Received signal strength based localization of a wireless sensor network using particle swarm optimization

Nguyen, Hong Anh

2010

Nguyen, H. A. (2010). Received signal strength based localization of a wireless sensor network using particle swarm optimization. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/46768>

<https://doi.org/10.32657/10356/46768>

Nanyang Technological University

Downloaded on 26 Apr 2023 22:24:41 SGT

RECEIVED SIGNAL STRENGTH BASED LOCALIZATION OF A WIRELESS SENSOR NETWORK USING PARTICLE SWARM OPTIMIZATION

NGUYEN HONG ANH

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering

2010

Acknowledgement

It has been my distinct honor and pleasure to work with Assoc. Prof Low Kay Soon during my MEng studies at the Nanyang Technological University. He has made available his support in a number of ways enabling me the knowledge and facilities to understand and broaden the subject. This thesis would not have been possible unless he had extended his assistance from the initial to the final stage. I have learnt much from him through his guidance and inspiration.

I owe my deepest gratitude to Ms Lim-Tan Geok Lan and Ms Ong-Lim Guek Har from the Control Engineering Laboratory for their numerous supports throughout the project. Their constant help has given me convenience and comfort during the project.

It is an honor for me to work with Mr. Guo Hao. His share of knowledge and assistance is invaluable for my research.

I would like to thank my colleagues – Nora and Nazri in the Intelligent System Centre for helping me with all the administrative matters. I would like to show my gratitude to my colleagues outside the group for their guidance. Their work has inspired me to open for other approaches.

I would like to thank my parents and my sister for their encouragements and beliefs. Without their support I would not have been what I am today.

Table of Contents

Acknowledgement..... ii

Summary xi

Chapter 1 Introduction..... 1

1.1 Wireless sensor networks and applications 1

1.2 Overview of localization techniques..... 2

1.3 Motivation..... 4

1.4 Objectives 7

1.5 Main contributions..... 8

1.6 Organization of thesis 9

Chapter 2 Literature review of localization techniques for wireless sensor networks..... 12

2.1 Introduction..... 12

2.2 Range-free localization 13

2.3 Range-based localization 17

2.3.1 Time-Of-Arrival (TOA)..... 17

2.3.2 Angle-Of-Arrival (AOA)..... 18

2.3.3 Received Signal Strength (RSS)..... 20

2.3.4 Triangulation, trilateration and maximum likelihood..... 23

2.4 Other localization techniques..... 24

2.4.1 Kalman filter 25

2.4.2 Monte Carlo localization 26

2.4.3 Multi-Dimensional Scaling (MDS) 27

2.5 Conclusion 28

Chapter 3 Real-time localization using Particle Swarm Optimization with log-barrier constraint..... 29

3.1 Introduction..... 29

3.2 System model..... 30

3.2.1 Communication path loss modeling 31

3.3	Problem formulation	32
3.4	Constrained nonlinear optimization using log-barrier method	35
3.4.1	Popular nonlinear optimization approaches	35
3.4.2	Optimization with log-barrier approach	39
3.4.3	Particle swarm optimization	42
3.5	Test case	46
3.5.1	Standard PSO versus PSO with classical log-barrier	46
3.5.2	Modified barrier function versus classical barrier function	48
3.6	Conclusion	50
Chapter 4 Performance analysis and parameters tuning of the proposed		
	algorithm.....	52
4.1	Introduction.....	52
4.2	PSO convergence	53
4.3	Effect of PSO parameters on localization performance	56
4.4	Localization performance under different RSSI standard deviations	61
4.5	Effect of barrier parameter on localization accuracy	64
4.6	Comparison between PSO – log barrier and memetic algorithms	66
4.7	Performance comparison	71
4.8	Conclusion	76
Chapter 5 Prototyping and experimental results.....		78
5.1	Introduction.....	78
5.2	Experimental Prototype	79
5.2.1	Serial interface protocol.....	82
5.2.2	Zigbee protocol	84
5.3	Data communication in the user application.....	88
5.4	Embedded system's memory constraint and execution time	91
5.5	RSSI measurements	94
5.6	Experimental results	100
5.7	Conclusion	104

Chapter 6 Localization using Particle Swarm Optimization with log-barrier

constraint for a general network 105

6.1 Introduction..... 105

6.2 Selection method..... 106

6.3 Case study 111

6.4 Large scale simulation study..... 114

6.5 Experiment..... 119

6.6 Conclusion 123

Chapter 7 Conclusion and future work 125

List of publication 128

Reference 129

List of figures

Fig. 2.1	Classification of localization techniques.....	12
Fig. 2.2	Sketch map of the bounding box algorithm	14
Fig. 2.3	Sketch map of the polygon-based algorithm.....	15
Fig. 2.4	Sketch map of the DV-hop algorithm	16
Fig. 2.5	Sketch map of HA-A2L	19
Fig. 2.6	Sketch map of triangulation	24
Fig. 3.1	Wireless sensor network.....	31
Fig. 4.1	PSO convergence with $\sigma = 3\text{dB}$	54
Fig. 4.2	PSO convergence with $\sigma = 1\text{dB}$	55
Fig. 4.3	PSO convergence with $\sigma = 5\text{dB}$	55
Fig. 4.4	PSO convergence with $\sigma = 7\text{dB}$	56
Fig. 4.5	PSO performance versus population size.....	57
Fig. 4.6	Execution time versus c_1 and c_2	59
Fig. 4.7	Localization error versus c_1 and c_2	60
Fig. 4.8	Estimation error at different locations ($\sigma = 1\text{dB}$)	62
Fig. 4.9	Estimation error at different location ($\sigma = 3\text{dB}$).....	62
Fig. 4.10	Estimation error at different locations ($\sigma = 5\text{dB}$)	63
Fig. 4.11	Average error versus standard deviation	64
Fig. 4.12	Average error versus barrier parameter μ with $\sigma = 3\text{dB}$	65
Fig. 4.13	Simulation flowchart for multilateration approach	73
Fig. 4.14	Localization error versus RSSI standard deviation for different location methods.....	74
Fig. 4.15	Average localization error of the proposed method versus α and σ	75
Fig. 4.16	Average error versus σ	75
Fig. 5.1	Network setup	79
Fig. 5.2	Experiment site.....	80
Fig. 5.3	Block diagram of the communication and measurement	80
Fig. 5.4	System data flow diagram in a UART-interface environment.....	83
Fig. 5.5	The layers of the Zigbee stack.....	85
Fig. 5.6	Zigbee protocol device types [91].....	86
Fig. 5.7	Illustration of the TDMA communication cycle	88
Fig. 5.8	Latencies involved in the process of measuring RSSI	90

Fig. 5.9	Latencies involved with two consecutive measurements.....	90
Fig. 5.10	Microcontroller execution time of PSO with log barrier	94
Fig. 5.11	Measured RSSIs at different distance	95
Fig. 5.12	Measured RSSI versus best fit RSSI	96
Fig. 5.13	Best fit curves for max, mean and min RSSIs.....	97
Fig. 5.14	Measured RSSIs at different distances, conducted at the Nanyang lake	98
Fig. 5.15	Measured RSSI versus best fit RSSI	98
Fig. 5.16	Best fit curves for max, mean and min RSSIs.....	99
Fig. 6.1	Selection scheme's flow chart.....	109
Fig. 6.2	Eight localization scenarios.....	111
Fig. 6.3	Estimated location from Fig. 6.2	113
Fig. 6.4	Average localization error versus number of anchor nodes	115
Fig. 6.5	PSO with log-barrier versus SBL (7 neighbors, $\alpha = 4$).....	117
Fig. 6.6	PSO with log-barrier versus SBL (10 neighbors, $\alpha = 2$).....	117
Fig. 6.7	PSO with log-barrier versus SBL (10 neighbors, $\alpha = 4$).....	118
Fig. 6.8	PSO with log-barrier versus SBL (10 neighbors, $\alpha = 6$).....	118
Fig. 6.9	Experimental site for the general network	119
Fig. 6.10	Estimated location for the experiment	121
Fig. 6.11	Flow chart of multilateration approach for the experiment.....	122

List of tables

Table 3-1	PSO performance without constraints	47
Table 3-2	PSO performance with constraints	48
Table 3-3	PSO performance with log-barrier constraints	48
Table 3-4	Classical barrier function versus modified barrier function	49
Table 3-5	Localization errors: classical barrier function versus modified barrier function	49
Table 4-1	Execution time versus c_1 and c_2	58
Table 4-2	Location error versus c_1 and c_2	59
Table 4-3	Location error versus c_1 and c_2 (extended investigation)	60
Table 4-4	Average error versus standard deviation	63
Table 4-5	Average error versus μ with $\sigma = 3\text{dB}$	65
Table 4-6	Memetic algorithm (DE and Marquardt - Levenberg)	67
Table 4-7	Memetic algorithm (DE-ML) versus PSO with log-barrier constraint	70
Table 5-1	Latencies involved with a single RSSI measurement	91
Table 5-2	Estimated path loss parameters at the RTP	97
Table 5-3	Estimated path loss parameters at the Nanyang lake	99
Table 5-4	Averaged RSSI (from 10 readings) at various locations	101
Table 5-5	Localization results (experiment)	102
Table 6-1	Average errors versus number of anchor nodes	115
Table 6-2	Average error over a range of RSSI standard deviation	119
Table 6-3	Unknown nodes versus number of neighbor anchor nodes	120
Table 6-4	Experiment: Selection scheme versus multilateration	123

List of Abbreviations

AOA	Angle-Of-Arrival
AODV	Ad-hoc On-demand Distance Vector routing
APS	Application Support Packet
CC	Command Sequence Character (for the XBee Znet 2.5 module)
DE	Differential Evolution strategy
DSSS	Direct Spread Spectrum
DTOA	Difference in Time of Arrival
DV	Distance Vector
GA	Genetic Algorithm
GN	Gauss-Newton method
GPS	Global Positioning System
GT	Guard Time (for the XBee Znet 2.5 module)
HS	Harmony Search
MBF	Modified Barrier Function
MCB	Monte Carlo Localization Boxed
MCL	Monte Carlo Localization
MDS	Multi-Dimensional Scaling
ML	Marquardt-Levenberg method
NLOS	Non-Line-Of-Sight
PAN	Personal Area Network
PMBF	Penalty/Modified Barrier Function
PSO	Particle Swarm Optimization
RF	Radio Frequency

RFID	Radio Frequency Identification
RLT	Reformulation-Linerization Technique
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SBL	Sequenced Based Localization
TOA	Time-Of-Arrival
UART	Universal Asynchronous Receive Transmit
WSN	Wireless Sensor Network

Summary

Wireless sensor network (WSN) has been actively researched in recent years because of its wide range of applications such as natural habitat monitoring, healthcare, home automation, traffic management and military applications. In many cases, the sensed data is not useful without knowing the sensor nodes' locations. Localization algorithms can be categorized as range-free and range-based. In this research, the radio frequency (RF) based approach is used. In conventional RF-based approaches, the path loss parameters in the received signal strength (RSS) – distance model are often determined based on extensive measurements conducted offline. Because the RSS measurements fluctuate with the environment, the use of fixed path loss parameters would reduce the localization accuracy. The purpose of this research is to develop a RF-based localization algorithm where the path loss parameters and the unknown locations are determined simultaneously in real time.

In this thesis, a new algorithm is first developed to solve the problem where an unknown node can communicate with four anchor nodes. The problem is modeled as a constrained nonlinear optimization problem. As conventional Particle Swarm Optimization (PSO) is unable to solve the problem, a new PSO combined with a log-barrier function is developed to tailor to this particular problem. The algorithm has been validated by simulation and experiment. The sequenced based localization (SBL) and the multilateration are used for benchmarking. For a 4dB RSSI measurement error (about 26 % in the measured received signal strength), the average localization error for the proposed method is 19%. It outperforms the multilateration and SBL

approaches that have errors of 32% and 27% respectively. The PSO parameters and the barrier parameter are also tuned to choose the values that give the smallest localization error. Experimental work has also been carried up and the results show that the error for the proposed approach is smaller than the multilateration by 1.22m on average. In the experiment, low-cost microcontrollers and RF modules are used. The time taken to measure the RSSIs and to estimate the unknown location is less than 1 second, which shows that the algorithm is suitable to perform real-time localization.

Finally, the algorithm has also been extended to more general cases where the unknown node can communicate with more than four anchor nodes. A selection scheme has been developed to choose the best estimated result. From our extensive simulation study, it is shown that the algorithm outperforms the SBL by at least 6 m.

Chapter 1

Introduction

1.1 Wireless sensor networks and applications

Wireless sensor network (WSN) is a system consisting of a large number of sensor nodes deployed in an area of interest. In the system, each tiny sensor node is an integration of sensors, a wireless communication system and a processor. There are many different types of sensors used to detect the environment. For instant, temperature, humidity, air quality, vibration sensors, etc. Through the use of advanced networking protocols, the sensor nodes form a community to monitor the environment without human involvement.

WSN has wide ranging of applications such as healthcare, natural habitat monitoring, home automation, traffic management and military applications [1-3]. With sensors embedded throughout a senior's home [4], he or she may not need to move into nursing home because the sensors will keep an eye on him when he becomes frail. Sensors will detect when he has sleepless nights or forgets to take the medication. Web-based computer will then notify the caregivers.

Another WSN's application is habitat monitoring, which is concerned about monitoring the plants and animals in the field [5]. Traditional invasive method may distort results by changing behavioral patterns or distributions. At worst it can

seriously reduce sensitive populations by increasing stress, reducing breeding success. Sensor networks represent a significant advancement over traditional method. For example, the investigation of the life cycle of a juvenile fish has been conducted by attaching a small sensor on it [6]. With the help of many ultrasonic sensors deployed in the ocean, the fish's movement and behavioral is reported to the research centre.

Wireless sensor network can also be embedded to a car park to detect the available lots and their locations [7-8]. The system can be equipped with an algorithm to guide the patrons to the nearest lots. Moreover, the car-park operator may use the information from the sensors for management and planning. For this application, the WSN provides reliable information without the need to install cabling for network and electricity to reach each sensing device.

In many applications, the sensed data might not be useful without knowing the sensor node physical locations. Consider the use of WSN for a fire fighting situation, a fireman without prior knowledge about the building needs to move from room to room to discover the fire and rescue people. For effective command and control, the commander needs to know the fireman's location to manage and instruct the team. Another example is the factory's warehouse. There are many different types of products stored in the same location. As the company always changes products to meet the market requirements, it is necessary to identify and locate different types of products automatically.

1.2 Overview of localization techniques

There are a number of localization solutions for wireless sensor networks [9-12].

Some of the localization systems require complex hardware to be stationed at some fixed locations, such as on the ceilings or walls. Cricket developed by Priyantha et al. [10] is such a system. The beacons with known locations are placed on the ceilings or walls to disseminate their locations periodically. The beacons send the RF signals together with the ultrasonic pulses. Based on the difference in time of arrival (DTOA) of the two signals, the distance between the beacon and the sensor is determined. Cricket is a decentralized beacon network. Thus it is easy to configure and manage. However, there is a lack of explicit coordination and leads to interferences between RF and ultrasonic signals, or between signals from different beacons. Hence, the system parameters must be carefully chosen.

Another example is the Spotlight localization system [9], which is applicable to outdoor environment. In this application, the wireless sensors are randomly deployed from an unmanned aerial vehicle. After the deployment, the sensor nodes self-organize into a network and execute a time-synchronization protocol. An aerial vehicle equipped with a device, called Spotlight, flies over the network and generates light events. The sensor nodes detect the events and timestamps them. These are then reported back to the Spotlight device. The Spotlight computes the location of the sensor nodes using the timestamp sequences and a known event distribution function. Spotlight offers good result but requires an aerial vehicle to generate light event onto the sensor field.

In contrast to the Spotlight system, concentric anchor beacon algorithm [11] is a localization algorithm that does not require complex hardware as it is dependent only on the sensors' connectivity. It is a distributed range-free approach and uses a small

percentage of anchor nodes. Each anchor node emits several beacon signals at different power levels. The beacon carries information including the anchor's position, its power level, estimated maximum distance that the beacon can travel. From the received information, the sensor nodes can determine within which annular ring (whose center is an anchor) they are located. Each node uses the approximated center of intersection of the ring as the position estimate. The concentric anchor beacon algorithm is cost effective, energy efficient, and simple to implement. However, like other range-free approaches, the location result is not very accurate.

Contrasting to the range-free approach is the range-based approach, where a sufficient number of pair-wise distances between unknown and localized nodes are used to compute the unknown node's position. To ensure that there are always enough pair-wise distances, Priyantha et al. [12] introduce a mobile device as "virtual" node to assist in measuring distances between node pairs regardless of how the sensor nodes are deployed. The mobile moves to various positions until these distance constraints form a "globally rigid" structure that guarantees a unique localization. Besides the needs to use a mobile node, another disadvantage is the needs to move sufficiently to have enough coverage to form a 'globally rigid' structure. The requirement of the movement depends on the scenarios. Hence, the localization is not robust.

1.3 Motivation

Accurate localization is essential to provide fine-grained location information. This can be achieved by employing many different types of physical signals, such as infra-red, ultrasonic, radio signals, etc. However, sensors emitting infra-red or ultrasonic

signals often operate in a limited angle. Hence, they are not able to cover a large area. In contrast, radio signal reaches easily to many other transceivers. Many researchers have proposed many different radio frequency (RF) – based localization solutions [12-16]. It is well-known that the received signal strength (RSS) is inversely proportional to the separating distance between the transmitter and the receiver. The separating distance can be calculated from the corresponding RSS. Once the relative distances between sensor nodes are known, the locations of the sensor nodes can be determined. The RSS is a simple and easy to implement approach because most of the COTS (commercial off-the-shelf) wireless communication modules are capable of providing the RSS from the received signal. As was discussed in section 1.2, the mobile-assisted localization approach [12] is RSS-based. The distances between unknown and localized nodes are deduced from the RSS measurements.

Another RSS approach is the power adaptive localization algorithm [13]. It is based on particle filtering for sensor networks assisted by multiple transmit-power information. From the RSS measurements, the distances between the unknown and anchor nodes are estimated. Initially, the unknown node's location is represented by a set of particles. At time t , the RSS between the unknown and anchor nodes are measured. A posterior probabilistic distribution function of the unknown location is constructed given all the measurements. The location of unknown node at time t is estimated based on all the measurement available up to time t and the posterior probability distribution function.

Both the mobile-assisted and power adaptive localization algorithms utilize the RSS information to deduce the corresponding distances, based on a RSS-distance

relationship model. The model parameters are estimated beforehand, hence gives limited accuracy. This is due to the fact that there is high variance of the RSS measurements and the logarithmic relationship between RSS and distance [17].

In this research, the RSS-based localization approach is investigated. The parameters of the model describing the RSS-distance relationship are estimated in real time, together with the locations of the unknown sensor nodes.

To solve this optimization problem, a real time Particle Swarm Optimization (PSO) scheme is used. The PSO method has been used by few researchers for the localization of wireless sensor networks [18-20]. They are all based on RSS measurements. In [19-20], the path loss parameters are determined off-line. The objective function of the PSO is the sum of squares of error between the estimated and the actual distance. In [20], weighted values are included in the objective function to further reduce the estimation error. Since the objective function is not highly nonlinear, conventional PSO is able to solve the problem. In our preliminary research [18], a PSO-based localization scheme is also proposed in which the path loss parameters are determined in real time. The optimization problem is to find the path loss parameters such that after calculating the angles from the unknown node to the four anchor nodes, summation of which is equal to 360° . The problem is not highly nonlinear; hence conventional PSO is sufficient to optimize it. The disadvantage is that the location is only estimated after obtaining the path loss parameters, which might affect the speed of the localization process. Different from these reported approaches, the path loss parameters and unknown coordinates are estimated in real time in this thesis by optimizing a highly nonlinear objective function. As conventional PSO is not able to

solve the problem, a new PSO approach is proposed.

