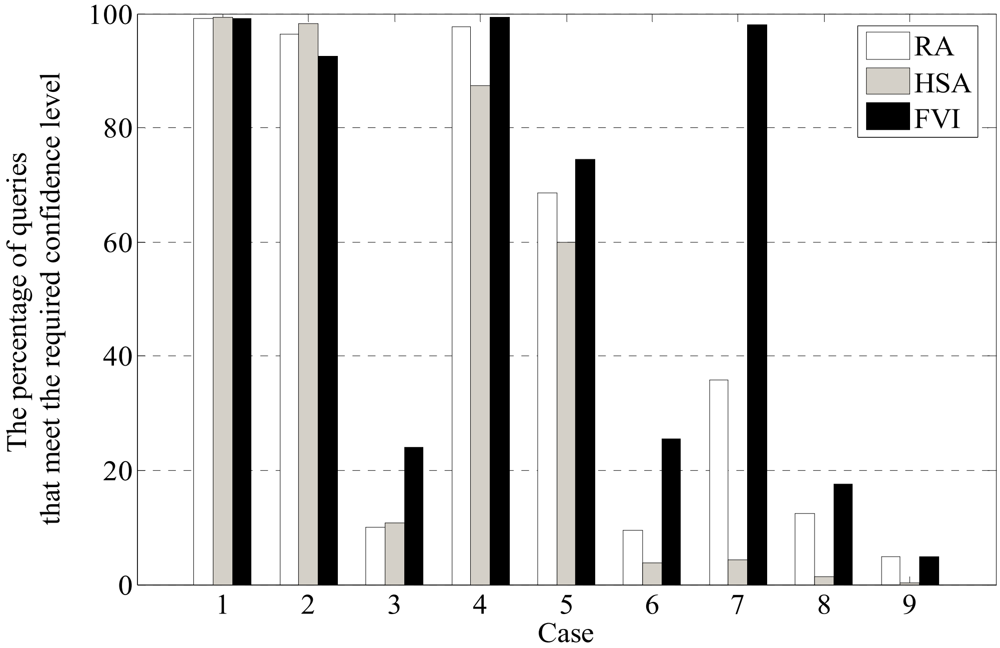

Figure 4.11 The percentage of queries that meet the required confidence bound for different cases of data uncertainty as shown in table 4.6.

4.4.3 Spatial-Temporal Correlation Model

Figures 4.12-4.14 compare results between using both the spatial and temporal correlation model and using the temporal correlation model alone in terms of normalized average long-term reward[15], normalized average energy consumption[16] and the percentage of queries that meet the required confidence level. Fig.4.12 shows that using the both spatial and temporal correlation model (FVI-ST) can attain higher normalized average long-term reward than using the temporal correlation model alone (FVI-T). In Fig. 4.13, it can be shown that using the spatial and temporal correlation model can reduce more average energy consumption than using only the temporal correlation model. In addition, using the spatial and temporal correlation model can meet the user's specified confidence threshold more often when compared using only the temporal correlation model as shown in Fig. 4.14. This is because the temporal correlation model can reduce the frequency for acquiring data from each sensor in the sensor network and the spatial correlation model can help reduce the density for acquiring data from nearby area. Therefore, these preliminary results suggest that using the both correlation models is better than using a single correlation alone.

---

[15] $Normalized\ average\ long\_term\ reward = 100 - \left(\frac{average\ long\_term\ reward}{\min.\ average\ long\_term\ reward} \right) \times 100$

[16] $Normalized\ average\ energy\ consumption = \left(\frac{average\ energy\ consumption}{\max.\ average\ energy\ consumption} \right) \times 100$

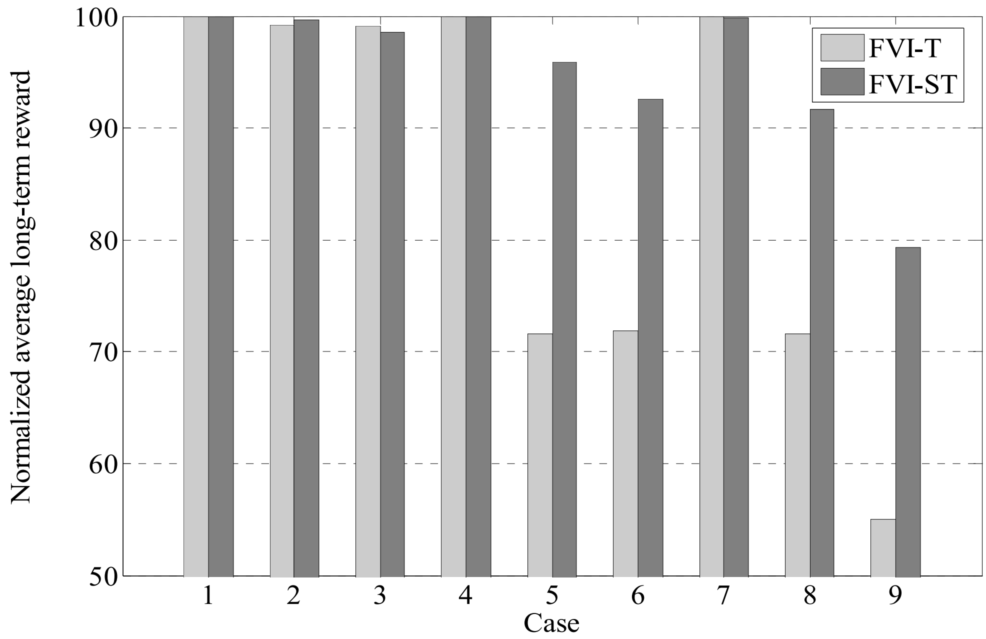

Figure 4.12 Normalized average long-term reward when using the temporal correlation alone and both spatial and temporal correlation.

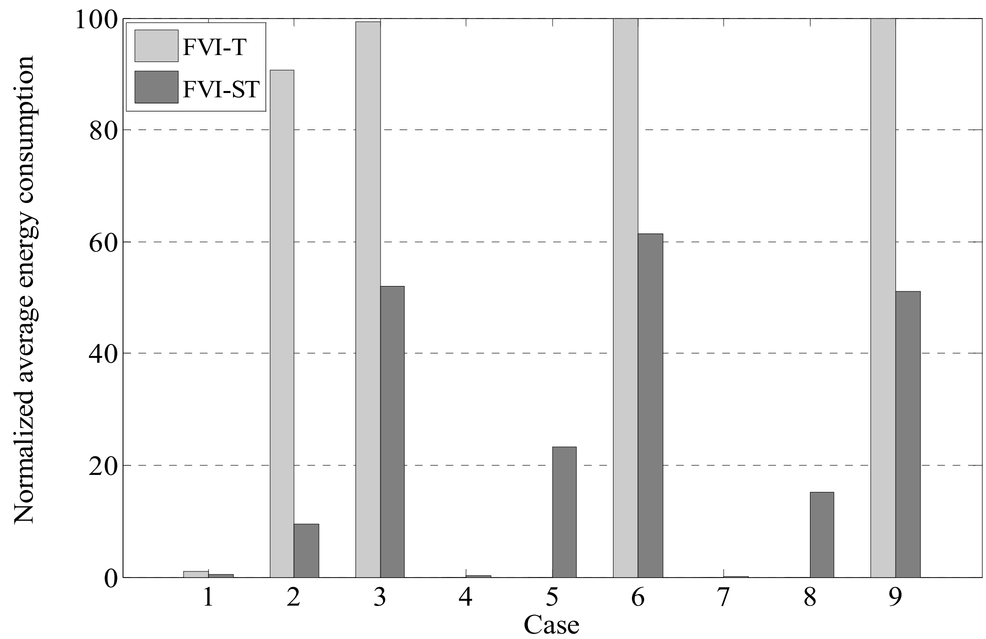

Figure 4.13 Normalized average energy consumption when using the temporal correlation alone and both spatial and temporal correlation.

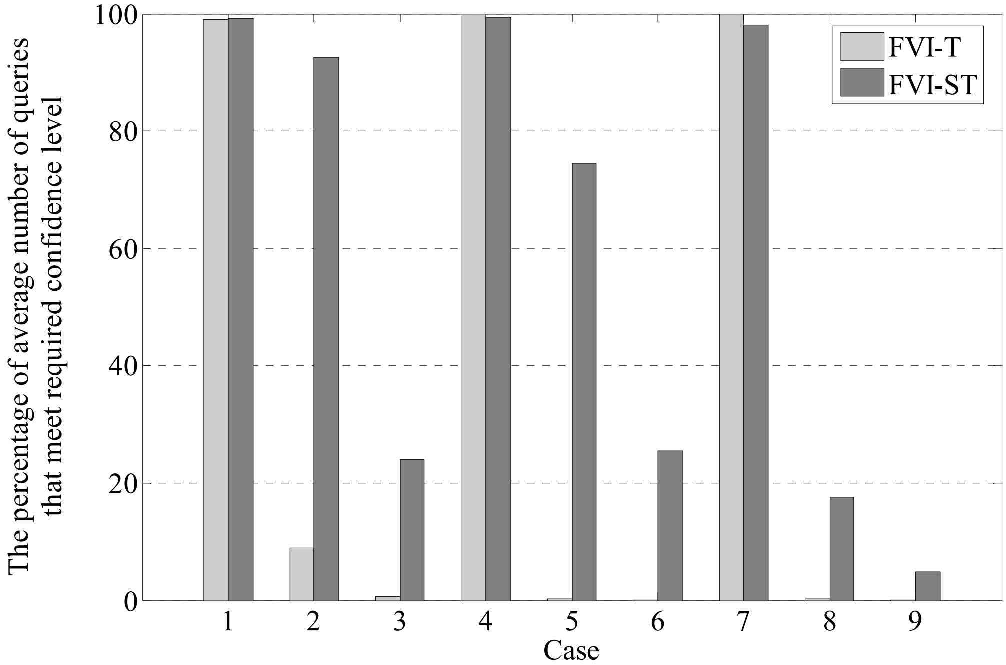

Figure 4.14 The percentage of average number of queries that meet the required confidence level when using the temporal correlation alone and both spatial and temporal correlation.