1.4 Objectives

The aim of this research is to develop a novel RSS-based localization approach for a wireless sensor network. As was discussed in section 1.3, the conventional RSS-based approaches often determine the parameters of the RSS-distance model beforehand. This leads to serious error in real application. The objective of this research is to explore a new optimization approach that can estimate the sensor nodes' locations by using RSS with the RSS-distance model's parameters determined in real time. The new approach should have better accuracy as compared to the conventional approaches. The detail objectives are given as follows:

- Based on the RSS-distance relationship model, a constrained non-linear optimization problem will be formulated. Because the RSS measurement is erroneous, there is no direct analytical model to calculate the unknown location. Instead, the unknown location must satisfy constraints imposed by the RSS measurement and the RSS-distance relationship model.
- A new optimization approach will be developed to solve the formulated problem. The new approach that tailors to this particular problem with the logarithmic RSS-distance model will be studied. The localization algorithm should converge faster and provide more accurate results than the conventional approach.
- Simulation studies will be conducted to validate the effectiveness of the

developed localization method. The robustness of the approach will be investigated by varying the anchor and unknown nodes' locations as well as the parameters of the RSS-distance model.

- Experiment will be conducted in real environment using a small-scale WSN setup. The proposed algorithm shall be implemented for real time experimentation.

1.5 Main contributions

The main research contributions in the RF-based localization approach and constrained non-linear optimization are as follows:

1. Localization of WSN

- A new real-time RSS-based localization approach has been developed. Different from most RSS-based approaches, the path loss parameters are estimated together with the unknown location. Consequently, errors due to variations of the path loss parameters in real time are avoided.
- If an unknown node can communicate with more than 4 anchor nodes, a selection scheme is used to yield a result with the least error. The developed algorithm achieves high accuracy as benchmarked with the Sequenced Based Localization (SBL) [21], a recent RSS-based localization method. The average error obtained from the developed algorithm is at least 6m smaller than the SBL. Moreover, there is no restriction on the density and distribution of the anchor nodes.

2. Constrained non-linear optimization

- A new method in solving constrained non-linear optimization problem has been developed. In this new method, PSO together with a log-barrier function has been shown to work well to solve the localization problem. In contrast to other popular techniques such as the conventional PSO or the memetic algorithm (Differential Evolution Strategy and Marquadt-Levenberg) that cannot converge to an optimum, the developed optimizer converges quickly within 20 iterations. Furthermore, the coding of the optimizer is small enough to be implemented into a low cost microcontroller.

1.6 Organization of thesis

The outline of this thesis is as follows:

Chapter 1 introduces the overview of the topics, motivation, objectives and main contribution of the thesis.

Chapter 2 reviews various localization algorithm for WSNs developed in the literature. These techniques will be categorized and briefly discussed in a proper order.

Chapter 3 presents the system model. The fundamental scenario with four anchor nodes is discussed. After the problem is formulated, a new PSO with log-barrier approach is proposed and presented. The proposed approach tailors well to this particular problem, which belongs to the class of nonlinear constrained optimization.

It overcomes the scaling problem due to the logarithmic dependence between the variables.

Chapter 4 presents various simulation studies to validate the algorithm presented in chapter 3. Many different scenarios with different RSS-distance model's parameters are assumed for the simulation study. The tuning of all the PSO's parameters, such as the acceleration constants, population size, inertial weight, etc, has also been studied and presented. As the barrier parameter in the proposed method has an effect on the system performance, its impact is studied and an appropriate barrier parameter will be selected.

Chapter 5 discusses the hardware implementation of the sensor nodes that are used in the experiment. The process of implementing the algorithm is also explained. Experimental result of the implemented system are presented and discussed.

Chapter 6 extends the proposed algorithm to the general case with arbitrary number of anchor nodes. With the algorithm presented in chapter 3, there will be many estimated locations resulting from different sets of four anchor nodes. A selection scheme is presented to select a location that yields the best estimation. Simulation study has been conducted to verify the effectiveness of the algorithm with the number of anchor nodes varies from 5 to 10. The average localization errors are presented and compared with another RSS-based method, i.e. the sequenced based localization method (SBL) [21]. The results show that the proposed approach achieves better accuracy as compared to the SBL.

Chapter 7 concludes the work. Possible plans and future direction for further research

are also included.

Chapter 2

Literature review of localization techniques for wireless sensor networks

2.1 Introduction

Wireless sensor network (WSN) localization has been an active area of research in recent years [22-25]. Many solutions with different technologies have been developed. Fig. 2.1 shows the classification of localization techniques.

Fig. 2.1 Classification of localization techniques

As shown in Fig. 2.1, the localization of a WSN can be divided into range-free and range-based algorithms. In the range-free approach, there is no additional hardware used to measure the ranging information. Instead, the only information used is the

connectivity between sensor nodes. Different algorithms have been developed which can be classified into a few groups, namely: averaging, bounding box, DV-hop, etc. In the range-based approach, different techniques are utilized to provide the distance or angle information between sensor nodes. The more popular methods are Time-Of-Arrival (TOA), Angle-Of-Arrival (AOA) and Received Signal Strength (RSS). Besides the range-free and range-based approaches, there are other well-known methods. In this chapter, three popular approaches, namely the Kalman filter, Monte-Carlo localization and Multi Dimensional Scaling (MDS) are also presented.

2.2 Range-free localization

Range-free localization in a wireless sensor network is a method to estimate the sensors' position by using only communication and connectivity, without actual distance measurement.

LANDMARC [28] is a well-known range-free localization system. The system consists of RFID reference tags, tracked tags and readers. The tracked tag is the one whose location is being estimated. Each RFID reader compares the power level for the tracked tag with the power levels of the reference tags. The relative position between a tracked tag with the power levels of the reference tags. The relative position between a tracked and a reference tag is the sum of square of the difference in power level being read by every RF reader. The reference tag with smaller relative position is granted higher weighting value. These weighting values together with known location of reference tag are used to calculate the final location of the tracked tag.

In LANDMARC, it assumes that the tracked and reference tags can communicate with all the reference tags. In real application, this is difficult to implement. Moreover, the

unknown location is obtained by averaging the tracking tags' locations with a weighting factor, a lower deployment of reference tags would result in higher error range [27]. These shortcomings can be overcome by using multilateration in calculating the tracking tag's position.

Another range-free localization approach is to use geometry in calculating the location. If an unknown node receives a message from an anchor node; the unknown node is inside a region that the anchor node transceiver can cover. By intersecting these radio coverage regions, a smaller region containing the true location of the unknown node is obtained. The communication region is typically a circle. However, the radio coverage region is simplified as a rectangle in the bounding box approach [29]. This makes the calculation of the intersection region less complicated. Fig. 2.2 shows the principle of the bounding box algorithm.

Fig. 2.2 Sketch map of the bounding box algorithm

A polygon-based localization algorithm is proposed in [30]. The algorithm is similar to the bounding box approach, except that a convex polygon is used to represent an unknown node's current estimate of its probable locations. Fig. 2.3 shows the principle of the polygon-based algorithm. If an unknown node receives a message

from another unknown node (already localized), the new unknown node will be inside a polygon obtained by dilating the original polygon of the localized node by the communication distance. The intersection of the two polygons is the region that the unknown node resides. The disadvantage of this approach is the need to store the information of the polygons. Even though calculating the rectangle/polygon intersection region is simpler than calculating the circle intersection, the process still uses much processor resource.

Fig. 2.3 Sketch map of the polygon-based algorithm

Another range-free approach is the DV-hop algorithm (DV stands for Distance Vector) [31-32]. In this method, it first employs a distance vector exchange so that all the nodes in the network get the distances in terms of numbers of hops to the landmark. Once a landmark gets the distance to other landmark, it estimates the average size of a hop. This is then deployed as a correction to the entire network. When receiving the correction, an arbitrary node may then have the estimated

distances to the landmarks in meters which can then be used to perform triangulation.

Fig. 2.4 shows the sketch map of the DV-hop algorithm.

Fig. 2.4 Sketch map of the DV-hop algorithm

From Fig. 2.4, it may be observed that the DV-hop algorithm is prone to error because the real distance of hop 1, 2, 3 or 4 is larger than $SD/4$. To improve the accuracy, the expected hop progress algorithm [31] is proposed. In this method, the distance R which is the projection of distance 1 onto line SD is estimated instead of calculating the hop distance 1. In [31], an average error in distance estimation of 35% with a node connectivity of 40 is achieved. The network is very densely deployed. It is because in DV-hop approach, the more sensors connected, the more hop-distance information is available resulting in higher localization accuracy.

In conclusion, range-free approach only achieves a low level of accuracy in most situations [33]. For a system with low anchor density, each unlocalized node gets less anchor constraints and the estimated location error becomes large [34-35]. However, too many close anchor nodes in a small region cannot increase the localization accuracy. Instead it causes unnecessary waste of memory and computation overhead

[34-35]. In range-free approach, location information from other nodes is needed. Hence, it spends significant communication cost in forwarding the location information.

2.3 Range-based localization

Range-based localization usually consists of two steps: 1. distance (or ranging) estimation and 2. location estimation. The distance is estimated by using RSS, AOA, TOA or TDOA. The estimated distances are then used to locate the unknown position by using triangulation, maximum likelihood or fingerprinting. In the following sections, we highlight several of these approaches.

2.3.1 Time-Of-Arrival (TOA)

Time-Of-Arrival (TOA) is the travel time of a signal between a transmitter and a receiver. Ultrasound signal is often used [37-40] for this approach as the speed of sound in the air at 331.3 m/s is slow. If a beacon transmits both the RF and ultrasound signals simultaneously, the ultrasound signal will arrive at the listener much later than the RF signal. Based on the time difference of arrival times, the listener can derive the distance from the beacon.

TOA is often used for precise indoor localization. LotTrack [37] is one such method and uses ultrasonic sensors to track the production lots in a wafer fabrication facility. In this application, the ultrasound emitters that are mounted on the ceiling send signals periodically. These will be received by the active RFID transponders that are equipped with the ultrasonic sensors. As the facility's layout contains no wide rooms but only interconnected corridors, a corridor ID and a distance value give sufficient

information to determine any place in the factory.

In a real application, the sensor network is often not synchronized. Therefore, there is an unknown clock offset between each pair of nodes, which is time varying. Yu et al. in [41] proposed a clock frequency offset estimation algorithm to compensate for the frequency difference between each pair of nodes. The error is well kept below 30 ppb (parts per billion).

Other factors such as the receiver noise, channel fading, and the bias due to non-line-of-sight (NLOS) propagation also contribute to the TOA measurement error. TOA measurement is often used to obtain coarse sensor nodes' position estimates [42]. In [43], it is combined with the received signal strength (RSS) measurements to provide fine sensor node localization.

2.3.2 Angle-Of-Arrival (AOA)

Angle-Of-Arrival (AOA) [44-46] provides information about the angle between transceivers. By measuring the Time Difference of Arrival (TDOA) at individual elements of an antenna array, the AOA can be calculated.

In [44], it is assumed only a fraction of the nodes (called landmarks) in the network has positioning capability and all the nodes have the AOA capability. A sensor node immediately adjacent to a landmark obtains the angle of arrival of receiving signals from the anchor nodes. After the sensor node is localized, its location in turn is transmitted to another neighboring node. However, the estimation error will propagate and accumulate throughout the network.

Boushaba et al. [47] proposed HA-A2L, an AOA-based algorithm, in which the sensor node decides whether to transmit the computed position. Moreover, a mechanism is developed to allow anchor nodes that are far from an unknown node to involve in the position estimation. Fig. 2.5 shows the map of the HA-A2L algorithm. Unknown node A obtains distance and angle information from the anchor nodes C and D. Hence the node A is inside the shaded regions centered at C and D respectively. Although A cannot communicate with the anchor node L, there is an unknown node B which can communicate with both A and L. By using information from B, a set of constraints is derived. This results in another shaded region centered at L containing the location of A. With information from other unknown nodes, an unknown node that is far from the anchor nodes can still be localized.

Fig. 2.5 Sketch map of HA-A2L

To limit the propagation error, a node that computes its position is limited to use information from nodes that are less than 10 hops. With a landmark density of 20%, the localization error is about 7% (compared to the communication distance). To achieve this performance, it requires a dense network of 250 unknown nodes.

2.3.3 Received Signal Strength (RSS)

Different from TOA or AOA where ultrasonic sensor or antenna array is required, the RSS requires no additional hardware. It is well-known that the signal strength is reduced inversely proportional to the separating distance between two transceivers [48]. In general, knowing the signal strength, the distance can be deduced.

There are many models describing the power-distance relationship. Some path loss models for outdoor case are Okumura model, Hata model, Cost-231-Walfisch-Ikegami model [48] etc. However, these models are not suitable for WSN as the communicating range is much shorter than the mobile communication. Instead, the following path loss model is widely used in WSN

$$P_L(d) = P_L(d_0) + 10\alpha \log(d/d_0) + X_\sigma \quad (2-1)$$

where $P_L(d_0)$ is the average path loss measured at a separating distance of d_0 , α is the path loss exponent, X_σ denotes a zero-mean Gaussian random variable of standard deviation σ , which reflects the variation that naturally occurs when a path loss model of this type is used.

For the indoor channel, it differs from the traditional outdoor mobile radio channel in two aspects: the distance covered is much shorter, and the variability of the environment is much greater for a much smaller range of transmitter and receiver separation distances. Propagation into and inside buildings has, to some extent, a more complex multipath structure than an outdoor propagation environment. Indoor radio propagation is affected by the layout of a building especially if there are various building materials. To take into account the attenuation due to walls and floors, two

addition terms are added to (2-1) as follows:

$$PL(d) = PL(d_0) + 10\alpha \log(d/d_0) + \sum_{q=1}^Q FAF(q) + \sum_{p=1}^P WAF(p) \quad (2-2)$$

where $FAF(q)$ and $WAF(p)$ are the floor and the wall attenuation factors, respectively.

This indoor propagation model is employed in [36]. Various measurements are taken to calculate the average WAF and FAF to compensate for the effect of walls and floors. In [14], another model described in the following equation is proposed:

$$P(d)[dBm] = P(d_0)[dBm] - 10n \log\left(\frac{d}{d_0}\right) - \begin{cases} nW \times WAF & nW < C \\ C \times WAF & nW \geq C \end{cases} \quad (2-3)$$

where C is the maximum number of obstructions (walls) up to which the attenuation factor makes a difference; nW is the number of obstructions (walls) between the transmitter and the receiver.

The model (2-1) is widely used to describe the power-distance relationship in WSNs [49-51]. The model parameters are usually estimated by taking various RSSI measurements at known distances, and followed by the model best fit the experimental data. However, such approach gave poor location accuracy due to the high variance of the RSS measurements [17].

M. Barralet et al. [17] studied the effects of polarization on the accuracy of an indoor location tracking system. Small simple Omni-directional antennae produce linearly polarized radiation. The electric and magnetic fields components of electrical magnetic (EM) radiation are perpendicular to the direction of propagation. Misaligned

antenna polarization between two transceivers also affects the received power. Another element is included in the path loss model to take into account the loss due to misaligned antenna.

The values of the path loss parameters are often determined in advanced through extensive measurements. From the known distances and the measured RSSIs, the path loss parameters are estimated which is then used for the localization process. However, the path loss parameters are specific to the area where the unknown node is located. Hence, the use of fixed and pre-determined path loss parameters could result in significant error.

There are localization algorithms developed recently to avoid using the pre-determined path loss parameters. In [52], a two-stage algorithm is proposed to adjust the path loss parameters. In the first stage, the pre-determined parameters are used to obtain a coarse location of the unknown node. The distance between the unknown and an anchor node is calculated again, from which a set of path loss parameters can be deduced. The new values will be used in the next iteration. The process stops once the path loss parameters converge.

In [15], Ahn and Yu use fixed reference nodes and tags. The fixed nodes and tags are placed at known locations. Hence the path loss parameters at the respective location can be calculated from the RSSI measurements in real time. In [53], the values of the path loss parameters are estimated in real time by using the RSSI between two anchor nodes. A selection scheme is proposed to select the anchor node that is close to the unknown node, so that the estimated values reflect the values at the area where the

unknown node is located more accurately.

2.3.4 Triangulation, trilateration and maximum likelihood

After obtaining the ranging information by using AOA, TOA or RSS, the trilateration or multilateration is often used to calculate the unknown sensor node's coordinates. Trilateration, or multilateration [15,17,54-55] is a method of determining the relative positions of objects using the geometry of triangles in a similar fashion as triangulation. Assuming that there are three or more anchor nodes (with known locations), and also the distances between the unknown and anchor nodes are known, the location of the unknown node is the intersection of various circles whose centers are the anchor nodes' locations. [56] shows that localization based on triangulation faces many problems that degrade the localization accuracy. The problems are uncertainty, non-consistency, ambiguity, and error propagation. These problems are inevitable because of the noisy measurements.

Similar to trilateration, triangulation uses angle measurements to calculate the subject's location. Fig. 2.6 shows the map of the triangulation method [44]. In triangulation, the vertices A , B , and C are localized. If the angles that the unknown node O sees the vertices (i.e. $\widehat{AOB}$, $\widehat{BOC}$, $\widehat{COA}$) are known, triangulation can determine the location of the unknown node O . Triangulation has been widely used in many localization methods [50,57]. Triangulation problem can also be reduced to trilateration by some simple transformations [44].

Fig. 2.6 Sketch map of triangulation

Location estimation can be done by using a probability based method, namely maximum likelihood estimation. The maximum likelihood method is more precise than the trilateration method. It makes use of an underlying probability model to build a likelihood function. The method picks the value of the unknown variable where the likelihood function is the largest. The maximum likelihood method has been widely used in localization of WSNs [58,59]. In [60], a Fault Tolerant Maximum Likelihood (FTML) method is proposed. FTML is modified from the classical maximum likelihood method to accommodate erroneous observation coming from faulty sensor nodes. In FTML, the fault probability is incorporated when calculating the likelihood function.

2.4 Other localization techniques

There are several other localization approaches which also provide good estimation. Kalman filter and Monte-Carlo localization are often used for WSN whose sensor nodes are mobile. Multidimensional scaling (MDS) is a set of statistical techniques that find the similarities and dissimilarities between the observed data and the location

coordinates.

2.4.1 Kalman filter

Kalman filter has proved to be efficient in localization or navigation system [61-63]. It is a recursive filter that estimates the state of a dynamic system from a series of noisy measurements. Only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. In contrast to batch estimation techniques, no history of observations and/or estimates is required.

In its original form, the Kalman filter does not correctly handle non-linear or non-Gaussian motion [64]. Some modified Kalman filters which are used with non-linear system are Extended Kalman filter and Unscented Kalman filter. In Extended Kalman filter, the non-linear predict and update functions are linearized by using Taylor expansion. However, when these functions are highly non-linear, Extended Kalman filter gives poor performance [61]. Different from the Extended Kalman filter, Unscented Kalman filter [61] uses a deterministic sampling technique to pick a minimal set of sample points around the mean. These points are then propagated through the non-linear functions, from which the mean and covariance of the estimate are then recovered.

In [61], a moving node in indoor environment equipped with an ultrasound sensor to measure the distance between movement steps. Due to multi-path, the Unscented Kalman filter is employed to 'pre-filter' such a bouncing problem and to localize the mobile nodes more accurately and reliably.

2.4.2 Monte Carlo localization

Although modified Kalman filter can be used to cope with non-Gaussian system, newer methods have used richer scheme to represent uncertainty, moving away from the restricted Gaussian density assumption inherent in Kalman filter. In Monte Carlo localization method, instead of describing the probability function itself, a set of sample is randomly drawn from it. This density representation is updated over time.

In [65], Monte Carlo method is successful in building a low-cost system for personal tracking in indoor environments. Three pieces of information which are the local mobility model, indoor map information and proximity information from static localized nodes are combined.

Sequential Monte Carlo Localization method [35] is a popular MCL method where observations arrived sequentially in time. It updates the posterior distribution as data become available. Computation becomes easier because there is no need to store all the observed data [66].