4.4.4 Storage and Computational Time

WIT requires a complete knowledge of the state transition and observation transition probabilities. The transition probability matrix requires a number of $|S|\times|S|\times|A|$ parameters and the observation transition matrix requires a number of $|\Theta|\times|S|\times|A|$ parameters, where $|S|$, $|A|$ and $|\Theta|$ are the cardinality of the state, action and observation spaces, respectively. For example, in the both spatial and temporal correlation study with 2 sensors and with 2 quantization levels of temperature, the total number parameters needed for the transition probability matrix is $4\times4\times4=64$ parameters and the total number parameters needed for the observation transition matrix is $4\times4\times4=64$ parameters. Hence, the total number of parameters required for WIT is *128 parameters*. In addition, the time WIT spends to select an action online is approximately 0.0008s in the temporal correlation study by using a HP Compaq dx6120 MT PC model, with an Intel 3 GHz Pentium4 CPU and 4 GHz RAM. Note that such duration increases in polynomial with $|S|$, $|A|$, $|\Theta|$, $|\nu_{t-1}|$, and $|\Omega_t^a|$ (Kaelbling, Littman an Cassandra, 1998). On the other hand, FVI does not need a complete knowledge of these two matrices (as it computes them by means of Monte Carlo integration). However, FVI still needs storage for its belief point set (β) and the value functions of each belief point, (Ψ). In this thesis, we use a low dimensional vector of sufficient statistics, Φ, to represent each *belief point*. In particular, we represent each belief point in a parametric form of a Gaussian distribution, whose parameters include the means and the variances of each attribute. Hence, the total number of parameters required for the execution of FVI is $|\beta|\times|\Phi|+|\Psi|$ parameters, where β, Φ and Ψ are the sizes of belief points in sufficient statistics space, sufficient statistics and belief point value functions. For example, in the both spatial and temporal correlation study with 2 sensors and with 2 quantization levels of the mean and variance, the total number parameters needed for FVI is $(16\times4)+16=$ *80 parameters*. It can be seen that FVI requires the storage memory less than WIT. Alternative ways to further reduce the number of parameters and enhance the scalability of the framework can be found in section 5.2.1.

The computational complexity of the algorithm is $O\left(|\beta||A||\Theta|C_{\mathrm{EV\,AL}}\left(F_\beta,|\beta|\right)C_{\mathrm{EV\,AL}}(\Phi)\right)$ where $|A|$ and $|\Theta|$ are the number of actions and observations, $C_{\mathrm{EV\,AL}}\left(F_\beta,|\beta|\right)$ is the complexity of evaluating the function approximator, and $C_{\mathrm{EVAL}}(\Phi)$ is the complexity of evaluating the belief state update function, respectively (Brooks, Makarenko, Williams and D-Whyte, 2006). Since the function approximator and the belief update function are in the center of the loop and will hence be executed frequently, it is imperative that these functions have low complexity. In our algorithm, the belief update function can be analytically determined by simply averaging the current and the most recent values of the means and variances. We also experimentally evaluated the computational time of FVI as well using a HP Compaq dx6120 MT PC model, with an Intel 3 GHz Pentium4 CPU and 4 GHz RAM. The time the FVI spends to decide an action online is approximately 0.2s and 2.2s for the temporal correlation study and the both spatial and temporal correlation study. This is the computational time mainly required to estimate the state transition and observation transition probabilities by means of Monte Carlo integration.

4.5 Conclusions

The contribution in this chapter is two-fold. Firstly, a parametric partially observable Markov decision process (PPOMDP) framework is applied to select sensors in a data acquisition problem which supports probabilistic confidence requirements in error prone WSNs. As most sensor readings are inherently continuous-valued, such a formulation enables us to cast the data acquisition problem as a continuous-state POMDP which is more realistic than a discrete state POMDP. Secondly, an existing tool which is used to obtain PPOMDP solutions, called the fitted value iteration (FVI), is applied to solve for a good sensor selection scheme in the long run to collect data from an error prone WSN, while the probabilistic assurance requirements on the sensor readings are still satisfied.

Numerical results show that when only temporal correlation model is considered, FVI can achieve good average long-term rewards, with reasonable average energy consumption and provide probabilistic guarantees on the query result more often when compared to other existing algorithms. Furthermore, when both spatial and temporal correlation models are considered, FVI still outperform other algorithms. Moreover, the numerical results in this chapter also suggest that both spatial and temporal correlation model when incorporated together outperform the temporal correlation model alone.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this thesis, we proposed a framework based on a partially observable Markov decision process (POMDP) which can satisfy probabilistic data quality assurances for a data acquisition scheme in an error-prone wireless sensor network (WSN). The main contribution of this thesis can be classified into two parts. The first contribution is formulating the sensor selection problem using a discrete-state POMDP framework. The witness algorithm which is used for determining a good sensor selection policy in the discrete-state POMDP is presented in **chapter 3**. The second contribution is formulating the sensor selection problem using a continuous-state POMDP framework. The fitted value iteration algorithm which is employed to solve for good sensor selection policy in the continuous-state POMDP is presented in **chapter 4**. The findings of this thesis can be summarized as follows.

5.1.1 Chapter 3

The purpose of this chapter is to formulate the data acquisition problem which can satisfy probabilistic confidence requirements of the acquired data in an error-prone wireless sensor network (WSN) as a discrete-state partially observable Markov decision process (POMDP) framework. To solve this problem, we applied the witness algorithm to find a good sensor selection policy. In the numerical study, a comparison was made with the randomized algorithm and heuristic search algorithm under various degrees of data uncertainty. Simulation results showed that the proposed scheme can determine a sensor selection policy that can attain high long-term average rewards and still satisfy the probabilistic data quality requirement. However, most sensor readings are inherently continuous-valued and the computational burden for the discrete-state POMDP increases rapidly with fine state discretisation. Therefore, discrete-state POMDP formulation is inappropriate for the data acquisition problem in many realistic situations.

5.1.2 Chapter 4

The purpose of this chapter is to extend the discrete-state POMDP framework in **chapter 3** to a more realistic scenario using the continuous-state POMDP framework. In particular, a type of continuous-state POMDP formulation called the parametric partially observable Markov decision process (PPOMDP) has been used to formulate the data acquisition problem which supports probabilistic confidence requirements in an error-prone WSN. The fitted value iteration (FVI) was employed to solve the data acquisition problem. In the numerical study, comparisons were made with the randomized algorithm, the heuristic search algorithm and the witness algorithm under various degrees of data uncertainty. Numerical results showed that when only the temporal correlation model is employed, the FVI can achieve good average long-term rewards, with reasonable average energy consumption and can satisfy the probabilistic requirements on the query result more often when compared to other existing algorithms. Moreover, when both spatial and temporal correlation models are considered, FVI still outperform other algorithms, i.e. FVI can achieve high average long-term rewards, with reasonable average energy consumption, and provide probabilistic guarantees on the query results more often when compared to other algorithms.

5.2 Recommendation for Future Work

5.2.1 Increase the Network Size

The main objective of this thesis is to show that the data acquisition problem in WSNs can be formulated with a POMDP framework. To evaluate the performance of this POMDP framework, we only considered the data acquisition problem under a small network scenario. However, WSNs are commonly used for environment and habitat monitoring which usually require a large number of sensor nodes. Therefore, we can extend the POMDP framework to a large network size for more practical applications. For instance, we can apply the clusterization method by Liu, Wu and Pei (2007) to enhance scalability for large networks by using the spatial correlation to dynamically partition the sensors into clusters and the temporal correlation to consider how frequent a sensor reading should be collected in a cluster.

5.2.2 Extension to Multiple Attributes and Fine Quantization Levels

To study the POMDP framework concept for the data acquisition problem in WSNs, we only considered a simple network. Furthermore, it is assumed that each sensor can collect only a single attribute, or a single parameter of physical information in the environment. Recent sensor development allows sensor nodes to collect many attributes which may have correlation between different attributes as shown in Fig. 3.2. Therefore, instead of the actual queried attribute, the BS may be faced with the decision of acquiring other attributes which may consume less energy as shown in table 3.1. This issue has not been investigated explicitly in this thesis yet. In the multiple attributes case, the BS still maintains a correlation model as in the single attribute case. However, such correlation model would represent the relation between different attributes, instead of different (spatially distributed) sensors. Consequently, the problem can be solved by algorithms proposed in this thesis. Additionally, the attribute values were quantized into only two levels in this thesis. As most attributes are real-valued, we can extend the POMDP framework to study WSNs which consist of multiple-attribute sensors and finer quantization levels.

5.2.3 Study Sensor Node Mobility

In this thesis, we focused on the environment monitoring application in WSNs where the location of the sensor is fixed. However, in many applications sensors are mobile such as in search and rescue operations and military applications. In such applications, sensor readings may frequently give irrelevant information to the user and waste resource consumption. For example, in an application where mobile sensors are used for explosives search, most mobile sensors may not be able to locate the explosive device. However, search data are reported to the user and consume significant amount of resource. Therefore, a data acquisition scheme which efficiently collects sensor readings is needed. In the future work, we can extend the framework to support the mobile sensor node scenario.

5.2.4 Improve Monte Carlo Integration Performance

The main objective of **chapter 4** is to show that the data acquisition problem in WSNs can be formulated using a parametric POMDP framework. To solve for a sensor selection plan, the FVI algorithm is used. In this algorithm, a simple Monte Carlo integration is applied to estimate the sufficient statistics observation probability and the sufficient statistics reward function in Eq. (4.9) and (4.11), respectively. The computation of this integration technique has a tradeoff between time and accuracy. Therefore, a significant direction worthwhile exploring is to improve the efficiency of the Monte Carlo integration to reduce the computational time and cater more real-time applications.

5.2.5 Study Performance with Raw Data

The main objective of this thesis is to show that the data acquisition problem in WSNs can be formulated as a POMDP framework. We only simulated our data acquisition problem and modeled the sensor readings with a time-varying multivariate Gaussian distribution to capture the inherent spatial and temporal correlation of WSNs. Therefore, an important future research is to extend the framework either to employ raw data collected from the field measurement which could be used for training the transition models, or implement the framework in an actual sensor network.

REFERENCES

Brooks, A. (2007). Parametric POMDPs for planning in continuous state spaces. **Thesis on Doctor of Philosophy**, University of Sydney.

Brooks, A., Makarenko, A., Williams, S., and D-Whyte, H. (2006). Parametric POMDPs for planning in continuous state spaces. **Robotics and Autonomous Systems**, vol. 54, pp. 887-897.

Cardei, M. and Du, D. Z. (2005). Improving wireless sensor network lifetime through power aware organization. **Wireless Networks**, vol. 11, pp. 333-340.

Cassandra, A. R. (1995). Optimal policies for partially observable Markov decision processes. **Technical Report CS-94-14**, Brown University.

Chobsri, S. and Usaha, W. (2008) A POMDP framework for data acquisition in wireless sensor networks. **Proceedings of the International Conference on Electrical Engineering, Electronics, Computer, Telecommunications, and Information Technology**.

Davies, S. (1996) Multidimensional triangulation and interpolation for reinforcement learning. **Advances in Neural Information Processing Systems**, vol. 9, pp. 1005–1011.

Deshpande, A., Guestrin, C., Madden, S. R., Hellerstein, J. M. and Hong, W. (2004). Model-driven data acquisition in sensor networks. **Proceedings of the 30th International Conference on Very Large Scale Databases**.

Fuemmeler, J. A. and Veeravalli, V. V. (2008). Smart sleeping policies for energy efficient tracking in sensor networks. **IEEE Transactions on Signal Processing**, vol. 56, no. 5, pp. 2091-2101.

Galmes, S. (2006). Lifetime issues in wireless sensor networks for vineyard monitoring. **Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems**.

Gordon, G. J. (1995). Stable function approximation in dynamic programming. **Proceedings of International Conference on Machine Learning**.

Gupta, V., Chung, T., Hassibi, B. and Murray, R. M. (2004). Sensor scheduling algorithms requiring limited computation, **Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing**.