The Monte Carlo Localization Boxed (MCB) scheme based on MCL is proposed in [67]. It decreases the probability of choosing wrong samples and increases the efficiency on sample selection by using the anchor information not only in the filtering phase, but also in the prediction phase. However, it cannot reduce the localization error if MCB and MCL have the same number of valid samples.

The sampling approach in MCL successfully draws the possible locations of normal nodes. However, there are two drawbacks in the above methods. First, in low anchor-density, each normal node gets less anchor constraints, and the estimated location

error becomes large. Second, in high anchor-density, each normal node gets more location constraints from anchors, and the possible located region of the normal node becomes small. It is difficult to find many valid samples in a small region. Therefore, too many close samples in a small region cannot increase the localization accuracy, but instead cause unnecessary waste of memory and computation overhead.

2.4.3 Multi-Dimensional Scaling (MDS)

Multi-dimensional scaling [68-69] is quite different from the probabilistic approaches (Kalman filter, MCL). It is a data analysis technique that transforms the pair-wise distance information into the relative coordinates of nodes.

Let p_{ij} refer to the proximity measure between objects i and j . The Euclidean distance between two points $\mathbf{X}_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$ and $\mathbf{X}_j = \{x_{j1}, x_{j2}, \dots, x_{jm}\}$ in an m -dimensional space is d_{ij} . When the geometrical model of the objects fits the proximity data perfectly, the Euclidean distances are related to the proximities by a transformation $d_{ij} = f(p_{ij})$. The distances $D = \{d_{ij}\}$ are determined so that they are as close to the proximities $P = \{p_{ij}\}$ as possible. There are a variety of ways to define “close.” A common one is a least-squares definition, which is used by classical metric MDS.

Yi Shang et al. [70] proposed a localization algorithm based on centralized MDS (MDS-MAP (C)). The parameter C refers to centralized as the connectivity information of the network is sent to a central location where the computation is carried out. In this method, the proximities P are the shortest path distances between all pair nodes in the region of consideration.

2.5 Conclusion

In this chapter, an overview of various localization methods is presented. The range-free approach has the advantage of simple design and ease of implementation. However, the accuracy is poor as compared to the range-based approach. For the range-based method, the RSS is the easiest to be implemented in real application. Although Kalman filter and Monte Carlo localization methods make use of continuous measurements to improve the localization accuracy, they require the sensor nodes to be mobile. MDS does not appeal to our research as RSS with an appropriate power-distance relationship has been chosen. The challenge is to model the power-distance relationship close to the real environment. To provide good location estimation for a WSN, a localization algorithm based on RSS will be investigated with the power-distance relationship model in real time. The problem formulation and its solution will be presented in chapter 3.

Chapter 3

Real-time localization using Particle Swarm Optimization with log-barrier constraint

3.1 Introduction

In chapter 2, various localization methods have been reviewed. As one of the objectives of the thesis is to develop a low-cost localization solution, the RSS-based approach is chosen. In this chapter, we discuss the use of multiple RSSI information to provide location estimation at significantly higher accuracy. To achieve this result, an optimization problem is first formulated. Moreover, we also present a new optimizer consisting of PSO and log-barrier function that performs well for this particular problem with low level of complexity. The developed localization algorithm is simple and effective for real-time localization.

In this chapter, we first present and discuss a localization scenario to illustrate the real application. Based on the RSS path loss model, the feasibility of RSS-based approach is discussed, leading to a new RSS-based localization technique. By using the maximum likelihood theory, a constraint nonlinear optimization problem is formulated. Since simple PSO cannot solve the problem; a new PSO is developed by combining PSO and log-barrier function. The effectiveness of the combination is also

discussed.

3.2 System model

Fig. 3.1 shows an example of a WSN that has many sensors deployed inside an area of $300\text{m} \times 300\text{m}$. The network consists of N anchor nodes with known locations, and M sensor nodes without knowledge about their locations. In this study, the anchor nodes refer to sensor nodes that have knowledge of their locations (e.g. equipped with GPS) or have been localized. Typically, every wireless sensor node has a RF transceiver that is capable of measuring signal strengths from the received packets. For the sensor network used in this study, the communication range is 100m. In Fig. 3.1, the filled circles represent anchor nodes and the unfilled circles represent unknown nodes. In the figure, a line joining an unknown node and an anchor node indicates that the two nodes are within the communication range. For example, the unknown node 'X' in Fig. 3.1 is able to communicate with four neighbor anchor nodes A , B , C , and D . In the figure, there are different line styles representing the communication to differentiate between different unknown nodes.

Fig. 3.1 Wireless sensor network

3.2.1 Communication path loss modeling

For each sensor node, it transmits a RF signal to the surrounding environment. Signal propagating from the base station to a mobile unit is usually subject to a condition such that the propagation between them is largely through scattering, either by reflection or diffraction from buildings and terrain or objects within buildings, because of obstruction of the line-of-sight (LOS) path. Various propagating models have been reported [37,48-51]. In this study, the following path loss model that is widely used for WSNs [49-51] will be used

$$P_L(d) = P_L(d_0) + 10 \cdot \alpha \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (3-1)$$

where $P_L(d_0)$ is the average large-scale path loss at a reference distance of d_0 in dB. α

is the path loss exponent. X_σ denotes a zero-mean Gaussian random variable of standard deviation σ , which reflects the variation that naturally occurs when this type of path loss model is used. In (3-1), it is noted that $P_L(d) = \log\left(\frac{P_t}{P_r}\right)$, where P_t , P_r are the transmitted and received powers in Watts. Hence, the terms in (3-1) are dimensionless.

Given a transmitted power P_t from the sensor node, the received power $P_r(d)$ (in dB) at a distance d is

$$P_r(d) = P_t - P_L(d) \quad (3-2)$$

Substituting (3-1) into (3-2) yields

$$P_r(d) = P_0 - 10 \cdot \alpha \cdot \log\left(\frac{d}{d_0}\right) - X_\sigma \quad (3-3)$$

where $P_0 = P_t - \overline{P_L}(d_0)$ is the average received power at a reference distance d_0 . In practice, most of the transceivers used in the WSN are capable of measuring the received power $P_r(d)$, which is termed as the Received Signal Strength Indicator (RSSI).

3.3 Problem formulation

Let's assume that an unknown sensor has n neighbor anchor nodes in its communication range. Thus, there are n RSSIs, P_r , $i = 1, \dots, n$. Observing from (3-3), there are 3 parameters (i.e. P_0 , α and σ) that define the path loss model of the communication channel between two sensor nodes. The corresponding path loss

parameters of n pairs of unknown and anchor nodes are P_{0i} , α_i and σ_i , with $i=1,\dots,n$.

As these parameters are the characteristics of the same communication channel, we have

$$\begin{aligned} P_{01} &= P_{02} \dots = P_{0p} = P_0 \\ \alpha_1 &= \alpha_2 \dots = \alpha_p = \alpha \\ \sigma_1 &= \sigma_2 \dots = \sigma_p = \sigma \end{aligned} \quad (3-4)$$

Let x_i, y_i be the Cartesian coordinates of the neighbor anchor node i . Moreover, let x and y be the Cartesian coordinates of the unknown node. The distance d_i between the unknown node and its neighbor anchor node i is

$$d_i = \sqrt{(x-x_i)^2 + (y-y_i)^2} \quad (3-5)$$

Substitute (3-5) into (3-3) yields the RSSI between the unknown node and the anchor node i as

$$P_{r_i} = P_0 - 10 \cdot \alpha \cdot \log \left(\frac{\sqrt{(x-x_i)^2 + (y-y_i)^2}}{d_0} \right) - X_{\sigma_i} \quad (3-6)$$

where X_{σ_i} is a Gaussian distributed random variable with zero mean, i.e. $X_{\sigma_i} \sim N(0, \sigma^2)$. From (3-6), P_{r_i} at a specific location is also Gaussian distributed. Thus, we have

$$P_{r_i} \sim N \left(P_0 - 10 \cdot \alpha \cdot \log \left(\frac{\sqrt{(x-x_i)^2 + (y-y_i)^2}}{d_0} \right), \sigma^2 \right) \quad (3-7)$$

The probability of a given measured P_{r_i} can be obtained as:

$$P(P_{r_i}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(P_0 - 10 \cdot \alpha \cdot \log\left(\frac{\sqrt{(x-x_i)^2 + (y-y_i)^2}}{d_0}\right) - P_{r_i})^2}{2\sigma^2}\right) \quad (3-8)$$

From the estimation theory, the likelihood function of P_0, α, x, y based on the measured P_{r_i} is equal to the probability of P_{r_i} . Let

$\tilde{X}_{\sigma_i} = P_0 - 10 \cdot \alpha \cdot \log\left(\frac{\sqrt{(x-x_i)^2 + (y-y_i)^2}}{d_0}\right) - P_{r_i}$ and substitute into (3-8) yields

$$L(P_0, \alpha, x, y | P_{r_i}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\tilde{X}_{\sigma_i}^2}{2\sigma^2}\right) \quad (3-9)$$

With a network of n neighbor anchor nodes and n independent RSSI measurements, the overall likelihood function can be generalized as follows:

$$\begin{aligned} L(P_0, \alpha, x, y | \tilde{P}_{r_i}, i = 1, \dots, n) &= \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\tilde{X}_{\sigma_i}^2}{2\sigma^2}\right) \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \exp\left(-\frac{\sum_{i=1}^n \tilde{X}_{\sigma_i}^2}{2\sigma^2}\right) \end{aligned} \quad (3-10)$$

From (3-10), the likelihood function L is maximized if $\sum_{i=1}^n \tilde{X}_{\sigma_i}^2$ is minimized. It is

noticed that there are n RSSI equations with four unknown variables P_0, α, x and y .

Hence, it is sufficient to solve the problem with n equals to 4. In the sequel, we will

first solve the problem with n equals to 4. For the case that the unknown node has

more than 4 neighbor nodes, modification of the algorithm is required and will be discussed in chapter 6.

From (3-7) and (3-10), the problem can be formulated in the following mathematical form with $n = 4$:

$$\begin{aligned} & \min_{P_0, \alpha, x, y} f(P_0, \alpha, x, y \mid P_{ri}, i = 1, \dots, 4) \\ \text{subject to } & \begin{cases} \alpha_{\min} \leq \alpha \leq \alpha_{\max} \\ P_{0\min} \leq P_0 \leq P_{0\max} \\ x_{\min} \leq x \leq x_{\max} \\ y_{\min} \leq y \leq y_{\max} \end{cases} \end{aligned} \quad (3-11)$$

$$\text{where } f = \sum_{i=1}^4 \left(P_0 - 10 \cdot \alpha \cdot \log \left(\frac{\sqrt{(x-x_i)^2 + (y-y_i)^2}}{d_0} \right) - P_{ri} \right)^2 \quad (3-12)$$

is the nonlinear objective function to be minimized.

3.4 Constrained nonlinear optimization using log-barrier method

3.4.1 Popular nonlinear optimization approaches

The problem (3-11) - (3-12) belongs to the class of constrained optimization which happens in many applications [71-72]. This class of problem is often transformed into the unconstrained type that is easier to solve. There are several optimization methods such as adding penalty, barrier, Lagrangian functions or using a sequence of unconstrained sub-problems [73-74].

In [73], Tan et al. replace a nonlinear optimization problem by a linear programming problem. The nonlinear objective function is linearized by using Taylor expansion.

Beside Taylor expansion, there is other linearization technique, such as the Reformulation – Linearization Technique (RLT) [75-76]. The essence of the RLT is to replace each polynomial term with one new variable and add a set of new linear constraints based on the bounds for partition variables (variables in a nonlinear term) [75]. For example, the optimized function in [76] involves the following nonlinear term

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

To solve the optimization problem, two new variables are introduced

$$Z_{ij} = \sqrt{W_{ij}}$$

$$W_{ij} = (x_i - x_j)^2 + (y_i - y_j)^2$$

A set of linear constraints is formulated for the two new variables. The problem then becomes linear and can be solved by linear programming method. The accuracy of the result is dependent on the chosen set of linear constraints. The more constraints, the more accurate will be the result. In [76], branch-and-bound technique is employed. The result obtained from RLT is treated as a lower bound (LB). Using that result as a starting point, a local search algorithm is used to find a feasible solution, treated as an upper bound (UB). The process continues until $LB \geq (1 - \varepsilon)UB$.

The RLT is often applied to polynomial nonlinear function. However, for nonpolynomial nonlinear function, such as logarithmic function, other technique is required. In [75], the logarithmic function is linearized as follows

$$\ln(1+x) \approx x \text{ when } 0 < x \ll 1$$

For the problem (3-12), there are logarithmic functions. However, these functions often do not satisfy the condition, i.e.:

$$\frac{\sqrt{(x-x_i)^2+(y-y_i)^2}}{d_0} - 1 \geq 1$$

Hence the Reformulation – Linearization Technique is not suitable.

There are other methods to solve constrained nonlinear optimization problem, such as penalty, barrier or augmented Lagrangian methods. Penalty method is often used for problems with equality constraints. On the other hand, barrier method is often used for problems with boundary type of constraints [77]. Hence, the barrier method is considered in this thesis.

The barrier function is formulated by combining the objective function and constraint in such a way as to introduce a “barrier” – an infinite penalty along the constraint boundary. In [78], the classical logarithmic barrier function is included in the objective function making it an unconstrained optimization problem. It is then solved by using Broyden-Fletcher-Goldfarb-Shanno method. In the barrier function, there is a barrier parameter that affects the optimization process. The larger the barrier parameter, the more severe it is penalized if the constraints are violated. Theoretically, the barrier parameter must gradually reduce to zero, so that the optimum of the objective function with barrier method is the same as that of the original objective function. However, as the parameter converges to zero, the unconstrained problem becomes increasingly ill-conditioned [79]. To alleviate the ill-condition difficulty,

modified barrier function (MBF) method for inequality constrained problem is developed. A variation of the MBF is the penalty/modified barrier function (PMBF) method [79]. In the PMBF, the objective function with barrier functions has the following form:

$$g(x, \mu, \lambda) = f(x) - \mu \sum_{i=1}^n \lambda_i \Phi(c_i)$$

where Φ denotes the barrier function, $c_i \geq 0$ with $i=1, \dots, n$ are the constraints. The vector of λ_i is the Lagrange multiplier. Because if x is an optimum for the original constrained problem, then there exists a vector of λ_i such that (x, λ_i) is a stationary point for the Lagrange function (stationary points are those points where the partial derivatives of g are zero). The Lagrange multiplier is introduced to provide a necessary condition for optimality of the constrained problem. The values of μ and λ are not constants, and may change with the iterations. By using the Lagrange multipliers, it is not necessary for the barrier parameter to converge to zero to obtain an exact solution. Polyak [80] has shown that there exists a threshold value $\bar{\mu} > 0$ such that for any fixed value of barrier parameter $\mu < \bar{\mu}$, the algorithm converges to at least a linear rate to a solution of the original problem by updating only Lagrange multipliers λ . Moreover, in the classical barrier function

$$\Phi(c_i) = \ln(c_i)$$

the solution cannot reside on the boundary. In the PMBF method, there is another parameter s_i used in the enlargement of the feasible region for the lower and upper bounds. The logarithmic barrier function will have the following form:

$$\Phi(c_i) = \ln \left(s_i + \frac{c_i}{\mu} \right)$$

Hence, a solution which resides on the boundary can be obtained.

3.4.2 Optimization with log-barrier approach

In this thesis, we propose an optimization approach using a modified log-barrier method to solve the constrained nonlinear optimization problem (3-11) – (3-12). The log-barrier approach is modified to enlarge the feasible region. Consequently if the solution happens to be on the boundary, the algorithm still performs well.

In this problem, the boundary constraints are:

$$\begin{aligned} \alpha_{min} &\leq \alpha \leq \alpha_{max} \\ P_{0min} &\leq P_0 \leq P_{0max} \\ x_{min} &\leq x \leq x_{max} \\ y_{min} &\leq y \leq y_{max} \end{aligned} \quad (3-13)$$

Equivalently, we have the following constraints

$$c_i \geq 0, \quad i = 1, \dots, 8 \quad (3-14)$$

where

$$\begin{aligned} c_1 &= \alpha - \alpha_{min} & , & & c_2 &= \alpha_{max} - \alpha \\ c_3 &= P_0 - P_{0min} & , & & c_4 &= P_{0max} - P_0 \\ c_5 &= x - x_{min} & , & & c_6 &= x_{max} - x \\ c_7 &= y - y_{min} & , & & c_8 &= y_{max} - y \end{aligned} \quad (3-15)$$

There are two types of barrier function, namely *the inverse barrier function* and *the logarithmic barrier function*. These two functions are defined as follows:

$$\text{Logarithmic barrier function: } -\mu \sum_i \ln c_i \quad (3-16)$$

$$\text{Inverse barrier function: } -\mu \sum_i \frac{1}{c_i} \quad (3-17)$$

From (3-16) – (3-17), it is observed that the barrier function approaches infinity when the constraints in (3-14) are violated. Hence, if a potential solution approaches the boundaries, i.e. a constraint is violated; the barrier function becomes infinity. By adding the barrier function to the original objective function, the objective function will also become infinity. Consequently, such solution will not be favored and be eliminated in a few iterations. This will ensure that the final solution will not violate the boundary constraints.

From (3-16), it is observed that when the constraints approach zeros (where it is violated), the value of the barrier function approaches infinity. The logarithmic barrier function is preferred in this case, as (P_0, α) and (x, y) are logarithmically dependent. Thus, a little fluctuation in P_0 or α would lead to a large change in (x, y) due to scaling. Using logarithmic barrier function will overcome the scaling problem. Substitute the constraints (3-15) into (3-16), the objective function (3-12) can be modified as follows:

$$\begin{aligned} g(P_0, \alpha, x, y, \mu) = & f(P_0, \alpha, x, y) - \mu \left(\ln(P_0 - P_{0_{min}}) + \ln(P_{0_{max}} - P_0) + \right. \\ & \ln(\alpha - \alpha_{min}) + \ln(\alpha_{max} - \alpha) + \ln\left(\frac{x}{d_0} - \frac{x_{min}}{d_0}\right) + \ln\left(\frac{x_{max}}{d_0} - \frac{x}{d_0}\right) + \ln\left(\frac{y}{d_0} - \frac{y_{min}}{d_0}\right) + \\ & \left. \ln\left(\frac{y_{max}}{d_0} - \frac{y}{d_0}\right) \right) \end{aligned} \quad (3-18)$$

In (3-18), g, P_0, α are dimensionless. Moreover, d_0 is a fixed reference distance (i.e. 1m). The problem now becomes an unconstrained optimization problem as follows:

$$\min_{P_0, \alpha, x, y} g(P_0, \alpha, x, y, \mu) \quad (3-19)$$

where $g(\cdot)$ is given in (3-18).

In the classical barrier function (3-16), it is noticed that the optimum result cannot be on the boundary. In this case, we use the method in the penalty/modified barrier function (PMBF) discussed in 3.4.1 to enlarge the feasible region of the lower and upper bounds. The barrier function is defined as follows:

$$\Phi(c_i(x)) = \begin{cases} \ln\left(s_i + \frac{c_i(x)}{\mu}\right), & \text{for } i = 1, 2, \dots, n \text{ if } c_i(x) \geq -\beta s_i \mu \\ \varphi(c_i(x)), & \text{for } i = 1, 2, \dots, n \text{ if } c_i(x) < -\beta s_i \mu \end{cases} \quad (3-20)$$

$$\text{where } \varphi(c_i(x)) = \frac{1}{2} q_i^1 (c_i(x))^2 + q_i^2 (c_i(x)) + q_i^3 \quad (3-21)$$

and μ is the barrier parameter, n is the number of constraints.