Isler, V. and Bajcsy, R. (2006). The sensor selection problem for bounded uncertainty sensing models. **IEEE Transactions on Automation Science and Engineering**, vol. 3, no. 4, pp. 372-381.

Kaelbling, L. P., Littman M. L. and Cassandra A.R. (1997). Planning and acting in partially observable stochastic domains. **Artificial Intelligence**, vol. 101, no. 1, pp. 99-134.

Liu, C., Wu, K. and Pei, J. (2007). 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, pp. 1010-1023.

Lu, J., Bao, L., and Suda, T. (2005). Coverage-aware sensor engagement in dense sensor networks. **Proceedings of the International Conference on Embedded and Ubiquitous Computing**.

Mahalik, N. P. (2007). **Sensor networks and configuration: Fundamentals, standards, platforms, and applications**. Springer.

Meliou, A., Krause, A., Guestrin, C. and Hellerstein, J. M. (2007). Nonmyopic informative path planning in spatio-temporal models. **Proceedings of the 22nd Conference on Artificial Intelligence**.

Munos, R. and Moore, A. (2002). Variable resolution discretization in optimal control. **Machine Learning**. Springer.

Polastre, J. R. (2003). Design and implementation of wireless sensor networks for habitat monitoring. **Master's thesis**, University of California at Berkeley.

Rezaeian, M. (2007). Sensor scheduling for optimal observability using estimation entropy. **Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops**.

Rowaihy, H., Eswaran, S., Johnson, M., Verma, D., Bar-Noy, A., Brown, T. and Porta, T. L. (2007). A survey of sensor selection schemes in wireless sensor networks. **Proceedings of the SPIE**.

Sadaphal, V. and Jain, B. (2006). Spread-based heuristic for sensor selection in sensor networks. **Proceedings of the 1st International Conference on Communication System Software and Middleware**.

Shih, K. P., Chen, Y. D., Chiang, C. W. and Liu, B. J. (2006). A distributed active sensor selection scheme for wireless sensor networks. **Proceedings of the 11th IEEE Symposium on Computers and Communications**.

Sutton, R. S. and Barto, A. G. (1998). **Reinforcemnt learning: An introduction**. The MIT press.

Wuthrich, B. (1994). A probabilistic query language. **Technical Report CS-94-8**, The Hong Kong University of Science and Technology.

Yavuz, M. and Jeffcoat, D. (2007). An analysis and solution of the sensor scheduling problem. **Advances in Cooperative Control and Optimization.** Springer-Verlag.

Zhang, C., Fang, S. and Li, Z. (2008). Data collecting in wireless sensor networks using spatial models. **Proceedings of the International Conference on Information and Automation**.

APPENDIX I

Freudenthal Triangulation

Freudenthal Triangulation

A Freudenthal triangulation (Brooks, 2007) allows an interpolation to be performed in $O(d \log d)$ time, examining only $d+1$ of the data points, while providing the two guarantees namely global continuity and fitting the data points exactly. Freudenthal triangulation is based on the division of each box into $d!$ hyper-triangles, or simplices. Figure A.1 shows the Freudenthal triangulation of two and three-dimensional spaces.

The triangulation of each box can be performed as follows. First, translate and scale the box such that it is the unit hypercube, with diagonally opposite corners lying on $(x_1, x_2, ..., x_d) = (0, 0, ..., 0)$ and $(1,1,...,1)$. Second, consider all possible paths from $(0, 0, ..., 0)$ to $(1,1,...,1)$ along the (axis-aligned) edges of the box. There are $d!$ such paths, each consisting of $d+1$ points. The convex hull of each path defines one of the $d!$ hyper-triangles making up the triangulation. Note that each hyper-triangle corresponds to one possible permutation p of $(1, 2, ..., d)$, and bounds the set of points satisfying

$$0 \le x_{p(1)} \le x_{p(2)} \le ... \le x_{p(d)} \le 1 \tag{A.1}$$

In other words, each hyper-triangle is defined by a permutation of the order in which dimensions are traversed in paths between opposing corners, and bounds the set of points whose coordinates obey a particular inequality relationship. Figure A.1(a) illustrates this with a two-dimensional example. Finally, re-scale and translate the set of hyper-triangles back to their original positions. It is possible to perform an interpolation using this triangulation without ever explicitly generating all $d!$ simplices.

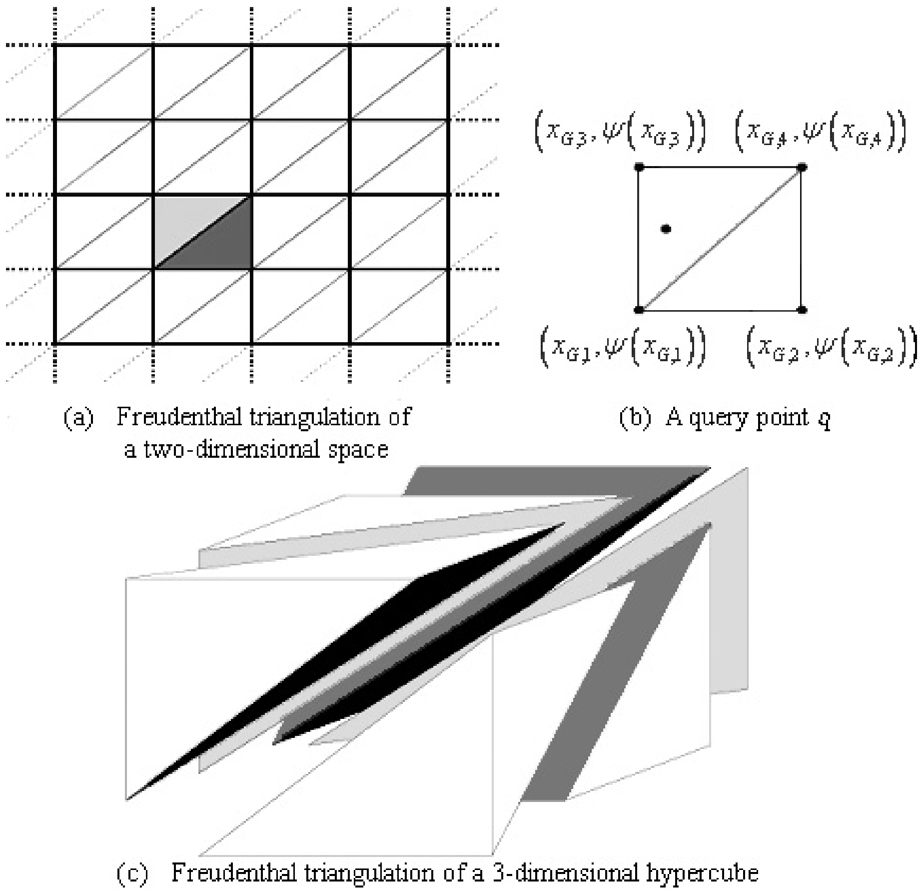

Figure A.1 Freudenthal triangulation.

Fig. A.1(a) shows the Freudenthal triangulation of a two dimensional space. The thick lines show the original hyper cubes. Each hypercube contains two hyper-triangles, or simplices, corresponding to the ordering of the two dimensions in paths from the lower-left to upper-right corners. Focusing on the shaded hyper-cube, the upper triangle (the path traverses y then x) contains all points for which

$y > x$, while the lower triangle (the path traverses x then y) contains all points for which $x > y$. The value of the two-dimensional query point q shown in Fig.A.1(b) can be expressed as a convex combination of the values $\psi\left(x_{G,1}\right)$, $\psi\left(x_{G,3}\right)$ and $\psi\left(x_{G,4}\right)$, stored at $x_{G,1}$, $x_{G,3}$, and $x_{G,4}$. Fig.A.1(c) shows how a three-dimensional hyper-cube is decomposed into 3! = 6 hyper-triangles.

Assuming a query point q defined by the coordinates $\left(x_1, x_2, ..., x_d\right)$, this interpolation can be performed as follows:

1. Translate and scale q's bounding box such that it is the unit hypercube, transforming the coordinates of q to $\left(x'_1, x'_2, ..., x'_d\right)$;

2. Sort the coordinates $x_1^{'}$ though $x_d^{'}$ from largest to smallest. This identifies the bounding simplex, or hyper-triangle (using Eq. A.1);

3. Produce a set of coefficients by expressing $\left(x'_1, x'_2, ..., x'_d\right)$ as a convex combination of the coordinates of the bounding simplex's $\left(d+1\right)$ corners; and

4. Use the coefficients determined in the previous step as the weights for a weighted sum of the data values stored at the corresponding corners.

For a detailed explanation of the third step, see Munos and Moore (2002). The computational cost of this algorithm is dominated by the sorting in the second step, which can be achieved in $O\left(d \log d\right)$ time.

APPENDIX II

Statistical Tests of Markov Property

Statistical Tests of Markov Property

The Markov property of the sensor attribute readings has been verified by statistical tests performed on sequences of sensor attribute readings generated by various Gaussian distributions. Based on our testing as shown in the figure below, it can be observed that Gaussian distributions with high variance have in significant margins of error, |e|, which is defined as the difference between the probability of occurrence of future states in the process given the present state, $P\left(S_{t+1} \middle| S_t\right)$, and the probability of occurrence of future states in the process given all previous states, $P\left(S_{t+1} \middle| S_t, \ldots, S_0\right)$. Therefore, the Markov property is assumed to hold.

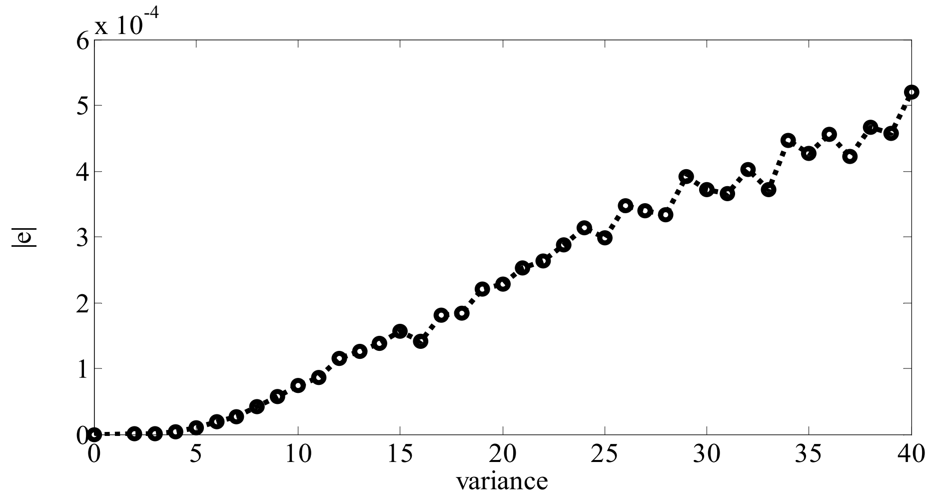

Figure B.1 The difference between the probability of occurrence of future states in the process given the present state, $P\left(S_{t+1} \middle| S_t\right)$, and the probability of occurrence of future states in the process given all previous states, $P\left(S_{t+1} \middle| S_t, \ldots, S_0\right)$.