In (3-20), the value of β defines how close to the singularities the logarithmic terms are extrapolated. The coefficients q_i^1 , q_i^2 and q_i^3 of the quadratic function in (3-21) are chosen so that their values, and their first and second derivatives match those of the logarithmic term in (3-20) at the point $c_i(x) = -\beta s_i \mu$. Hence, we have:

$$q_i^1 = \frac{-1}{(s_i \mu (1-\beta))^2} \quad (3-22)$$

$$q_i^2 = \frac{1-2\beta}{s_i \mu (1-\beta)^2} \quad (3-23)$$

$$q_i^3 = \frac{\beta(2-3\beta)}{(2(1-\beta)^2)} + \ln(s_i (1-\beta)) \quad (3-24)$$

The localization problem in (3-19) is modified as follows:

$$\min_{P_0, \alpha, x, y} g(P_0, \alpha, x, y | P_{ri}, i = 1 \dots 4)$$

$$\text{where } g(P_0, \alpha, x, y) = f(P_0, \alpha, x, y) - \mu \left(\begin{array}{l} \Phi(P_0 - P_{0,min}) + \Phi(P_{0,max} - P_0) \\ + \Phi(\alpha - \alpha_{min}) + \Phi(\alpha_{max} - \alpha) \\ + \Phi(x - x_{min}) + \Phi(x_{max} - x) \\ + \Phi(y - y_{min}) + \Phi(y_{max} - y) \end{array} \right) \quad (3-$$

25)

Thus the nonlinear constrained optimization problem (3-11) – (3-12) has been transformed into a nonlinear unconstrained optimization problem (3-25). The problem (3-25) can be solved by using some popular optimizers such as Particle Swarm Optimization which will be discussed in the next section.

3.4.3 Particle swarm optimization

There are several optimization methods for solving the unconstrained problems. They can be classified as deterministic and stochastic. Deterministic methods often start with a particular potential solution. In the next iteration, the solution moves to another location nearer to the optimum. At each step, the moving vector is uniquely determined.

Stochastic methods, on the other hand, often execute the optimization by using a population of potential solutions. Each individual in the population interacts with others to produce an individual that is better with respect to the value of the fitness function. Stochastic methods sometimes perform better than deterministic methods, in the sense that the latter might face ill-conditioned problems [77].

There is an increasing interest in a class of stochastic methods, which is nature-

inspired optimization [81-82]. Some examples are Genetic algorithm (GA), Particle Swarm Optimization (PSO), Harmony Search (HS), etc. The nature-inspired optimization imitates the natural processes such as the evolution theory, bird-flocking or fish-schooling. It outperforms the deterministic optimizations in optimal scheduling problems [83]. They are also shown to be effective in solving multi-objective optimization problems [84-86].

In this study, the PSO approach is considered for searching the best P_0 , α , x , y that minimize the objective function in (3-25) as a population-based heuristic method. For the PSO optimization, it is first initialized with a population of random solutions, also called particles. Unlike genetic algorithm (GA), each particle keeps track of its best fitness solution (also known as *pbest*). Moreover, the global best fitness solution (also known as *gbest*) is also stored in the optimizer. In each time step, the PSO optimizer changes the velocity so that the particle will move towards its *pbest* and *gbest* with a random weight. In this investigation, each particle consists of four members, namely the path loss parameters P_0 , α , and coordinates of the unknown node, x and y .

Observing from (3-25), the *barrier parameter* μ has a powerful affection on the optimizing problem. If μ is large, we penalize the constraint violations more severely. As μ reaches 0, the solution approaches a solution of (3-9). In this problem, a small constant of 5 is used for μ . The tuning of PSO parameters will be discussed in chapter 4.

PSO implementation is described step by step as follows:

Step 0: Initialize the PSO parameters: population size, inertia weight ω , accelerations c_1 and c_2 . Choose the barrier parameter μ .

Step 1: Measure the RSSIs between the nodes A, B, C, D and O .

Step 2: Generate a random population of N particles, a random initial set of velocities, $pbest$ and $gbest$. Each particle has 4 dimensions which are α, P_0, x and y .

Step 3: Calculate the fitness function (3-25) for every particle.

Step 4: Compare each particle's fitness with its $pbest$. If it is better, then its value and $pbest$ are updated. The best fitness value among the $pbest$ is used as the $gbest$.

Step 5: Let c_1 and c_2 be two constants. Set the member velocity v of each particle as follows:

$$v_{i,P_0}(k+1) = \omega v_{i,P_0}(k) + c_1 \lambda_1 (pbest_{i,P_0} - P_{0i}(k)) + c_2 \lambda_2 (gbest_{P_0} - P_{0i}(k)) \quad (3-26)$$

where $i = 1, 2, \dots, N$, v_{i,P_0} represents the velocity of the i^{th} particle for P_0 , and ω is the inertia weight that controls the exploration of local and global search space. λ_1 and λ_2 are random numbers between 0 and 1.

The constants c_1 and c_2 are also known as the acceleration constants. They are the cognition and social components respectively that determine how the speed a particle would accelerate towards the $pbest$ and the $gbest$ [87].

Similarly, we obtain the velocity for other members α, x , and y as follows:

$$v_{i,\alpha}(k+1) = \omega v_{i,\alpha}(k) + c_1 \lambda_1 (pbest_{i,\alpha} - \alpha_i(k)) + c_2 \lambda_2 (gbest_{\alpha} - \alpha_i(k)) \quad (3-27)$$

$$v_{i,x}(k+1) = \omega v_{i,x}(k) + c_1 \lambda_1 (pbest_{i,x} - x_i(k)) + c_2 \lambda_2 (gbest_x - x_i(k)) \quad (3-28)$$

$$v_{i,y}(k+1) = \omega v_{i,y}(k) + c_1 \lambda_1 (pbest_{i,y} - y_i(k)) + c_2 \lambda_2 (gbest_y - y_i(k)) \quad (3-29)$$

From (3-26) – (3-29), it is observed that the new velocity at time step $k+1$ is influenced by the old velocity at time step k by the inertia weight. In this investigation, the inertia weight is varied linearly according to the following equation:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{k_{max}} \times k \quad (3-30)$$

where k is the current iteration number and k_{max} is the maximum iteration number.

Step 6: The particle position is modified by adding the new velocity to the existing particle position as follows:

$$P_{0i}(k+1) = P_{0i}(k) + v_{i,P_0}(k) \quad (3-31)$$

$$\alpha_i(k+1) = \alpha_i(k) + v_{i,\alpha}(k) \quad (3-32)$$

$$x_i(k+1) = x_i(k) + v_{i,x}(k) \quad (3-33)$$

$$y_i(k+1) = y_i(k) + v_{i,y}(k) \quad (3-34)$$

Step 7: If the number of iteration k is less than the maximum number of iterations k_{max} and the stopping criterion is not met, repeat step 3. Otherwise, proceed to step 8.

Step 8: The resulting *gbest* gives the best α , P_0 , x and y .

Stopping criteria

The algorithm is considered converge when no further progress can be made and the velocities given in (3-26) – (3-29) of the particles are negligible. In our problem, PSO will terminate the optimization process when the velocity becomes a small number, which is 10^{-4} in this study.

3.5 Test case

3.5.1 Standard PSO versus PSO with classical log-barrier

To test the effectiveness of the developed optimizer, a simulation is conducted for a representative test case. Let’s consider the case that A, B, C, D are positioned at $[0,0]$, $[120,0]$, $[120,120]$, and $[0,120]$ for an area of 120×120 . O is located at $[30,60]$ and the RSSIs between O and A, B, C, D are -85.2308, -92.6884, -90.8974, -84.5105.

If the optimizer is the standard PSO without any constraints, the PSO cannot converge for this test case. It continuously runs until the maximum number of iterations is reached. Table 3-1 shows the values p_0, α, x, y of the best particle in the swarm through iterations. From the result, it is observed that the swarm population is continuously moving to update the new positions without constraints. The value of p_0 eventually goes beyond the boundary, which is $[-60, -20]$.

Table 3-1 PSO performance without constraints

Iteration	p_0	α	x	y
100	-27.536	3.150	28.205	63.192
500	-16.283	3.734	33.592	63.046
1000	-12.619	3.924	34.954	63.004
1500	-4.129	4.364	37.702	62.721
2000	1.890	4.677	39.255	62.735

If the constraints given in (3-11) are taken into account by changing the PSO, such that

$$p_0 \leq p_{min} \rightarrow p_0 = p_{min} \qquad p_0 \geq p_{max} \rightarrow p_0 = p_{max} \qquad (3-35)$$

Table 3-2 shows the corresponding simulation result. From the results, it is observed that no further progress can be made when one of the constraints is violated. Consequently, the estimated value of p_0 is not accurate.

By using the proposed PSO with log-barrier scheme and a barrier parameter $\mu=5$, the PSO converges to the following result: $p_0 = -31.170$, $\alpha = 2.962$, $x = 27.990$, $y = 63.417$ which is shown in Table 3-3.

In conclusion, PSO converges to the solution with the log-barrier function added. The accuracy of the developed optimizer will be investigated in the next chapter.

Table 3-2 PSO performance with constraints

Iteration	p_0	α	x	y
100	-32.368	2.898	24.837	63.363
300	-29.111	3.068	27.201	63.347
500	-28.040	3.123	27.918	63.279
700	-25.760	3.242	29.324	63.234
900	-22.642	3.404	30.891	63.134
1100	-21.848	3.445	31.473	63.172
1300	-20	3.541	32.067	63.171
1500	-20	3.541	32.141	63.127

Table 3-3 PSO performance with log-barrier constraints

Iteration	p_0	α	x	y
50	-31.144	2.963	28.094	63.254
100	-31.188	2.962	27.970	63.416
150	-31.170	2.962	27.991	63.416
200	-31.170	2.962	27.990	63.417
250	-31.170	2.962	27.990	63.417
300	-31.170	2.962	27.990	63.417
450	-31.170	2.962	27.990	63.417

3.5.2 Modified barrier function versus classical barrier function

In the modified barrier function, the feasible region is enlarged. This enables the optimizer to converge to locations on the boundary. In this section, we test the

effectiveness of the function at various locations, especially those near the boundary. Table 3-4 compares the results obtained by using the classical barrier function and results obtained by using the modified barrier function. In the table, the actual values of P_0 , α , x and y are also shown.

Table 3-4 Classical barrier function versus modified barrier function

Real				Classical barrier function				Modified barrier function			
P_0	α	x	y	P_0	α	x	y	P_0	α	x	y
-30	3	0	60	-33.2	2.98	3.78	68.9	-27.8	3.17	0	67
-30	3	10	115	-26.4	3.18	10.8	110.6	-44.3	2.17	6.22	118.7
-30	3	10	120	-33.2	2.78	0.49	112.2	-37.9	2.58	0.62	114.1
-30	3	50	0	-26	3.17	95.6	24.94	-24.7	3.20	71.8	1.78
-30	3	120	30	-35.5	2.61	79	21.3	-29.2	3.05	103	22.5

Table 3-5 Localization errors: classical barrier function versus modified barrier function

Real				Classical barrier function			Modified barrier function		
P_0	α	x	y	P_0	α	Position	P_0	α	Position
-30	3	0	60	3.2	0.02	9.66	2.2	0.17	7
-30	3	10	115	3.6	0.18	4.47	14.3	0.83	5.28
-30	3	10	120	3.2	0.22	12.29	7.9	0.42	11.08
-30	3	50	0	4	0.17	51.97	5.3	0.2	21.87
-30	3	120	30	5.5	0.39	41.91	0.8	0.05	18.58

Table 3-5 shows the corresponding errors. It is observed that the errors of P_0 and α are smaller when the classical barrier function is used as compared with the modified barrier function. It is because the modified barrier function is introduced so that the optimum result can reside on the boundary, while the classical barrier function cannot produce such a result. In Table 3-5, the unknown node resides on or very close to the boundary. The classical barrier function produces results that are not very accurate, whereas the results obtained from the modified barrier function are closer to the boundary and more accurate. However, in exchange of the accuracy, the values of P_0 and α are sacrificed, which is acceptable in this case because the localization accuracy is more important. If the unknown node is not on the boundary and the path loss parameters reside on the boundary, the errors of P_0 and α obtained from the modified barrier function will also be smaller than those obtained from the classical barrier function.

It is observed from Table 3-4 that most of the locations of the unknown node reside on the boundaries, except location [10,115]. From the results, we may conclude that the modified barrier function gives a more accurate estimation especially when the unknown location is on the boundary.

3.6 Conclusion

In this chapter, a RSS-based localization approach has been developed. Because of the high variance of RSSI measurement, a real time localization technique is introduced. The formulated problem is a constraint nonlinear optimization that cannot be solved by most optimizers. PSO with log-barrier is introduced to overcome the weaknesses of

the existing PSO scheme. The effectiveness of the developed optimizer is discussed and compared with other optimization techniques, such as standard PSO and PSO with classical barrier function using a test case. Detail study with simulation and experiment to validate the algorithm will be presented in chapters 4 and 5.

Chapter 4

Performance analysis and parameters tuning of the proposed algorithm

4.1 Introduction

In this chapter, simulation is conducted to validate the proposed localization approach. For each simulation study, it consists of two stages: 1. generate the RSSI measurement and 2. Perform the localization using the proposed PSO with log-barrier scheme. For the RSSI measurement, it is generated using a Gaussian distribution with a standard deviation σ in dB. For the simulation study, the value of σ is varied in a wide range from 1 to 14 to cover most of the RSSI measurement's error in real application. For the PSO, it is implemented with the following parameters:

- Population size: $ps=30$
- Maximum generations to be evaluated: 500
- Members of particle: P_0, α, x and y
- Inertia weight factor $\omega_{max}=0.9, \omega_{min}=0.4$
- Acceleration constant $c_1=2$ and $c_2=2$
- Barrier parameter $\mu = 5$

In general, the path loss exponent is in the range of 1.6 to 6.5 [48]. Moreover, P_0 at 1m is in the range of -25 to -45dB [48]. The other four boundary values are set to their

respective location bounds.

In this chapter, the simulation study is conducted using Matlab version 7.4.0 on a personal computer running at 2.66GHz. The proposed algorithm is tested with many different anchor nodes and unknown node's locations to determine its robustness. For all the simulation study, the error is calculated as the Euclidean difference between the Cartesian true coordinate and the estimated coordinates of the unknown node. The purpose of the simulation is to study the PSO's convergence and the localization accuracy.

4.2 PSO convergence

To estimate the path loss parameters and the unknown coordinates, the PSO optimizer needs to converge to an optimum. In this section, the PSO convergence is studied to ensure a statistically unbiased result. The rate of convergence is also observed. The anchor nodes and the unknown node A , B , C , D and O are placed at 5 different sets of locations, which are generated randomly. For each set of location, the PSO is executed to determine the position of O based on simulated RSSI using (3-3) with $\sigma=3\text{dB}$. For the same set of locations, the simulation is repeated 30 times with different set of RSSI measurements. To present a uniform graph, each run is terminated only when the maximum number of generations (i.e. 1000) has been elapsed. The value of the fitness function g_{best} of each run is kept. Fig. 4.1 plots the average value of g_{best} over 30 independent trials versus the iterations. From the figure, it is observed that the PSO always converges, irrespective of the sensor nodes' positions. Moreover, the fitness function reduces rapidly and started to converge after 20 iterations. Overall, the

execution time takes less than a second. It is noticed that the objective function may become negative. When the PSO reaches a solution that is close to the actual path loss parameters and the unknown location, the original objective function f is very small. With the log-barrier function yields a negative value, the overall objective function g becomes negative.

Fig. 4.1 PSO convergence with $\sigma = 3\text{dB}$

Investigation with other RSSI standard deviations has also been conducted. Fig. 4.2 to Fig. 4.4 plot the average g_{best} versus iterations with $\sigma=1\text{dB}$, 5dB , and 7dB . From these results, it is observed that the PSO converges to the minimum of the objective function (3-25) and is independent of the anchor nodes and unknown node's positions. It is also independent on the error in RSSI measurement. From these results, it is concluded that the PSO often converges after 20 iterations.

Fig. 4.2 PSO convergence with $\sigma = 1\text{dB}$

Fig. 4.3 PSO convergence with $\sigma = 5\text{dB}$

Fig. 4.4 PSO convergence with $\sigma = 7\text{dB}$

4.3 Effect of PSO parameters on localization performance

There are a few parameters that might affect the PSO's performance in terms of location error and execution time. These PSO parameters are: population size ps , acceleration constants c_1 and c_2 .

The population size ps is the number of particles that will interact with others and move towards the optimum. In this study, the population size is varied from 20 to 70 with a step size of 10. 30 sets of anchor nodes's locations in an area of 100×100 are first randomly generated. For each set of locations, the unknown node O is placed at 100 different grid points with a spacing of 10 units, ranging from $[5,5]$ to $[95,95]$. The localization performance is assessed in terms of localization error and execution time. The total number of simulations is 3000 (i.e. 30 sets of anchor node locations $\times$ 100

unknown node locations). After every simulation, the localization error and execution time are determined. The final localization error and execution time are taken by averaging from the 3000 results obtained. Fig. 4.5 plots the PSO performance versus population size. From the figure, it is observed that the execution time increases linearly with respect to ps . This is due to the fact that larger ps leads to larger number of particles. Consequently, more computation is performed. From the results, it is also observed that increasing ps does not necessarily improve the location estimation. From the figure, it is observed that $ps=40$ is a good choice with a small average error and a reasonable execution time.

Fig. 4.5 PSO performance versus population size

Besides the population size, the acceleration constants c_1 and c_2 are the other parameters that affect the optimum searching. They are the cognition and social components respectively that determine the speed a particle would accelerate towards the $pbest$ and the $gbest$. Similar simulation is conducted with different pairs of acceleration constants. Tables 4-1 and 4-2 show the execution time (in seconds) and the localization error versus c_1 and c_2 . Figs 4.6 and 4.7 plot the data corresponding to Tables 4-1 and 4-2 respectively. Fig. 4.6 shows that the execution time increases linearly with increasing c_1 and c_2 . From the figure, it is observed that the larger the acceleration constants, the longer it takes to converge. It is noted that larger values of c_1 and c_2 lead to larger velocities change. This causes the particles to move faster. It was then harder for the algorithm to converge. The convergence time is also longer. Similar observation is also reported in [92].

Table 4-1 Execution time versus c_1 and c_2

$c_1 \backslash c_2$	1	1.5	2	2.5	3	3.5
1	0.2921	0.3687	0.5106	0.6175	1.0318	1.7725
1.5	0.3631	0.5123	0.7078	0.9467	1.5546	2.2909
2	0.5012	0.8313	1.1236	1.5379	2.3401	2.6481
2.5	0.8790	1.1871	1.5803	2.3206	3.0399	3.2139
3	1.3104	1.6227	2.0960	2.7880	3.4164	3.3489
3.5	1.7587	2.4700	3.1104	3.3949	3.8695	3.6717

Fig. 4.6 Execution time versus c_1 and c_2

Table 4-2 Location error versus c_1 and c_2

$c_2 \backslash c_1$	1	1.5	2	2.5	3	3.5
1	19.0628	20.0024	18.9259	20.4589	19.4541	20.5742
1.5	19.1902	20.4193	18.5911	19.1458	19.0867	20.5613
2	19.1665	18.5858	20.6540	18.3208	19.1204	19.0199
2.5	18.8699	20.2280	20.6273	20.6208	20.5245	18.9049
3	20.2108	20.0789	19.2820	20.0194	19.5852	20.7850
3.5	19.3890	19.9053	20.0890	19.7603	19.6393	20.1051

Fig. 4.7 Localization error versus c_1 and c_2

Table 4-3 Location error versus c_1 and c_2 (extended investigation)

$c_1=c_2=1$	$c_1=c_2=2$	$c_1=c_2=3$	$c_1=c_2=5$	$c_1=c_2=7$	$c_1=c_2=10$	$c_1=c_2=15$
19.0628	20.6540	19.5852	20.1935	20.1466	36.3520	36.7793

From Table 4-2, it is observed that the localization error fluctuates from 18 to 20. The investigation on effect of c_1 , c_2 on the localization is therefore extended. Table 4-3 shows the results. From the table, it is shown that when the acceleration constants are too large, the optimizing algorithm does not function well. The location error is suddenly increased by 50% with $c_1=c_2=10$. It is because when the acceleration constants are too large, the steps that the particles move will also be large. Hence, the optimum is often missed and results in large location error. Thus, c_1 and c_2 should be in the range of 1 to 7.