APPENDIX III

Expanded Form of Table 4.6

Table C.1 Observation Matrices of Case 1

$r(\theta\|s',a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta\|s',a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	1	0	0	0	$\theta=(0,0)$	1	0	0	0
$\theta=(0,1)$	0	1	0	0	$\theta=(0,1)$	0	1	0	0
$\theta=(1,0)$	0	0	1	0	$\theta=(1,0)$	0	0	1	0
$\theta=(1,1)$	0	0	0	1	$\theta=(1,1)$	0	0	0	1
$r(\theta\|s',a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta\|s',a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	1	0	0	0	$\theta=(0,0)$	1	0	0	0
$\theta=(0,1)$	0	1	0	0	$\theta=(0,1)$	0	1	0	0
$\theta=(1,0)$	0	0	1	0	$\theta=(1,0)$	0	0	1	0
$\theta=(1,1)$	0	0	0	1	$\theta=(1,1)$	0	0	0	1

Table C.2 Observation Matrices of Case 2

$r(\theta\|s',a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta\|s',a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.79	0.06	0.10	0.05	$\theta=(0,0)$	1	0	0	0
$\theta=(0,1)$	0.33	0.27	0.11	0.28	$\theta=(0,1)$	0	1	0	0
$\theta=(1,0)$	0.30	0.13	0.28	0.29	$\theta=(1,0)$	0	0	1	0
$\theta=(1,1)$	0.20	0.19	0.14	0.47	$\theta=(1,1)$	0	0	0	1
$r(\theta\|s',a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta\|s',a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	1	0	0	0	$\theta=(0,0)$	1	0	0	0
$\theta=(0,1)$	0	1	0	0	$\theta=(0,1)$	0	1	0	0
$\theta=(1,0)$	0	0	1	0	$\theta=(1,0)$	0	0	1	0
$\theta=(1,1)$	0	0	0	1	$\theta=(1,1)$	0	0	0	1

Table C.3 Observation Matrices of Case 3

$r(\theta\|s',a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta\|s',a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.35	0.24	0.23	0.18	$\theta=(0,0)$	1	0	0	0
$\theta=(0,1)$	0.26	0.23	0.22	0.28	$\theta=(0,1)$	0	1	0	0
$\theta=(1,0)$	0.25	0.24	0.29	0.23	$\theta=(1,0)$	0	0	1	0
$\theta=(1,1)$	0.21	0.23	0.26	0.30	$\theta=(1,1)$	0	0	0	1
$r(\theta\|s',a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta\|s',a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	1	0	0	0	$\theta=(0,0)$	1	0	0	0
$\theta=(0,1)$	0	1	0	0	$\theta=(0,1)$	0	1	0	0
$\theta=(1,0)$	0	0	1	0	$\theta=(1,0)$	0	0	1	0
$\theta=(1,1)$	0	0	0	1	$\theta=(1,1)$	0	0	0	1

Table C.4 Observation Matrices of Case 4

$r(\theta \| s',a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \| s',a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	1	0	0	0	$\theta=(0,0)$	0.77	0.10	0.09	0.05
$\theta=(0,1)$	0	1	0	0	$\theta=(0,1)$	0.30	0.32	0.11	0.27
$\theta=(1,0)$	0	0	1	0	$\theta=(1,0)$	0.30	0.15	0.27	0.29
$\theta=(1,1)$	0	0	0	1	$\theta=(1,1)$	0.17	0.23	0.13	0.48
$r(\theta \| s',a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \| s',a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.78	0.09	0.08	0.05	$\theta=(0,0)$	0.79	0.09	0.08	0.04
$\theta=(0,1)$	0.29	0.33	0.10	0.27	$\theta=(0,1)$	0.28	0.36	0.08	0.27
$\theta=(1,0)$	0.31	0.11	0.30	0.28	$\theta=(1,0)$	0.30	0.12	0.30	0.28
$\theta=(1,1)$	0.14	0.21	0.13	0.53	$\theta=(1,1)$	0.12	0.23	0.10	0.54

Table C.5 Observation Matrices of Case 5

$r(\theta \| s',a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \| s',a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.79	0.06	0.10	0.05	$\theta=(0,0)$	0.77	0.10	0.09	0.05
$\theta=(0,1)$	0.33	0.27	0.11	0.28	$\theta=(0,1)$	0.30	0.32	0.11	0.27
$\theta=(1,0)$	0.30	0.13	0.28	0.29	$\theta=(1,0)$	0.30	0.15	0.27	0.29
$\theta=(1,1)$	0.20	0.19	0.14	0.47	$\theta=(1,1)$	0.17	0.23	0.13	0.48
$r(\theta \| s',a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \| s',a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.78	0.09	0.08	0.05	$\theta=(0,0)$	0.79	0.09	0.08	0.04
$\theta=(0,1)$	0.29	0.33	0.10	0.27	$\theta=(0,1)$	0.28	0.36	0.08	0.27
$\theta=(1,0)$	0.31	0.11	0.30	0.28	$\theta=(1,0)$	0.30	0.12	0.30	0.28
$\theta=(1,1)$	0.14	0.21	0.13	0.53	$\theta=(1,1)$	0.12	0.23	0.10	0.54