4.4 Localization performance under different RSSI standard deviations

To test the robustness of the algorithm, the unknown node is placed at various locations inside $ABCD$. In the following study, the locations of A, B, C, D are set as $[0,0], [120,0], [120,120], [0,120]$. The area $ABCD$ is divided into a grid with 10 unit spacing from $[1,1]$ to $[111, 111]$. Figs 4.8 – 4.10 show the average error obtained from running the algorithm for 50 times at every grid point at different RSSI standard deviation $\{1\text{dB}, 3\text{dB}, 5\text{dB}\}$.

From the figures, we may conclude that larger fluctuation in RSSI measurement would result in larger error in localization. Table 4-4 and Fig. 4.11 summarize the error in estimating P_0 , α or location which are obtained from Figs 4.8 – 4.10. In Fig. 4.11, there are three different line styles representing the mean, maximum and minimum errors. It is observed that errors in estimating the path loss parameters and the unknown positions increase when the errors in RSSI measurement increase. The result shows that the error of α is in the range of $0.04 - 0.09$. The error of P_0 is around 3 units. The error in estimating the unknown location can be as small as 0.1062 units while the worst case error is 19.1322 units (15.94% of the dimension of the deployment area). With a RSSI standard deviation of 3dB, i.e. 26% error in measuring received signal strength, the average localization error is 3.8301 units (3.19% of the dimension of the deployment area).

Fig. 4.8 Estimation error at different locations ($\sigma = 1\text{dB}$)

Fig. 4.9 Estimation error at different location ($\sigma = 3\text{dB}$)

Fig. 4.10 Estimation error at different locations ($\sigma = 5\text{dB}$)

Table 4-4 Average error versus standard deviation

σ		1dB	3dB	5dB
Error of P_0	Max	3.9691	4.0567	5.9432
	Mean	1.5011	2.5856	3.7060
	Min	0.6528	1.4824	2.1045
Error of α	Max	0.1848	0.2133	0.2932
	Mean	0.0798	0.1378	0.1984
	Min	0.0355	0.0877	0.1340
Error of Position	Max	4.1408	10.8378	19.1322
	Mean	2.0351	3.8301	6.3199
	Min	0.1062	0.3147	0.4176

Fig. 4.11 Average error versus standard deviation

4.5 Effect of barrier parameter on localization accuracy

The barrier parameter μ in (3-25) is a weight of the boundary constraints on the overall objective function [77]. If μ is set too small, its effect on the boundary constraints might be negligible. On the other hand, if μ is set too big, the optimized objective function (3-25) might be much different from the original objective function (3-12). To investigate the effect of the barrier parameter on the final solution, the algorithm is executed with μ varied from 0.01 to 5. Table 4-4 and Fig. 4.12 show the average error from 50 simulations with a RSSI standard deviation of 3dB.

Table 4-5 Average error versus μ with $\sigma = 3\text{dB}$

μ		0.01	0.1	1	5	7	10
Error of P_0	Max	13.9370	12.4382	8.236	4.0567	3.9077	3.8442
	Mean	11.2761	9.7534	5.5049	2.5856	2.1764	1.8388
	Min	8.0858	7.2482	2.7934	1.4824	1.3869	1.3359
Error of α	Max	0.7270	0.6036	0.406	0.2133	0.1987	0.1789
	Mean	0.5748	0.4969	0.2814	0.1378	0.1184	0.1016
	Min	0.4170	0.3875	0.158	0.0877	0.0757	0.0742
Error of Position	Max	23.8621	23.4051	20.9055	21.0865	19.939	23.0012
	Mean	17.4830	16.8861	15.1193	14.2501	14.2617	14.1552
	Min	1.5537	1.3450	0.9211	0.5620	0.5101	0.623

Fig. 4.12 Average error versus barrier parameter μ with $\sigma = 3\text{dB}$

From Fig. 4.12, it is observed that the errors in estimating P_0 , α and position are large when the barrier parameter are smaller than 0.1. With the barrier parameter equals to 5, the boundary constraints are significant and the optimum is not too far from the original optimum. However, if the barrier parameter is increased beyond 5, the errors do not improve significantly. Hence, the barrier parameter is fixed at 5 in the later discussion.

4.6 Comparison between PSO – log barrier and memetic algorithms

Modern stochastic algorithms draw inspiration from biological evolution. Evolutionary algorithms are capable of exploring and exploiting promising regions of the search space. However, they can take a relatively long time to locate the exact local optimum in a region of convergence. Opposed to stochastic algorithms, there are some local searches such as steepest descent, Newton methods, etc which take a short time to converge to an optimum. However, they might be trapped in a local optimum.

Memetic algorithm is designed to achieve better exploration and exploitation using a combination of dedicated population based and individual learning searches (local searches). It has the potential of exploiting the complementary advantages of stochastic optimizer and problem-specific local search [88]. The memetic algorithm not only is general, robust with global search efficiency but also converges rapidly toward local minima.

Among the many evolutionary algorithms, Differential Evolutionary algorithm (DE) has been proven to be a promising candidate to solve real-valued optimization

problems [89]. The main structure of the DE consists of initialization, mutation, crossover and selection operations. DE is a parallel direct search method which makes use of a set of multidimensional parameter vectors. The key element distinguishing DE from other population-based techniques is the differential mutation mechanism. Together with arithmetic crossover, this gives the different dynamic effects of search.

The problem (3-12) is a non-linear least square problem. A popular individual learning search for this type of optimizing problem is the Gauss-Newton (GN) method [43]. To find the best update velocity, the GN algorithm calculates the inversion of a matrix. If the matrix is singular or close to singular, the calculated velocity will not be accurate. The Marquardt-Levenberg (ML) method overcomes the GN's singular problem [77]. Different from the GN method, the ML method uses the trust region strategy instead of the line search strategy. The use of trust region avoids one of the weaknesses of GN which is its behavior when the Jacobian matrix is singular or nearly so. In the trust region method, it approximates the objective function by a modal function (often a quadratic). Then it steps to the minimizer of the modal function, which yields a significant reduction in the objective function and is better than the line search method. In the following study, DE and ML are chosen to form a memetic algorithm.

Table 4-5 shows the steps in executing a memetic algorithm by using DE and Marquardt-Levenberg algorithms.

Table 4-6 Memetic algorithm (DE and Marquardt - Levenberg)

1. Generate the initial population
$P_{0i}(k) = P_{0min} + R_{p_0}(P_{0max} - P_{0min})$ $\alpha_i(k) = \alpha_{min} + R_{\alpha}(\alpha_{max} - \alpha_{min})$

$$\begin{aligned}x_i(k) &= x_{min} + R_x(x_{max} - x_{min}) \\y_i(k) &= y_{min} + R_y(y_{max} - y_{min})\end{aligned}$$

where R_{p_0} , R_α , R_x and R_y are a random numbers.

2. Mutation: Mutant vectors are generated as follows:

$$v_i(k) = [v_{p_0} \quad v_\alpha \quad v_x \quad v_y]_i(k) = m_{best}(k) + P(m_\alpha(k) - m_\beta(k))$$

where m_{best} , m_α , m_β represent the best and two arbitrary members of the population at generation k . The $P \in [0,2]$ is a constant mutant vector.

3. Crossover: New vectors are created as follows:

$$\begin{aligned}u_{p_0,i}(k) &= \begin{cases} v_{p_0,i}(k), & \text{for } R_{p_0} \leq C_R \\ p_{0,i}(k), & \text{otherwise} \end{cases} \\u_{\alpha,i}(k) &= \begin{cases} v_{\alpha,i}(k), & \text{for } R_\alpha \leq C_R \\ \alpha_i(k), & \text{otherwise} \end{cases}\end{aligned}$$

$$\begin{aligned}u_{x,i}(k) &= \begin{cases} v_{x,i}(k), & \text{for } R_x \leq C_R \\ x_i(k), & \text{otherwise} \end{cases} \\u_{y,i}(k) &= \begin{cases} v_{y,i}(k), & \text{for } R_y \leq C_R \\ y_i(k), & \text{otherwise} \end{cases}\end{aligned}$$

where $C_R \in [0,1]$ is a constant.

4. Selection: the next generation members in the population are evaluated as follows:

$$m_i(k+1) = \begin{cases} v_i(k), & \text{for } O(v_i(k)) < O(m_i(k)) \\ m_i(k), & \text{otherwise} \end{cases}$$

5. Select a random subset from the new population to undergo individual improvement
6. Marquardt – Levenberg: perform ML on each member m^* of the subset:

- a. Initialization: Set $m^0 = m^*$. Choose the trust region size Δ .
- b. At step k , calculate the Jacobian matrix J_k , and

$$r_k = \left[m_{k,p_0} - 10 \cdot m_{k,\alpha} \cdot \log \left(\frac{\sqrt{(m_{k,x} - x_i)^2 + (m_{k,y} - y_i)^2}}{2\sigma^2} \right) \right]_{i=1,\dots,4}$$

and form the modal function as follows:

$$M_k(p) = \frac{1}{2} \|r_k\|^2 + p^T J_k^T r_k + \frac{1}{2} p^T J_k^T J_k p$$

- c. Find p^{GN} (if possible) such that: $J_k^T J_k^T p^{GN} = -J_k^T r_k$. If $\|p^{GN}\| < \Delta$, then $m_{k+1} = m_k + p^{GN}$. Go to b.
- d. If cannot find p^{GN} , find λ such that there are orthogonal matrix Q_λ and upper triangular matrix R_λ satisfying: $\begin{bmatrix} R_\lambda \\ 0 \end{bmatrix} = Q_\lambda^T \begin{bmatrix} J \\ \sqrt{\lambda} I \end{bmatrix}$ and $R_\lambda^T R_\lambda = J_k^T J_k + \lambda I$.
- e. Find p^{LM} such that: $(J_k^T J_k + \lambda I)p^{LM} = -J_k^T r_k$. Then, $m_{k+1} = m_k + p^{LM}$. Go to b.

7. Update the population with the new member obtained after performing individual learning on the subset. Go to 2.

Memetic algorithm and PSO log-barrier are both applied to the original localization problem (3-11) – (3-12). By adding the log-barrier function, the PSO will optimize the objective function (3-25), whereas memetic algorithm will optimize the objective function (3-11) – (3-12). The purpose is to show the effectiveness of using log-barrier to optimize constrained non-linear problem. The unknown node's location $[x,y]$ is assumed inside a square of 120×120 . The locations of the four anchor nodes are: $[0,0]$, $[120,0]$, $[120,120]$, $[0,120]$. There are 8 different cases, in which four variables p_0 , α , x , and y are tuned. As in previous simulation study, the RSSI information is generated following (3-3), with the standard deviation X_σ of 1, 2, 3 or 4. Table 4-6 shows the results.

Table 4-7 Memetic algorithm (DE-ML) versus PSO with log-barrier constraint

		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Real	p_0	-30	-30	-30	-40	-40	-40	-30	-30
	α	3	3	3	4	4	4	4	3
	x	12.5	37	56	25	25	50	50	100
	y	25	89	11	100	25	25	70	70
MA (DE-ML)	p_0	-20	-48.96	-25.22	-20	-57.56	-20	-20	-49.69
	α	3.49	2	3.22	5	3.06	5.06	4.38	2
	x	15.73	32.11	52.83	27.81	33.96	57.95	53.86	93.45
	y	28.93	100.35	0	99.78	0	38.11	70.95	85.42
PSO – log barrier	p_0	-36	-26.57	-24.59	-29.87	-46.99	-39.23	-31.35	-27.86
	α	3	3.15	3.27	4.49	3.61	4.06	3.79	3.12
	x	13.52	40	52.76	25.79	37.87	58.24	53.07	81.12
	y	27.11	86.79	7.5	102.97	8.36	32.31	72.57	76.20

From the table, it is observed that the MA reaches the boundaries in every case. The estimated parameter that is on the boundary is highlighted in the table. During the memetic algorithm’s execution, when p_0 , α , x , or y of a particle goes beyond its lower or upper bounds, it will be reset to the corresponding lower or upper bounds as shown below:

$$P_0 \leq P_{0min} \rightarrow P_0 = P_{0min}; \quad P_0 \geq P_{0max} \rightarrow P_0 = P_{0max}$$

$$\alpha \leq \alpha_{min} \rightarrow \alpha = \alpha_{min}; \quad \alpha \geq \alpha_{max} \rightarrow \alpha = \alpha_{max}$$

$$x \leq x_{min} \rightarrow x = x_{min}; \quad x \geq x_{max} \rightarrow x = x_{max}$$

$$y \leq y_{min} \rightarrow y = y_{min}; \quad y \geq y_{max} \rightarrow y = y_{max}$$

As shown in Table 4-6, the MA cannot find the optimum inside the boundary. The best member in the population always goes beyond the boundary. From the result, it is concluded that the MA does not converge.

On the other hand, the PSO with log-barrier performs well. It converges to a solution within the boundaries, and the solution is relatively close to the actual data.

4.7 Performance comparison

In this section, we evaluate the proposed PSO with log barrier approach and compared it with multilateration approach as well as Sequence Based Localization (SBL) scheme [21]. Multilateration is a popular localizing method [90] with the distances estimated using RSSI, TOA or other ranging methods. To compare it with the proposed method, the distances in multilateration are obtained by using the RSSI as follows:

$$d = d_0 \cdot 10^{\frac{P_0 - RSSI}{10\alpha}}$$

where the path loss parameters P_0 and α are determined in advance by using a set of prior RSSI measurement.

Different from the proposed method, multilateration method requires the estimated distance between the unknown and the anchor nodes. Hence the path loss parameters must be known. In the following simulation for multilateration, the exact path loss

parameters are $P_0=-30\text{dB}$ and $\alpha=4$. With very accurate RSSI measurement, the calculated path loss parameters are exactly $(-30,4)$. In erroneous measurement, the calculated path loss parameters are assumed to be $(-31,4)$, $(-33,4.4)$ or $(-36,4.8)$. Multilateration's performance is tested with the estimated path loss parameters (-30.4) , $(-31,4.1)$, $(-33,4.4)$, or $(-36,4.8)$. Fig. 4.13 shows the flowchart for performing the localization simulation using the multilateration approach.

To provide a fair comparison, simulations are conducted using the same scenarios and condition as in [21] so that the SBL's localization result can be used. The sensor field is set as a square of 100×100 . The path loss parameters are $P_0=-30\text{dB}$ and $\alpha=4$. Furthermore, it is assumed that every sensor inside the field is able to communicate with one another. For the system, there are four anchor nodes randomly placed inside the square for 100 different locations. For each set of anchor nodes' positions, the unknown node is randomly assigned at 100 different locations. Hence, we have a total of 10,000 simulations for a given set of simulation parameters. The localization error is averaged from these 10,000 simulations. The simulation results are presented in Fig. 4.14 by varying the RSSI standard deviation.

Fig. 4.13 Simulation flowchart for multilateration approach

Fig. 4.14 shows the average localization error at different RSSI standard deviation σ for the proposed PSO-log barrier, SBL and multilateration method. From the figure, it is observed that the multilateration method has the worst performance. The results

show that the localization error increases dramatically when the RSSI standard deviation increases. The localization error reaches 120 units when the RSSI standard deviation is 8dB. For the SBL and PSO with log-barrier methods, the localization error increase gradually with RSSI error. The PSO with log-barrier approach achieves a localization error 8 meters smaller than the SBL. In conclusion, the proposed PSO-log barrier method outperforms the other methods. It consistently achieves the smallest errors with σ varies from 0 to 8 dB. The exception is when σ is 0dB for the multilateration approach. However, this does not happen in real environment.

Fig. 4.14 Localization error versus RSSI standard deviation for different location methods

Fig. 4.15 Average localization error of the proposed method versus α and σ

Fig. 4.16 Average error versus σ

The proposed localization method has also been evaluated with other path loss parameters, i.e. $\alpha=[2,6]$, and $\sigma=[2,14]$ dB. The average localization error is plotted in Figs. 4.15 and 4.16. From these figures, it is observed that the localization error increases with increasing standard deviation σ , and decreases with increasing path loss exponent α .

4.8 Conclusion

In this chapter, various simulation studies have been conducted with different sensor nodes' locations and RSSI error standard deviation. From the results, it is observed that the proposed algorithm performs well in every situation.

The effect of PSO's parameters on the localization performance has been investigated. It is found that these parameters have an impact on the localization error. With $p_s=40$, $c_1=2$, $c_2=2.5$, the localization error is the smallest. Moreover, an increase in the population size or the acceleration constants c_1 or c_2 leads to longer convergent time. The smallest execution time is 0.29 seconds for one optimization.

The effect of the barrier parameter μ has also been investigated. It is observed that large error occurs with small barrier parameter (i.e. $\mu<0.1$). On the other hand, the error does not reduce significantly if $\mu>5$. Thus, it is deduced that $\mu=5$ is a suitable choice for the scenario used, where the localization error is the smallest.

The proposed algorithm is effective in reducing the localization error. For a 4dB RSSI measurement error (about 26 % in measuring received signal strength), the average localization error obtained from the proposed method is 19%. It outperforms the

multilateration and Sequence Based Localization (SBL) approaches that have errors of 32% and 27% respectively.

The proposed method will be validated by real experiment presented in chapter 5. The method will also be extended to other scenarios. The modified method will also be compared to the SBL method. These will be presented in chapter 6.

Chapter 5

Prototyping and experimental results

5.1 Introduction

In this chapter, the proposed algorithm presented in chapter 3 will be validated by experiments. A star network consisting of one coordinator and four routers has been setup. For the experiment, the routers' locations are known, while the coordinator's location is to be estimated. Fig. 5.1 shows the system setup of the experimental network. For the system, each sensor node consists of a RF transceiver and a processing unit to control its operation. In this setup, the XBee Znet 2.5 RF module is used. The coordinator (XBee XB24-BxIT-002) is connected to a laptop; while the router (XBee XB24-BxIT-004) is connected to a microcontroller PIC18F4620. The coordinator will control the network's operation and gather RSSI information. From the RSSI information, the unknown location of the coordinator is estimated by using the proposed localization method. The RSSI information can be relayed to a remote centre for localization processing or alternatively performed using the coordinator's laptop.

In this chapter, the hardware is first briefly described. The RF module's serial interface and Zigbee protocols are then discussed in details. The data communication process of the sensor nodes and RSSI measurement are also explained. The optimization algorithm has also been implemented using a low-cost microcontroller.

The experimental results are presented and benchmarked with other localization methods.

Fig. 5.1 Network setup

5.2 Experimental Prototype

To validate the proposed approach and the conducted simulation results as shown in Chapter 4, an experimental prototype has been developed to conduct various experiments in a car park located in the University campus (Fig. 5.2). For the experiments, five sets of RF modules (XBee ZNet 2.5, Digi International Inc) are used. Four of the RF modules are configured as routers *A*, *B*, *C* and *D*. These routers are placed at the far ends of the car park area forming a quadrangle with a dimension of 16m×35m. The remaining module is configured as a mobile unknown node that

could be placed anywhere inside the quadrangle. The RSSIs between O and A, B, C, D can be measured from the RF transceiver of each node.

Fig. 5.2 Experiment site

Fig. 5.3 Block diagram of the communication and measurement

For the experimental prototype, each sensor node comprises a microcontroller

(Microchip PIC18LF4620, 40 MHz) connected to a RF module through the RS232 interface. The RF module's operations are controlled through the AT commands. The microcontroller has 64k bytes of flash program memory and 3968 bytes of SRAM data memory. The proposed PSO with log barrier algorithm has been realized on the microcontroller and it occupies approximately 13k bytes program memory, and 2.2k bytes of SRAM data memory.

Fig. 5.3 shows the block diagram of one sensor node. The transceiver (XBee unit) also has a processing unit to perform communication based on Zigbee protocols, such as network formation, packet routing, error check, etc. Because the transceiver is configured to operate in AT transparent mode, whatever data it receives will be relayed to the microcontroller. An important feature of the transceiver is that it automatically measures the RSSI while receiving signal. To obtain the RSSIs, the microcontroller queries from the transceiver using AT commands. After collecting enough RSSIs (e.g. 10), the PSO optimizer will perform localization. An alternative is to perform localization via a remote server by programming the sensor nodes to transmit the RSSI readings to the server.

The communication range of the RF transceiver is 40 meters for indoor and 120 meters for outdoor. The RF data rate is 250,000bps. The modulation technique uses the DSSS (Direct Sequence Spread Spectrum). Each direct sequence channel has over 65,000 unique networking addresses available. Point-to-point, point-to-multipoint and peer-to-peer topologies are supported. The communication protocol is enhanced with retries, acknowledgement. Before transmission, the route between transmitter and receiver is discovered by using route discovery process with AODV (Ad-hoc On-

demand Distance Vector routing) protocol. Security is enhanced by using AES-128 algorithm to encrypt and decrypt RF data packets.

The PIC18F4620 is a low-power microcontroller from Microchip. It has many features which allow it to interact with and process information from other devices, such as interrupts, Capture/Compare/PWM, I2C or Enhanced USART, etc. The microcontroller supports four oscillator modes; up to 40MHz. In this application, the oscillator is 4MHz. The microcontroller can be programmed in C using an In Circuit Debugger (ICD) from Microchip. The PIC16F4620 has 64Kbytes of program memory and 4096 bytes of data RAM memory.

5.2.1 Serial interface protocol

The microcontroller PIC18F4620 is connected in series with the RF module through RS232 interface. The transmission between two devices through RS232 is controlled and monitored by the Universal Asynchronous Receive Transmit (UART) modules in both devices. The module UART performs tasks such as timing and parity checking, that are needed for data communications. Serial communication depends on two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits) [91].

Fig. 5.4 System data flow diagram in a UART-interface environment

Fig. 5.4 shows the data flow datagram between the two devices. They are connected through 4 pins – DI, DO, $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$. The RF module can operate in transparent mode or in Application Programming Interface (API) mode. When operating in transparent mode, the modules act as a serial line replacement. All UART data received through the DI pin is queued up for RF transmission. When the RF data is received, it is sent out through the DO pin. API operation is an alternative to transparent operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module. When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module. In this project, the RF modules used operates in the transparent mode.

The RF module settings can also be modified through the serial interface by using AT commands. For example, command ‘BD’ sets the serial interface data rate, command ‘ND’ tells the RF module to discover and report all RF modules found, etc. The full list of AT commands can be found in [91]. To enter the AT command mode instead of normal RF transmission operation, the Command Sequence Character (CC) must be

entered, no serial data is sent before and after CC in a Guard Time (GT) period.

In normal operation, serial data is transferred from the serial buffer to the radio buffer for transmission. The transmission protocol is Zigbee which is discussed next.

5.2.2 Zigbee protocol

The XBee RF module operates based on the Zigbee protocol which is built upon the IEEE 802.15.4 protocol. Fig. 5.5 shows the layers of the Zigbee stack. There are four distinct layers, the physical layer, the MAC layer, the network layer and the application layer. The two lowest layers are defined by the IEEE 802.15.4-2003 standard. This standard defines a protocol for wireless devices to be low-power consuming. The idea is that a device will enter in sleep mode when it does not have a task to perform. Consequently, the power consumption is much lower than in normal operating state. The IEEE 802.15.4-2003 standard defines how to associate to a coordinator, disassociate from a coordinator and how to send messages between an end device to a coordinator.

Fig. 5.5 The layers of the Zigbee stack

The network layer is the lowest layer defined by the Zigbee standard. This layer defines network formation, network address discovery, route discovery, retries and acknowledgement. It is noted that there are three types of devices in a Zigbee network – coordinator, router and end device. Fig. 5.6 shows the typical functions of these devices. In the Znet 2.5 RF module series, the coordinator performs a series of scans to discover the level of RF activity on different channels (energy scan), and to discover any nearby operating PANs (PAN scan) to start a personal area network (PAN). Once the coordinator has completed the energy and PAN scans, it attempts to start on an unused PAN ID and channel. Router and end device also perform a PAN scan to find a valid Zigbee network to join. However, each coordinator and router can only support a finite number of end device children. Once that number of end devices has joined that particular router or coordinator, the device can no longer allow end devices to join it.

When a node joins a network, a 16-bit network address is assigned to it. The network address is unique to each node in the network. However, network address is not static and it can be changed. Besides, each node contains a unique and permanent 64-bit address which is assigned to it once manufactured. Instead of using the complete 64-bit address, the 16-bit address is used in communication in the network.

Zigbee device type	Typical function
Coordinator	One per network. Select the channel and ID to form a personal area network (PAN). Allow other routers and end devices to join the network. Can transmit and receive RF data, route data through the mesh network. Should be mains powered.
Router	Must join a Zigbee PAN. Allow other routers and end devices to join the PAN. Can transmit and receive RF data, route data through the network. Should be mains powered.
End device	Must join a Zigbee PAN. Cannot allow other devices to join the network, cannot route data. Can transmit and receive RF data. Intended to be battery powered.

Fig. 5.6 Zigbee protocol device types [91]

Prior to transmitting data, network address discovery is taken place so that a device knows the 16-bit network address corresponding to the unique 64-bit address of the destination device. Routers and coordinators can participate in establishing routes between source and destination devices using a process called route discovery. The Route discovery process is based on the AODV (Ad-hoc On-demand Distance Vector routing) protocol. Acknowledgement packet (Ack) is required when data is

transmitted from one node to its neighbor, and from the node that originates the transmission to the destination device. If the Ack is not received, the transmitting device will retransmit the data.

If security is enabled, the coordinator will start up using a 128-bit AES encryption key. Only device that has the same security key can communicate in the PAN. Routers and end devices that will join a secure PAN must obtain the correct security key. RF data packets are encrypted and decrypted at the network layer according to AES-128 (FIPS 197 algorithm).

Data can be sent as a broadcast, or unicast transmission. Broadcast transmissions within the ZigBee protocol are intended to be propagated throughout the entire network such that all nodes receive the transmission. To accomplish this, all devices that receive a broadcast transmission will retransmit the packet 3 times.

Unicast ZigBee transmissions are always addressed to the 16-bit address of the destination device. However, only the 64-bit address of a device is permanent; the 16-bit address can be changed. Therefore, ZigBee devices may employ network address discovery to identify the current 16-bit address that corresponds to a known 64-bit address, and route discovery to establish a route.

ZigBee includes acknowledgment packets at both the Mac and Application Support (APS) layers. When data is transmitted to a remote device, it may traverse multiple hops to reach the destination. As data is transmitted from one node to its neighbor, an acknowledgment packet (Ack) is transmitted in the opposite direction to indicate that the transmission was successfully received. If the Ack is not received, the transmitting

device will retransmit the data, up to 4 times. This Ack is called the Mac layer acknowledgment. In addition, the device that originated the transmission expects to receive an acknowledgment packet (Ack) from the destination device. This Ack will traverse the same path that the data traversed, but in the opposite direction. If the originator fails to receive this Ack, it will retransmit the data upto 2 times until an Ack is received. This Ack is called the ZigBee APS layer acknowledgment.

5.3 Data communication in the user application

Fig. 5.7 Illustration of the TDMA communication cycle

The experimental network consisting of one unknown and four anchor nodes is a simple network and can be configured as a star network. The unknown node is the coordinator while the anchor nodes are the routers. To obtain the RSSIs between the separating distance of coordinator – anchor nodes, data communication between them is needed. The coordinator will communicate with one anchor node at a time to avoid

signal interference. Fig. 5.7 shows the data communication cycle of the network.

As shown in Fig. 5.7, the unknown node as a coordinator will send a message to the anchor node 1. The anchor node 1 upon receiving the signal will reply by sending a message back to the unknown node. The unknown node then measures the RSSI. From (3-3), the RSSI is a random number with normal distribution. Hence, it is better to take many RSSI readings, average of which will give a more accurate RSSI. To get N RSSI readings, the router 1 will send N messages. The anchor node 1 will delay by a time interval T before sending another message so that the unknown node finishes measuring RSSI. How to choose T will be explained later. The larger N is, the more accurate the average RSSI and the longer the measurement. After considering between accuracy and time, N is chosen to be 10.

After receiving N messages from anchor node 1, the unknown node starts to communicate with the anchor node 2. The communication and measurement process continues until $4 \times N$ RSSIs from 4 anchor nodes have been collected. The localization algorithm is then executed.

To choose an appropriate T interval, the latency is investigated. Latency is the time when the router transmits signal until the coordinate finishes measuring RSSI. Fig. 5.8 shows the latencies involved in the process of measuring RSSI.

Fig. 5.8 Latencies involved in the process of measuring RSSI

Fig. 5.9 Latencies involved with two consecutive measurements

Fig. 5.9 shows the latencies involved in two consecutive measurements. After the time $T_{sr} + T_{air} + T_{sc}$, data has reached the microcontroller/laptop at the coordinator side. To avoid signal collision, the $(n+1)^{\text{th}}$ message should arrive when the coordinator has finished RSSI measurement for the n^{th} message. Hence

$$T \geq T_{sc} + T_{pr} + T_{qr} \quad (5-1)$$

These latencies can be determined using the CPU clock in the PC. Moreover, the data transmission time over the serial link and the air can be calculated using the RF data rates and the serial baud rates which are 250kbps and 9600bps respectively.

Table 5-1 Latencies involved with a single RSSI measurement

$T_{sc} = T_{sr}$	T_{air}	$(T_{pr}+T_{qr})_{min}$	$(T_{pr}+T_{qr})_{mean}$	$(T_{pr}+T_{qr})_{max}$
1.67ms	640μs	0.098s	0.118s	0.139s

Table 5-1 shows the latencies involved with a single RSSI measurement. It is shown that the major latency contributing to the system comes from the computer processing time and the time taken to query RSSI from the RF module. To query RSSI from the RF module, the PC needs to do the followings: (1) send the Command Sequence Character (CC) to enter the AT command mode, (2) send the ‘ATDB’ command to query RSSI, and (3) exit the AT command mode. After each transmission, the PC has to wait for the RF module’s response before proceeding. Hence, the query time is much longer than the other latencies.

From (5-1), T is chosen to be $T_{sc} + (T_{pr} + T_{qr})_{max}$, hence, $T = 0.14s$.

5.4 Embedded system’s memory constraint and execution time

To test the feasibility of the proposed PSO with log-barrier approach, the algorithm has been implemented using a low-cost microcontroller (PIC18F4620). For the PIC18F4620, it has 4096 bytes of data memory. The memory space is divided into 16 banks with each bank contains 256 bytes. For the implementation, the data memory is used to store the following variables:

① RSSI between $O-A$, $O-B$, $O-C$, $O-D$

② Denote the PSO population size as ps , then we have ps sets of PSO particles.

Each particle consists of 4 elements – p_0, α, x, y

③ ps sets of particle's velocities, each contains the velocity of p_0, α, x, y

④ ps sets of best positions that the particles have achieved so far

⑤ ps sets of cognition accelerations of particles, each contains the cognition acceleration of p_0, α, x, y

⑥ ps sets of social accelerations of particles, each contains social acceleration of p_0, α, x, y

⑦ Other variables

Items ② to ⑥ are arrays of size $ps \times 4$. Each real variable is declared as a floating type that occupies 4 bytes. Hence, each array occupies $ps \times 4 \times 4$ bytes. To allocate each array within one bank of size 256 bytes, $ps \leq 16$. RSSIs and other variables can be easily allocated into other banks.

Each array can also be allocated in more than one bank. To store one array into two banks, the following steps are required:

- Allocate one array into its own section using the `#pragma` directive

```
#pragma udata section3
```

```
float pos[32][4];
```

It is noticed that with two banks of data memory ps can be as large as 32.

- A new region that span across two banks must be created in the linker script

Linker script before modification:

```
DATABANK NAME=grp2 START=0x200 END=0x2FF
```

```
DATABANK NAME=grp3 START=0x300 END=0x3FF
```

Linker script after modification:

```
DATABANK NAME=big START=0x200 END=0x3FF
```

```
PROTECTED
```

- The array's section must be assigned to the new region

```
SECTION NAME=section3 RAM=big
```

Then, the array `pos[32][4]` will be allocated in region *big* by combining bank 2 and 3.

In 16 banks of the data memory, there are 2 access banks, 1 bank for stack. We will use 12 banks to allocate the 5 arrays above and one bank for other variables. Hence, the maximum value of ps is $12 \times 5 \times 256 / (5 \times 4 \times 4) \approx 38$. However, we choose $ps=32$ so that each array fits into 2 banks, which will be easier to create new regions in the linker script.

Overall, the memory requirement by the optimization program is 13k bytes out of 64k bytes program memory of the microcontroller.

To execute the optimization algorithm on the microcontroller, RSSIs are collected for

15 unknown locations at the experimental site (16mx35m) as shown in Fig. 5.2. The unknown locations are varied while the anchor nodes' locations are fixed. The time taken by the microcontroller to complete an optimization for each unknown node location has also been investigated. Fig. 5.10 plots the histogram of execution times at 17 different locations. From the figure, it is observed that the execution time varies from 1 to 4.5 minutes with average of 2 minutes 50 seconds. The execution time can be improved by using a microcontroller that has a faster clock rate. The PIC18F4620 (in PICDEM Z board) used in this experiment has a clock rate of 4MHz. There are some microcontrollers such as PIC18F46K20 or PIC18F66T50 which support clock rates up to 16MHz.

Fig. 5.10 Microcontroller execution time of PSO with log barrier

5.5 RSSI measurements

RSSI measurements have been performed experimentally to validate the RSSI model. The experiments were conducted around the Research Techno Plaza in the campus of

Nanyang Technological University. The area is hilly, and with light tree density. Fig. 5.11 shows the measured RSSIs at different distances. In the figure, the dots are the RSSI values, and the solid lines connecting the dots show that these RSSI values are measured at the same distance. From the figure, it is observed that the RSSI values fluctuate in a wide range. In the worst cases, the RSSI variation can be as large as 30dB as shown in Fig. 5.11 at 27m.

Fig. 5.11 Measured RSSIs at different distance

From the RSSIs obtained from the experiment, the corresponding maximum, mean and minimum RSSIs at every measured distance are calculated. A best fit curve is then plotted for each type of RSSIs as shown in Fig. 5.12. From these results, it is concluded that the best fit curve follows the RSSI-distance model as shown in (3-6). Fig. 5.13 plots all the three best fit curves together. Table 5-2 lists the estimate of path

loss parameters from the best fit curve. From Fig. 5.13, the error in RSSI measurement is determined by the deviation of the RSSI from the best fit curves. It is observed that the max RSSI has large error when measuring at long distance (40m to 70m); whereas the min RSSI has large error when measuring at short distance (10m to 30m). Large error in RSSI measurement will result in large localization error. Hence, instead of the max and min RSSI, the mean RSSI will be used for localization.

Fig. 5.12 Measured RSSI versus best fit RSSI
(a) Maximum RSSIs, (b) Mean RSSIs, (c) Minimum RSSIs

Fig. 5.13 Best fit curves for max, mean and min RSSIs

Table 5-2 Estimated path loss parameters at the RTP

	P_0	α	σ
Max RSSI	-44.4238	1.9365	3.5075
Mean RSSI	-45.79	2.0963	3.3824
Min RSS	-55.2536	1.7475	3.7716

Another experiment for the RSSI measurement has also been conducted at the Nanyang lake in the campus of the Nanyang Technological University. The results are plotted similarly in Figs. 5.14 – 5.16. From Fig. 5.14, it is observed that the measured RSSI also varies in a wide range. In the worst case, the measured RSSI deviates by 15dB at the same distance as shown in Fig. 5.14 at 31m.

Fig. 5.14 Measured RSSIs at different distances, conducted at the Nanyang lake

Fig. 5.15 Measured RSSI versus best fit RSSI
(a) Maximum RSSIs, (b) Mean RSSIs, (c) Minimum RSSIs

Fig. 5.16 Best fit curves for max, mean and min RSSIs

Table 5-3 Estimated path loss parameters at the Nanyang lake

	P_0	α	σ
Max RSSI	-37.8930	2.5331	3.6722
Mean RSSI	-41.2731	2.5599	3.8043
Min RSSI	-57.4025	1.7915	4.5314

Table 5-3 lists the estimated path loss parameters based on the best fit curves obtained from Fig. 5.16. From Fig. 5.16, it is also observed that the max RSSI has large measurement error at short (15m to 25m) and long (50m to 50m) distances; whereas the min RSSI has large measurement error at short distance (15m to 25m). At all the distances, the mean RSSI give the smallest measurement error. Large RSSI

measurement error will result in large localization error. The observation strengthens the conclusion in the previous measurement that the mean RSSI will be used in the localization algorithm. From Table 5-2 and 5-3, the path loss exponent α varies from 1.74 to 2.56; the range of p_0 is from -57 to -37. Hence, the boundaries of the path loss parameters used in the localization algorithm are determined as follows:

$$P_{0_{max}} = -20dB, P_{0_{min}} = -60dB, \alpha_{min} = 1.5, \alpha_{max} = 4 \quad (5-1)$$

5.6 Experimental results

For the experimental study, various measurements have been conducted in a car park located in the campus of Nanyang Technological University. Fig. 5.2 shows the experimental site. The functions of A , B , C , D and O have been explained in section 5.2. The module O is the unknown emitter node that could be placed anywhere inside the rectangle. Every known node is connected to a microcontroller board. For anchor nodes, the microcontroller is programmed so that the anchor nodes listen to the emitter node. The emitter node is connected to a laptop through USB interface. The laptop is programmed to command the emitter node to signal to every router. Each router, in turn, transmits a signal back so that the RSSIs between the nodes $O-A$, $O-B$, $O-C$, and $O-D$ can be measured. The computer executes the PSO algorithm in Matlab to determine the emitter's location. The execution time of the algorithm takes about 1.5s. However, the communication to transmit and receive signals with the respective router takes a relatively longer duration. Overall, the time taken to complete one cycle of optimization is about 4s.

Table 5-4 shows the averaged RSSIs based on 10 experimental readings at various

locations of the unknown node. The localization results calculated from the proposed algorithm are shown in Table 5-5.

Table 5-4 Averaged RSSI (from 10 readings) at various locations

Actual location		Averaged RSSI			
<i>x</i> (m)	<i>y</i> (m)	<i>OA</i>	<i>OB</i>	<i>OC</i>	<i>OD</i>
4	15	-69.92	-75.42	-73.49	-70.27
4	22	-71.99	-74.69	-73.71	-66.78
6	13	-66.03	-71.14	-74.66	-73.37
6	23	-73.37	-74.66	-71.14	-66.03
8	12	-66.35	-67.23	-74.91	-74.68
8	16	-71.90	-72.27	-70.28	-70.28
8	20	-70.28	-70.28	-72.27	-71.90
8	24	-74.68	-74.97	-67.23	-66.35
10	12	-70.25	-65.89	-74.08	-75.65
10	16	-73.93	-71.26	-70.26	-71.99
10	20	-71.99	-70.26	-71.26	-73.93
10	24	-75.65	-74.08	-65.89	-70.25
12	13	-71.56	-65.63	-74.08	-75.57
12	17	-71.46	-71.59	-75.42	-72.51
12	23	-75.57	-74.08	-65.63	-71.69

From Table 5-5, it is observed that the average localization error is 2.5 m. The largest error is 4 m, and the smallest error is 0.9 m. The average error in estimating the *x*

coordinates is 1.44 m. The largest error is 3.59 m and the smallest error is 0.25 m. For the y coordinates, the average, largest and smallest errors are 1.55, 3.65, 0.02 m respectively.