Table C.6 Observation Matrices of Case 6

$r(\theta \| s',a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \| s',a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.35	0.24	0.23	0.18	$\theta=(0,0)$	0.77	0.10	0.09	0.05
$\theta=(0,1)$	0.26	0.23	0.22	0.28	$\theta=(0,1)$	0.30	0.32	0.11	0.27
$\theta=(1,0)$	0.25	0.24	0.29	0.23	$\theta=(1,0)$	0.30	0.15	0.27	0.29
$\theta=(1,1)$	0.21	0.23	0.26	0.30	$\theta=(1,1)$	0.17	0.23	0.13	0.48
$r(\theta \| s',a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \| s',a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.78	0.09	0.08	0.05	$\theta=(0,0)$	0.79	0.09	0.08	0.04
$\theta=(0,1)$	0.29	0.33	0.10	0.27	$\theta=(0,1)$	0.28	0.36	0.08	0.27
$\theta=(1,0)$	0.31	0.11	0.30	0.28	$\theta=(1,0)$	0.30	0.12	0.30	0.28
$\theta=(1,1)$	0.14	0.21	0.13	0.53	$\theta=(1,1)$	0.12	0.23	0.10	0.54

Table C.7 Observation Matrices of Case 7

$r(\theta \mid s',a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \mid s',a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	1	0	0	0	$\theta=(0,0)$	0.35	0.24	0.23	0.19
$\theta=(0,1)$	0	1	0	0	$\theta=(0,1)$	0.26	0.25	0.22	0.26
$\theta=(1,0)$	0	0	1	0	$\theta=(1,0)$	0.22	0.25	0.31	0.22
$\theta=(1,1)$	0	0	0	1	$\theta=(1,1)$	0.22	0.25	0.24	0.28
$r(\theta \mid s',a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \mid s',a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.35	0.23	0.24	0.18	$\theta=(0,0)$	0.35	0.23	0.24	0.18
$\theta=(0,1)$	0.26	0.27	0.23	0.24	$\theta=(0,1)$	0.26	0.27	0.22	0.26
$\theta=(1,0)$	0.28	0.22	0.23	0.26	$\theta=(1,0)$	0.25	0.25	0..26	0.24
$\theta=(1,1)$	0.23	0.24	0.23	0.30	$\theta=(1,1)$	0.22	0.25	0.23	0.29

Table C.8 Observation Matrices of Case 8

$r(\theta \mid s',a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \mid s',a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.79	0.06	0.10	0.05	$\theta=(0,0)$	0.35	0.24	0.23	0.19
$\theta=(0,1)$	0.33	0.27	0.11	0.28	$\theta=(0,1)$	0.26	0.25	0.22	0.26
$\theta=(1,0)$	0.30	0.13	0.28	0.29	$\theta=(1,0)$	0.22	0.25	0.31	0.22
$\theta=(1,1)$	0.20	0.19	0.14	0.47	$\theta=(1,1)$	0.22	0.25	0.24	0.28
$r(\theta \mid s',a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \mid s',a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.35	0.23	0.24	0.18	$\theta=(0,0)$	0.35	0.23	0.24	0.18
$\theta=(0,1)$	0.26	0.27	0.23	0.24	$\theta=(0,1)$	0.26	0.27	0.22	0.26
$\theta=(1,0)$	0.28	0.22	0.23	0.26	$\theta=(1,0)$	0.25	0.25	0..26	0.24
$\theta=(1,1)$	0.23	0.24	0.23	0.30	$\theta=(1,1)$	0.22	0.25	0.23	0.29

Table C.9 Observation Matrices of Case 9

$r(\theta \mid s',a_0)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \mid s',a_1)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.35	0.24	0.23	0.18	$\theta=(0,0)$	0.35	0.24	0.23	0.19
$\theta=(0,1)$	0.26	0.23	0.22	0.28	$\theta=(0,1)$	0.26	0.25	0.22	0.26
$\theta=(1,0)$	0.25	0.24	0.29	0.23	$\theta=(1,0)$	0.22	0.25	0.31	0.22
$\theta=(1,1)$	0.21	0.23	0.26	0.30	$\theta=(1,1)$	0.22	0.25	0.24	0.28
$r(\theta \mid s',a_2)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$	$r(\theta \mid s',a_3)$	$s'=(0,0)$	$s'=(0,1)$	$s'=(1,0)$	$s'=(1,1)$
$\theta=(0,0)$	0.35	0.23	0.24	0.18	$\theta=(0,0)$	0.35	0.23	0.24	0.18
$\theta=(0,1)$	0.26	0.27	0.23	0.24	$\theta=(0,1)$	0.26	0.27	0.22	0.26
$\theta=(1,0)$	0.28	0.22	0.23	0.26	$\theta=(1,0)$	0.25	0.25	0..26	0.24
$\theta=(1,1)$	0.23	0.24	0.23	0.30	$\theta=(1,1)$	0.22	0.25	0.23	0.29

APPENDIX IV

List of Publications

List of Publication

Chobsri, S. and Usaha, W. A POMDP framework for data acquisition in wireless sensor networks. **The International Conference on Electrical Engineering, Electronics, Computer, Telecommunications, and Information Technology, 2008.**

Chobsri, S., Sumalai, W. and Usaha, W. A parametric POMDP framework for efficient data acquisition in error prone wireless sensor networks. **The 4th IEEE International Symposium on Wireless Pervasive Computing, 2009.**

Chobsri, S., Sumalai, W. and Usaha, W. Quality assurance for data acquisition in error prone WSNs. **The 1st IEEE International Conference on Ubiquitous and Future Networks, 2009.**

BIOGRAPHY

Miss Sunisa Chobsri was born on April 8, 1983 in Wangnamyen District, Sakeao Province. She finished high school education from Wangnamyen Witthayakom School, Sakeao Province. In 2002, she began studying her Bachelors degree at the School of Telecommunication Engineering, Institute of Engineering Suranaree University of Technology, Nakhon Ratchasima Province. After graduating, she continued to study for a Masters degree at the School of Telecommunication Engineering, Institute of Engineering, Suranaree University of Technology.