Table 5-5 Localization results (experiment)

Location		Result			
x (m)	y (m)	P_0	α	x (m)	y (m)
4	15	-29.96	3.22	3.30	18.62
4	22	-29.08	3.28	2.34	20.65
6	13	-31.42	3.10	4.06	12.98
6	23	-30.74	3.12	3.72	22.55
8	12	-31.40	3.07	7.51	11.19
8	16	-31.78	3.02	8.25	19.64
8	20	-31.78	3.02	8.25	16.35
8	24	-31.40	3.07	7.51	24.80
10	12	-31.12	3.13	11.86	12.40
10	16	-31.27	3.11	11.57	19.14
10	20	-31.27	3.11	11.57	16.85
10	24	-31.12	3.13	11.86	23.59
12	13	-30.79	3.18	13.29	13.03
12	17	-32.67	3.13	8.41	15.24
12	23	-30.79	3.18	13.79	22.96

The experimental error obtained from Table 5-5 is larger than the simulation error in

chapter 4. It is because, in chapter 4, the estimated location is averaged from 50 simulations. Hence, the localization error from simulation will be smaller than from experiment where the localization is done only once.

Table 5-6 Localization errors: PSO versus Multilateration

Actual location		Localization error (m)	
x (m)	y (m)	PSO	Multilateration
4	15	3.68	5.17
4	22	2.13	6.47
6	13	1.94	1.84
6	23	2.32	3.60
8	12	0.94	1.12
8	16	3.64	3.59
8	20	3.65	3.60
8	24	0.93	1.14
10	12	1.90	4.30
10	16	3.51	4.69
10	20	3.51	5.43
10	24	1.90	3.42
12	13	1.29	3.46
12	17	3.99	3.94
12	23	1.79	3.73

The localization method has also been benchmarked with the multilateration method.

Errors obtained from the two methods are calculated at every location of the unknown node and shown in Table 5-6. From the Table, it is concluded that the PSO achieves more accurate result than the multilateration method at almost every location. The improvement can be as high as 4.33m. On the average, the error of PSO's is smaller than multilateration by 1.22m.

5.7 Conclusion

In this chapter, an experimental system has been set up to validate the localization algorithm proposed in chapter 3. Every sensor node consists of a RF module and a microcontroller. A user-defined protocol has been developed to allow the coordinator (unknown node) to set up communication with a router (anchor node) and measure the RSSI.

In an area of 16m×35m, the average location error is 2.5m, the largest error is 4m and the smallest error is 0.9m. The experimental results also validate the results obtained from simulation. The proposed algorithm outperforms the multilateration method by 1.22m on the average. The algorithm is also compact enough to be implemented using a low-cost microcontroller.

Chapter 6

Localization using Particle Swarm

Optimization with log-barrier constraint for a general network

6.1 Introduction

In the previous discussions from chapters 3 to 5, the basic scenario with 4 neighboring anchor nodes has been investigated. A solution using the PSO with log-barrier approach has been proposed and validated in both simulation and experimental studies. In general, the unknown sensor node may communicate with more than 4 anchor nodes and receive their RSSI information. Hence, a more accurate estimation is possible. Though the localization algorithm proposed in chapter 3 can be easily extended to more than 4 anchor nodes, the PSO convergence is not guaranteed. With 4 variables (P_θ , α , x , y) to optimize the objective function, 4 sets of RSSI information are sufficient. With more RSSI information, the problem would become over-determined. Noticing that, we can always choose 4 among all the RSSI measurements to perform localization by using the proposed PSO with log-barrier approach. With n RSSI measurements from n anchor nodes, the possible combination is C_n^4 . Among the C_n^4 estimated locations, there are always solutions that are closer to the real unknown location. In this chapter, a selection scheme is proposed to choose a solution among all

the C_n^4 estimated location that is potentially closest to the real location.

6.2 Selection method

Consider an unknown node that has n neighbor anchor nodes with RSSI measurements denoted as $\xi_i, i = 1 \dots n, n > 4$. A subset of 4 distinct anchor nodes from the n anchor nodes is called a 4-combination. The number of 4-combinations from a set of n elements is $C_n^4 = \frac{n \times (n-1) \times (n-2) \times (n-3)}{4 \times 3 \times 2 \times 1}$. Using the PSO with log-barrier method as described in chapter 3, there are C_n^4 estimated locations.

Let i, j, k, l be 4 random anchor nodes. Moreover, let the estimated path loss parameters and unknown coordinates be denoted as $P_{0ijkl}, \alpha_{ijkl}, x_{ijkl}, y_{ijkl}$.

Substituting these values into (3-3), the estimated RSSI of node i is:

$$P_{r,i} = P_{0ijkl} - 10 \cdot \alpha_{ijkl} \cdot \log_{10} \left(\frac{\sqrt{(x_{ijkl} - x_i)^2 + (y_{ijkl} - y_i)^2}}{d_0} \right) - X_\sigma \quad (6-1)$$

where $P_{r,i}$ is the RSSI from anchor node i , and (x_i, y_i) are the Cartesian coordinates of anchor node i . Similarly, we have the following measured RSSIs for nodes j, k , and l :

$$P_{r,j} = P_{0ijkl} - 10 \cdot \alpha_{ijkl} \cdot \log_{10} \left(\frac{\sqrt{(x_{ijkl} - x_j)^2 + (y_{ijkl} - y_j)^2}}{d_0} \right) - X_\sigma \quad (6-2)$$

$$P_{r,k} = P_{0ijkl} - 10 \cdot \alpha_{ijkl} \cdot \log_{10} \left(\frac{\sqrt{(x_{ijkl} - x_k)^2 + (y_{ijkl} - y_k)^2}}{d_0} \right) - X_\sigma \quad (6-3)$$

$$P_{r,l} = P_{0ijkl} - 10 \cdot \alpha_{ijkl} \cdot \log_{10} \left(\frac{\sqrt{(x_{ijkl} - x_l)^2 + (y_{ijkl} - y_l)^2}}{d_0} \right) - X_\sigma \quad (6-4)$$

If the RSSI information from the anchor node i is measured for t times, the average of these RSSI values is

$$\bar{P}_{r,i} = \frac{\sum_1^t P_{r,i}}{t} = P'_0 - 10 \cdot \alpha_{ijkl} \cdot \log_{10} \left(\frac{\sqrt{(x_{ijkl} - x_l)^2 + (y_{ijkl} - y_l)^2}}{d_0} \right) \quad (6-5)$$

As mentioned in earlier chapter, the RSSI measurement has a Gaussian random distribution. Thus, we have

$$P_{r,i} = N(\bar{P}_{r,i}, \sigma) \quad (6-6)$$

Using 95% confidence interval, the following inequality must be satisfied

$$-1.96 \leq \frac{P_{r,i} - \bar{P}_{r,i}}{\sigma / \sqrt{t}} \leq 1.96 \quad (6-7)$$

$$-1.96 \leq \frac{P_{r,j} - \bar{P}_{r,j}}{\sigma / \sqrt{t}} \leq 1.96 \quad (6-8)$$

$$-1.96 \leq \frac{P_{r,k} - \bar{P}_{r,k}}{\sigma / \sqrt{t}} \leq 1.96 \quad (6-9)$$

$$-1.96 \leq \frac{P_{r,l} - \bar{P}_{r,l}}{\sigma / \sqrt{t}} \leq 1.96 \quad (6-10)$$

where t is the number of measurement; P and $\bar{P}$ denote the measured and mean RSSI.

To accept the result of $P_{0ijkl}, \alpha_{ijkl}, x_{ijkl}, y_{ijkl}$, the inequality (6-7) – (6-10) must be

satisfied. These inequalities constitute the first test in the selection scheme, which is the confidence interval test.

For an anchor node p , the measured RSSI is $P_{r,p}$. The mean RSSI can be calculated as follows:

$$\bar{P}_{r,p} = P_{0ijkl} - 10 \cdot \alpha_{ijkl} \cdot \log_{10} \left(\frac{\sqrt{(x_{ijkl} - x_p)^2 + (y_{ijkl} - y_p)^2}}{d_0} \right) \quad (6-11)$$

where (x_p, y_p) are the Cartesian coordinates of the anchor node p . Because RSSI measurement has a Gaussian distribution, we have

$$P_{r,p} = N(\bar{P}_{r,p}, \sigma) \quad (6-12)$$

The probability of a given measured $P_{r,p}$ can be obtained as:

$$P(P_{r,p}) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left(-\frac{(\bar{P}_{r,p} - P_{r,p})^2}{2\sigma^2} \right) \quad (6-13)$$

From the estimation theory, the likelihood function of $P_{0ijkl}, \alpha_{ijkl}, x_{ijkl}, y_{ijkl}$ based on the measured $P_{r,p}$ is equal to the probability of $P_{r,p}$. (6-13) becomes

$$L(P_{0ijkl}, \alpha_{ijkl}, x_{ijkl}, y_{ijkl} | P_{r,p}) = \frac{1}{\sigma\sqrt{2\pi}} \exp \left(-\frac{(\bar{P}_{r,p} - P_{r,p})^2}{2\sigma^2} \right) \quad (6-14)$$

The overall likelihood function is

$$L(P_{0ijkl}, \alpha_{ijkl}, x_{ijkl}, y_{ijkl} | P_{r,p}, p = 1 \dots n) = \prod_{p=1}^n \frac{1}{2\sqrt{\pi}\sigma} \exp \left(-\frac{(P_{r,p} - \bar{P}_{r,p})^2}{2\sigma^2} \right) =$$

$$\left(\frac{1}{2\sqrt{\pi}\sigma}\right)^n \exp\left(-\frac{\sum_{p=1}^n (\bar{P}_{r,p} - P_{r,p})^2}{2\sigma^2}\right) \quad (6-15)$$

Fig. 6.1 Selection scheme's flow chart

From (6-15), the likelihood function L is maximized when $\sum_{p=1}^n (P_{r,p} - \bar{P}_{r,p})^2$ is minimized. This constitutes the second test in the selection scheme, which is called the

maximum likelihood test. Among all the results that pass the confidence interval test, the smallest $\sum_{p=1}^n (P_{r,p} - \bar{P}_{r,p})^2$ is selected as the final estimated result.

Fig. 6.1 shows the flow chart of the selection method.

- Firstly, choose 4 neighbor anchor nodes i, j, k , and l from the set of N neighbor anchor nodes.
- By using the algorithm developed in chapter 3, $P_{0ijkl}, \alpha_{ijkl}, x_{ijkl}, y_{ijkl}$ are estimated.
- Calculate the average RSSIs $\bar{P}_{r,i}, \bar{P}_{r,j}, \bar{P}_{r,k}, \bar{P}_{r,l}$ by using (6-5).
- Check the confidence interval test (6-7) – (6-10).
- If the confidence interval test is passed, calculate $\bar{P}_{r,p}$ for other anchor node p by using (6-11).
- Calculate $\sum_{p=1}^n (P_{r,p} - \bar{P}_{r,p})^2$. Then choose another 4 neighbor anchor nodes.
- If the confidence interval test is not passed, then choose another 4 neighbor anchor nodes.
- If there is no more 4-combination, find the 4 anchor nodes i, j, k, l with the smallest $\sum_{p=1}^n (P_{r,p} - \bar{P}_{r,p})^2$. The corresponding $P_{0ijkl}, \alpha_{ijkl}, x_{ijkl}, y_{ijkl}$ is the final estimated P_0, α, x, y .

6.3 Case study

Fig. 6.2 shows an example of a sensor node deployment. The dimension of the deployment area is 1000×1000. In the figure, filled circle denotes anchor node and unfilled circle denotes unknown node. A solid line connecting between an unknown and an anchor nodes implies that the two nodes are able to communicate with each other. The communication range is 300 units. For this study, we only deploy 8 unknown nodes among the anchor nodes to illustrate the execution of the selection scheme. As shown in the figure, the unknown node can communicate with 7 anchor nodes in case 1. However, the unknown node can only communicate with 4 anchor nodes in case 8. The selection scheme developed in this chapter is applied to solve the first 7 different localization problems, while case 8 will not be studied.

Fig. 6.2 Eight localization scenarios

Fig. 6.3 plots the estimated locations obtained from the selection method for the first seven cases. The black star (★) denotes the actual location, the blue filled circle (●) denotes the estimated location after selection, the red asterisk (*) denotes some other estimated locations that are not selected. The unfilled blue circle denotes the anchor node. For example, in case 1, there are 7 neighbor anchor nodes, there should be $C_7^4 = 35$ estimated locations. After the confidence interval test, there are three locations (2 red asterisks and 1 blue filled circle), in which only one (blue filled circle) is chosen.

From the result, it is observed that the selection method performs well. The determined solution is close to the actual location. In cases 6 and 7, there are several locations that are obviously unacceptable. However, these locations are rejected after the selection process. To assess the performance of the localization approach, large-scale simulation is conducted in the next section.

Fig. 6.3 Estimated location from Fig. 6.2

6.4 Large scale simulation study

In the simulation, the number of anchor nodes, the path loss's parameters and the sensor nodes' locations are changed to test the algorithm's performance. The localization approach is assessed mainly based on the localization accuracy. The results are also benchmarked with the Sequence Based Localization (SBL) approach [21]. While there are many localization techniques, it is fairer to compare with a similar technique. Similar to the developed algorithm, the SBL assumes that the unknown node is localized by using a few anchor nodes. Its localization is also based on the received signal strength. There is also no restriction on the number of anchor nodes. Hence, simulation or experimental study can be conducted using similar environment for benchmarking.

First, we conduct simulations using the same scenarios and condition as in [21] to use the SBL's localization result. The sensor field is a 100×100 square. It is assumed that every sensor is able to communicate with each other inside the field. There are more than four, such as 5, 6, 7, etc anchor nodes randomly placed inside the square for 100 different locations. At each time, the unknown node is randomly assigned at a different location. Error is being average from 10,000 simulation trials.

For the simulation, the path loss exponent is set as 4. The number of anchor node varies from 5 to 10. Results are presented in Table 6-1 and Fig. 6.4.

Table 6-1 Average errors versus number of anchor nodes

No. of anchor nodes	5	6	7	8	9	10
Mean	16.6280	14.5782	11.6671	10.6617	8.2073	7.7914
Max	47.1072	30.3874	28.3726	26.0560	22.9475	19.4513
Min	5.7969	6.1797	4.4953	5.0681	3.0579	2.7074

Fig. 6.4 Average localization error versus number of anchor nodes

Table 6-1 shows that the algorithm can achieve an average error as small as 7.8 meters when the number of anchor nodes is 10. Fig. 6.4 shows that the average error reduces when the number of anchor nodes increases. The maximum and minimum errors also

exhibit the same trend.

Fig. 6.5 to Fig. 6.8 plots the location error with varying RSSI standard deviation. The result is also compared with the SBL under the same scenarios. In each scenario with a particular N and α where N is the number of neighbor anchor nodes, the errors over all the RSSI standard deviations are summed up, the average of which is shown in Table 6-2. From the figures, it is observed that the localization error increases when the RSSI standard deviation increases. The rate of increase of the SBL and the PSO-log barrier methods are about the same. From Table 6-2, the localization error reduces with increasing the path loss exponent α . It means that the localization method performs better in an environment where the received signal strength reduces faster with increasing separating distances. In free space, the localization method performs worst. The figures and the table also show that the PSO-log barrier method outperforms the SBL method in all the simulation scenarios. The PSO-log barrier achieves an average error 8 meters smaller than the average error the SBL method.

Fig. 6.5 PSO with log-barrier versus SBL (7 neighbors, $\alpha = 4$)

Fig. 6.6 PSO with log-barrier versus SBL (10 neighbors, $\alpha = 2$)

Fig. 6.7 PSO with log-barrier versus SBL (10 neighbors, $\alpha = 4$)

Fig. 6.8 PSO with log-barrier versus SBL (10 neighbors, $\alpha = 6$)

Table 6-2 Average error over a range of RSSI standard deviation

	N=7, $\alpha=4$	N=10, $\alpha=2$	N=10, $\alpha=4$	N=10, $\alpha=6$
SBL	22.1	26.8	17.8	15.4
PSO-log barrier	13.9	18.3	9.9	7.2

6.5 Experiment

The experiment has been carried out in a Sport Centre. Fig. 6.9 shows the experimental site.

Fig. 6.9 Experimental site for the general network

As shown in Fig. 6.9, there are 12 anchor nodes being deployed and distributed evenly. There are marked as yellow squares. A local 2D Cartesian coordinate system

is defined, in which the anchor node A is designated as origin. The coordinates of the anchor nodes are: A (0, 0), B (30, 0), C (60, 0), D (100, 0), E (127.57, 15.915), F (127.57, 47.745), G (100, 63.66), H (60, 63.66), I (30, 63.66), J (0, 63.66), K (-27.57, 47.745), and L (-27.57, 15.915). The unknown nodes are placed at the locations (80, 10), (40, 40), (80, 40), (50, 50), (0, 30) and (-10, 50) and are marked as blue circle in Fig. 6.9.

Table 6-3 Unknown nodes versus number of neighbor anchor nodes

Node	(80, 10)	(40, 40)	(80, 40)	(50, 50)	(0, 30)	(-10, 50)
Number of neighbors	7	10	8	8	8	7

Table 6-3 shows the number of anchor nodes that the unknown node at each location can communicate with. For each unknown location, the selection scheme with PSO – log barrier is executed. Fig. 6.10 shows the results of the estimation based on the proposed approach. The black star ★ shown in Fig. 6.10 is the real location, the blue circle ● is the final estimated location, and the red asterisk * is the other estimated location. For the case of 7, 8, 10 neighbor anchor nodes, there are 35, 70, 210 estimated locations. Thus only a few estimated locations are plotted. From the figure, it is observed that the selection scheme is successful in eliminating the poor estimates. For some cases, the best estimated location is not selected due to measurement noise.

Fig. 6.10 Estimated location for the experiment

In the experiment, the PSO-log barrier scheme is also compared with multilateration approach, a popular localization method. Multilateration has been presented in chapter 4. Fig. 6.11 shows the flow chart of multilateration used for the experiment. Table 6-4 shows the results comparing multilateration and the selection scheme with PSO - log barrier approach. From the table, it is observed that the proposed selection scheme outperforms the multilateration approach at every location of the unknown node.

Fig. 6.11 Flow chart of multilateration approach for the experiment

Table 6-4 Experiment: Selection scheme versus multilateration

Real location		Proposed PSO-log barrier scheme			Multilateration		
x	y	x	y	Error	x	y	Error
80	10	78.3642	7.6322	2.8779	76.059	17.9964	8.9148
40	40	39.84	35.73	4.273	33.9157	29.8179	11.8614
80	40	76.7813	32.2753	8.3684	74.4575	33.1685	8.7971
50	50	54.8551	51.5664	5.1015	48.6151	41.0703	9.0364
0	30	5.72	31.996	6.0583	9.8835	31.2986	9.9684
-10	50	-21.8992	54.0247	12.5614	8.9037	45.2259	19.4972

6.6 Conclusion

In this chapter, a selection scheme has been proposed to select the best result among the estimated locations obtained from the PSO with log-barrier optimization. From the extensive simulation, it is observed that the algorithm achieves high accuracy, with an average error of 5.9 m with a RSSI standard deviation of 4 dB in an area of 100×100. In contrast, the SBL method can only achieves an average error of 13 m. The average errors obtained from the developed algorithm are at least 6 m smaller than the SBL. Under the same environment condition, the more neighboring anchor node, the smaller the localization error. From the simulation, the localization error reduces from 16.6 m to 7.8 m when the number of anchor nodes increases from 5 to 10. The localization method performs better with larger path loss exponent α . Experiment has also been carried out. It is observed that the selection scheme is successful in

eliminating poor estimates and outperforms the multilateration approach.

Chapter 7

Conclusion and future work

Wireless sensor networks have attracted significant attentions of research in recent years because of its wide application potential. One problem arises in WSN application is localization. In this thesis, a new localization algorithm based on PSO with log-barrier function has been developed. Besides its simplicity, the proposed localization algorithm also estimates the unknown locations more accurately than some advanced localization methods such as the Sequenced Based Localization.

The developed localization method is RF-based, in which the log-normal path loss model that describes the received power – distance relationship is used. In conventional RF-based localization methods, the path loss parameters are often determined based on an extensive RSSI measurement conducted offline. For the proposed method, the path loss parameters are estimated in real time, together with the unknown location. The problem is modeled as a constrained nonlinear optimization problem. As the conventional PSO is not able to solve the problem, a new PSO with log-barrier function which tailors to this problem has been developed. Moreover, the proposed algorithm is fast enough to be executed in real time.

The proposed algorithm has been validated by simulation and experimental results. In the simulation, the localization errors are obtained for different unknown node's positions and different RSSI standard deviations. With an average of 25% error in the

measured RSSI, the average localization error is 14.25 units (11.8% of the deployment area's dimension). Average errors in estimating P_0 and α are 8.6% and 4.6% respectively. The study shows that the proposed localization algorithm performs reliably with any unknown node's location and RSSI standard deviation.

The tuning of PSO parameters and the barrier parameter have also been discussed. After considering the tradeoff between the execution time and the localization error, the population size, the cognitive and the social components of the acceleration constants are chosen to be 40, 2 and 2.5 respectively. The most suitable barrier parameter is 5.

The performance of the localization algorithm has also been benchmarked with two other well known algorithms namely the multilateration and the Sequence Based Localization. With a 4dB error in RSSI measurement, the errors obtained for the proposed algorithm, multilateration and the SBL are 19%, 32% and 27% respectively.

The proposed algorithm has also been extended to a general case with more than 4 neighboring anchor nodes. A selection scheme is developed to choose the most suitable result. From the extensive simulation, it is observed that the proposed algorithm achieves high accuracy with an average error of 5.9 m at a RSSI standard deviation of 4 dB in an area of 100m×100m. When compared to the SBL algorithm, the average errors obtained from the proposed algorithm are at least 6 m smaller than the SBL.

In conclusion, a new localization algorithm for a WSN has been proposed. The algorithm performs well for a variety of networks with different topologies and

densities. For the future work, the algorithm can be extended to localize mobile sensor networks. Networks with both stationary and mobile sensor nodes can also be considered.

List of publication

1. H.A. Nguyen, H. Guo, K.S. Low, "Real Time Estimation of Sensor Node's Position Using Particle Swarm Optimization with Log Barrier Constraint," *IEEE Trans. Instrumentation and Measurements*, in press.
2. H. Guo, K.S. Low, H.A. Nguyen, "Optimizing the Localization of a Wireless Sensor Network in Real Time Based on a Low Cost Microcontroller," *IEEE Trans. Ind. Electron.*, in press, web version published.
3. H.A. Nguyen, K.S. Low, H. Guo, "Real Time Determination of Sensor Node Location in a Wireless Sensor Network using Particle Swarm Optimization," in *Proc. Int. Conf. Evolutionary Computing*, Prague, Czech Republic, 23 – 25 Mar. 2009, pp. 140 – 145.
4. K.S. Low, H.A. Nguyen, H. Guo, "Optimization of Sensor Node Locations in a Wireless Sensor Network," in *Proc. 4th IEEE Int. Conf. Natural Computation*, Jinan, China, 18 – 20 Oct. 2008, pp. 286 – 290.
5. K.S. Low, H.A. Nguyen, H. Guo, "A Particle Swarm Optimization Approach for the Localization of a Wireless Sensor Network," in *Proc. IEEE Int. Symp. Ind. Electron.*, Cambridge, UK, 30 Jun. – 2 Jul. 2008, pp. 1820 – 1825.
6. H. Guo, K.S. Low, H.A. Nguyen, and M.J. Er., "Localization in a Sparse Sensor Network Using Pedometer and Communication Ranging Measurements," in *Proc. 33rd IEEE Annu. Conf. Control and Instrumentation*, Taipei, Taiwan, 7 – 10 Nov. 2007, pp. 2627 – 2632.

Reference

- [1] L. S. Xiao, X. H. Peng, Z. X. Wang, B. Xu, and P. Z. Hong, "Research on traffic monitoring network and its traffic flow forecast and congestion control model based on wireless sensor networks," in *Proc. ICMTMA Conf.*, 2009, pp. 142 – 147.
- [2] K. Khakpour, M. H. Shenassa, "Industrial control using wireless sensor networks," in *Proc. ICTTA Conf.*, 7 – 11 Apr. 2008, pp. 1 – 5.
- [3] S. M. Diamond, M. G. Ceruti, "Application of wireless sensor networks to military information integration," in *Proc. 5th IEEE Int. Conf. Industrial Informatics*, 23 – 27 Jun. 2007, pp. 317 – 322.
- [4] Y. Xu, Z. Wang, F. S. Makedon, J. D. Pearlman, "BP-neural network based-characterization of electrographic magneto hydrodynamic signals in MR," in *Proc. 26th IEEE Annu. Int. Conf. Engineering in Medicine and Biology Soc.*, San Francisco, Sep. 2004, pp. 431 – 433.
- [5] G. J. Pendock, L. Evans, G. Coulson, "Wireless sensor module for habitat monitoring," in *Proc. 3rd Int. Conf. Intell. Sensors, Sensor Netw. Inform.*, 3 – 6 Dec. 2007, pp. 699 – 702.
- [6] D. W. Welch, M. C. Melnychuk, R. O'Dor, "Application of the Pacific Ocean shelf tracking system (POST): A permanent continental-scale acoustic tracking array for fisheries research & ocean observation," in *Proc. OCEANS Conf.*, 18 – 21 Sep. 2006, pp. 1 – 5.
- [7] C. H. Lee, L. B. M. S. Lim, "The optimum configuration of car parking guide system based on wireless sensor networks," in *Proc. IEEE Int. Symp. Ind. Electron.*, 5 – 8 Jul. 2009, pp. 1199 – 1202.
- [8] J. Chinrungrueng, U. Sunantachaikul, S. Triamlumlerd, "Smart Parking: An application of optical Wireless sensor networks," in *Proc. Int. Symp. Applications and the Internet Workshops*, 15 – 19 Jan. 2007, pp. 66 – 66.
- [9] R. Stoleru, T. He, J. A. Stankovic, D. Luebke, "A high-accuracy, low-cost localization system for wireless sensor networks," in *Proc. 3rd ACM Int. Conf. Embedded Netw. Sensor Syst.*, USA, 2005, pp. 13 – 26.
- [10] N. B. Priyantha, A. Chakraborty, H. Balakrishnan, "The Cricket location-support system," in *Proc. 6th ACM Annu. Int. Conf. Mobile Comput. Netw.*, Boston, Aug 2000, pp. 32 – 43.
- [11] V. Vivekanandan, Vincent W. S. Wong, "Concentric anchor beacon localization algorithm for Wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 2733 – 2744, Sep. 2007.
- [12] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, S. Teller, "Mobile-assisted localization in wireless sensor networks," in *Proc. 24th Annu. Joint. Conf. Comput. Comm. Societies*, vol. 1, 13 – 17 Mar. 2005, pp. 172 – 183.
- [13] H. L. Ren and Max Q. H. Meng, "Power adaptive localization algorithm for Wireless sensor networks using Particle Filter," *IEEE Trans. Veh. Technol.*, vol. 58, no. 5, pp. 2498 – 2508, Jun. 2009.
- [14] P. Bahl, V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proc. INFOCOM Conf.*, Mar. 2000, pp. 775 – 784.
- [15] H. S. Ahn, W. Yu, "Environmental-adaptive RSSI-based indoor localization," *IEEE Trans. Automation Science and Engineering*, vol. 6, no. 4, pp. 626 – 633, Oct. 2009.
- [16] W. D. Wang, Q. X. Zhu, "RSS-based Monte Carlo localization for mobile sensor networks," *Commun. IET*, vol. 2, no. 5, pp. 673 – 681, May 2008.
- [17] M. Barralet, H. Xu and D. Sharma, "Effect of antenna polarization on RSSI based location identification," in *Proc. 11th Int. Conf. Advanced Comm. Technol.*, 2009, pp. 260 – 265.
- [18] K. S. Low, H. A. Nguyen, H. Guo, "Optimization of sensor node locations in a wireless sensor network," in *Proc. 4th IEEE Int. Conf. Natural Computation*, 18-20 Oct. 2008, pp. 286 – 290.
- [19] P. J. Chuang, C. P. Wu, "An effective PSO-based node localization scheme for wireless sensor networks," in *Proc. 9th Int. Conf. Parallel and Distributed Computing, Appl. and Technol.*, 2008, pp. 187-194.

- [20] L. J. Gopakumar, "Localization in wireless sensor networks using particle swarm optimization," in *Proc. IET Int. Conf. Wirel., Mobile and Multimedia Netw.*, 2008, pp. 227–230.
- [21] K. Yedavalli, and B. Krishnamachari, "Sequenced-based localization in Wireless sensor network," *IEEE Trans. Mobile Computing*, vol. 7, no. 1, pp. 81 – 94, Jan. 2008.
- [22] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663 – 1678, Dec. 2006.
- [23] B. M. Sadler, "Fundamentals of energy-constrained sensor network systems," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 20, no. 8, pp. 17 – 35, Aug. 2005.
- [24] Y. Zou, K. Chakrabarty, "Energy-aware target localization in Wireless sensor networks," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Comm.*, 2003, pp. 60 – 67.
- [25] A. Wang, A. Chandrakasan, "Energy efficient system partitioning for distributed wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2001, vol. 2, pp. 905 – 908.
- [26] S. Y. Park, H. S. Ahn, W. Yu, "Adaptive path-loss model-based indoor localization," in *Proc. Int. Conf. Consumer Electron.*, Jan. 2008, pp. 1 – 2.
- [27] G. Y. Jin, X. Y. Lu, M. S. Park, "An indoor localization mechanism using active RFID tag," in *Proc. IEEE Int. Conf. Sensor Netw., Ubiquitous and Trustworthy Comput.*, Jun. 2006, pp. 40 – 43.
- [28] L. M. Ni, Y. H. Liu, Y. C. Lau, A. P. Patil, "LANDMARC: indoor location sensing using active RFID," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2003, pp. 407 – 415.
- [29] A. P. Peng, X. S. Guo, W. Cai, H. B. Li, "A distributed localization scheme for Wireless sensor networks based on Bounding box algorithm," in *Proc. 9th Int. Conf. Electron. Measurement and Instrumentations*, 2009, pp. 984 – 988.
- [30] S. Datta, C. Klinowski, M. Rudafshani, S. Khaleque, "Distributed localization in static and mobile sensor networks," in *Proc. IEEE Conf. Wirel., Mobile Comput., Netw., Commun.*, Jun. 2006, pp. 69 – 76.
- [31] Y. Wang, X. D. Wang, D. M. Wang, D. P. Dgrawal, "Range-free localization using expected hop progress in Wireless sensor networks," *IEEE Trans. Parallel and Distribution Syst.*, vol. 20, no. 10, pp. 1540 – 1552, Oct. 2009.
- [32] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, A. A. F. Loureiro, "DV-Loc: a scalable localization protocol using Voronoi diagrams for wireless sensor networks," *IEEE Trans. Wirel. Commun.*, vol. 6, no. 2, pp. 50 – 55, 2009.
- [33] J. P. Sheu, W. K. Hu, J. C. Lin, "Distributed localization scheme for mobile sensor networks," *IEEE Trans. Mobile Computing*, vol. 9, no. 4, pp. 516 – 526, Apr. 2010.
- [34] A. Baggio, K. Langendoen, "Monte-Carlo localization for mobile wireless sensor networks," *J. Ad Hoc Netw.*, vol. 6, no. 5, pp. 718 – 733, 2008.
- [35] L. X. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. 10th Int. Conf. Mobile Computing and Netw.*, Sep. – 1 Oct. 2004, pp. 45 – 57.
- [36] K. W. Cheung, J. H. M. Sau, R. D. Murch, "A new empirical model for indoor propagation prediction," *IEEE Trans. Veh. Technol.*, vol. 47, no. 3, pp. 996 – 1001, Aug. 1998.
- [37] F. Thiesse, E. Fleisch, M. Dierkes, "LotTrack: RFID-based process control in the semiconductor industry," *IEEE Pervasive Comput.*, vol. 5, no. 1, pp. 47 – 53, Jan. – Mar. 2006.
- [38] C. C. Hsiao, P. Huang, "Two practical considerations of beacon deployment for ultrasound-based indoor localization systems," in *Proc. IEEE Int. Conf. Sensor Netw., Ubiquitous and Trustworthy Comput.*, 11 – 13 Jun. 2008, pp. 306 – 311.
- [39] B. H. Kim, J. S. Choi, S. I. Ko, M. Park, "Improved active beacon system using multi-modulation of ultrasonic sensors for indoor localization," in *Proc. SICE – ICASE Int. Joint. Conf.*, Oct. 2006, pp. 1366 – 1371.
- [40] B. H. Kim, J. S. Choi, "Active beacon system with the fast processing architecture for indoor localization," in *Proc. IEEE Conf. Emerging Technologies and Factory Automation*, Sep. 2007, pp. 892 – 895.

- [41] K. Yu, Y. J. Guo, M. Hedley, "TOA-based distributed localization with unknown internal delays and clock frequency offsets in wireless sensor networks," *IET Signal Processing*, vol. 3, no. 2, pp. 106 – 118, 2009.
- [42] M. Sun, K. C. Ho, "Successive and asymptotically efficient localization of sensor nodes in closed-form," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4522 – 4537, Nov. 2009.
- [43] H. Guo, K. S. Low, H. A. Nguyen, "Optimizing the localization of a wireless sensor network in real time based on a low cost microcontroller," *IEEE Trans. Ind. Electron.*, in press.
- [44] D. Niculescu, N. Badri, "Ad hoc positioning system (APS) using AOA," in *Proc. INFOCOM Conf.*, 3 Mar. – 3 Apr. 2003, vol. 3, pp. 1734 – 1743.
- [45] K. C. Ho, Luis M. Vicente, "Sensor allocation for source localization with decoupled range and bearing estimation," *IEEE Trans. Signal Processing*, vol. 56, no. 12, pp. 5773 – 5789, Dec. 2008.
- [46] M. Gavish, Anthony J. Weiss, "Array Geometry for Ambiguity Resolution in Direction Finding," *IEEE Trans. Antennas and Propagations*, vol. 44, no. 6, pp. 889 – 895, Jun. 1996.
- [47] M. Boushaba, A. Hafid, A. Benslimane, "High accuracy localization method using AOA in sensor networks," in *Int. J. Computer and Telecommun. Netw.*, vol. 53, no. 18, pp. 3076 – 3088, Dec. 2009.
- [48] A. Goldsmith, "Wireless communications," Cambridge university press, 2005, pp. 42 – 47.
- [49] R. P. Pires, L. F. Wanner, A. A. Frohlich, "An efficient calibration method for RSSI-based location algorithms," in *Proc. 6th IEEE Int. Conf. Industrial Informatics*, Jul. 2008, pp. 1183 – 1188.
- [50] J. G. Castano, M. Svensson, and M. Ekstrom, "Local positioning for wireless sensors based on Bluetooth," in *Proc. IEEE Conf. Radio and Wirel.*, Sep. 2004, pp. 195 – 198.
- [51] E. Cassano, F. Florio, F. De Rango and S. Marano, "A performance comparison between ROC-RSSI and trilateration localization techniques for WPAN sensor networks in a real outdoor testbed," in *Proc. Wirel. Telecommun. Symp.*, 2009, pp. 344 – 351.
- [52] R. Zemek, D. Anzai, S. Hara, K. Yanagihara, K. I. Kitayama, "RSSI-based localization without prior knowledge of channel model parameters," *Int. J. Wirel. Inform. Netw.*, pp. 128 – 136, 2008.
- [53] Y. C. Shih, Y. Y. Hsu, C. H. Chen, C. C. Tseng, E. Sha, "Adaptive attenuation factor model for localization in wireless sensor networks," *Int. J. Pervasive Computing and Commun.*, vol. 4, no. 3, pp. 257 – 267, 2008.
- [54] L. S. Jayashree, S. Arumugam, M. Anusha, A. B. Hariny, "On the Accuracy of Centroid based Multilateration Procedure for Location Discovery in Wireless Sensor Networks," in *Proc. IEEE Int. Conf. Wirel. and Optical Commun. Netw.*, 2006, pp. -6.
- [55] P. J. Chuang, C. P. Wu, "An Effective PSO-based Node Localization Scheme for Wireless Sensor Networks," in *Proc. 9th IEEE Int. Conf. Parallel and Distributed Computing, Appl. and Technol.*, Dec. 2008, pp. 187 – 194.
- [56] Y. Zheng, Y. Liu, "Quality of trilateration: Confidence-based iterative localization," in *IEEE 28th Int. Conf. Distributed Comput. Syst.*, Jun. 2008, pp. 446 – 453.
- [57] W. Wang, B. H. Soong, "A distributed heuristics of localization in wireless sensor network," in *Proc. IEEE Int. Conf. Wirel. Commun., Netw. and Mobile Computing*, Sep. 2006, pp. 1 – 4.
- [58] X. R. Li, "Collaborative localization with received-signal strength in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3807 – 3811, Nov. 2007.
- [59] S. Vijayakumaran, Y. Levinbook, T.F. Wong, "Maximum likelihood localization of a diffusive point source using binary observations," *IEEE Trans. Signal Processing*, vol. 55, no. 2, pp. 665 – 676, Feb. 2007.
- [60] M. P. Michaelides, C. G. Panayiotou, "Fault tolerant maximum likelihood event localization in sensor networks using binary data," *IEEE Signal Processing Letters*, vol. 26, no. 5, pp. 406 – 409, May 2009.
- [61] S. I. Ko, J. S. Choi, B. H. Kim, "Performance enhancement of indoor mobile localization system using Unscented Kalman filter," in *Proc. SICE-ICASE Int. Joint Conf.*, Oct. 2006, pp. 1355 – 1360.
- [62] P. N. Parthirana, N. Bulusu, A. V. Savkin, S. Jha, "Node localization using mobile robots in delay-tolerant sensor networks," *IEEE Trans. Mobile Computing*, vol. 4, no. 3, pp. 285 – 296, May – Jun. 2005.

- [63] R. Jirawimut, P. Ptasiński, V. Garaj, F. Ceceilja, W. Balachandran, "A method for dead reckoning parameter correction in pedestrian navigation system," *IEEE Trans. Instrumentation and Measurement*, vol. 52, no., pp. 209 – 215, Feb. 2003.
- [64] F. Dellaert, D. Fox, W. Burgard, S. Thrun, "Monte Carlo localization for mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, 1999, pp. 1322-1328.
- [65] L. Klingbeil, T. Wark, "Demonstration of a wireless sensor network for real-time indoor localization and motion monitoring," in *Proc. 7th Int. Conf. Information Processing in Sensor Netw.*, Apr. 2008, pp. 543 – 544.
- [66] A. Doucet, N. De Freitas, N. Gordon, "Sequential Monte Carlo method in practice," Springer, 2001.
- [67] A. Baggio and K. Langendoen, "Monte-Carlo Localization for Mobile Wireless Sensor Networks," in *Proc. 2nd Int. Conf. Mobile Ad-hoc and Sensor Netw.*, Hong Kong, China, Dec. 2006, pp. 317 – 328.
- [68] F. K. W. Chan, H. C. So, "Efficient weighted multidimensional scaling for wireless sensor network localization," *IEEE Trans. Signal Processing*, vol. 57, no. 11, pp. 4548 – 4553, Nov. 2009.
- [69] X. Ji and H. Zha, "Sensor positioning in wireless ad hoc networks using Multidimensional Scaling," in *Proc. IEEE INFOCOM*, 2004, pp. 2652 – 2661.
- [70] Y. Shang, W. Rumi, Y. Zhang, M. Fromherz, "Localization from connectivity in a network," *IEEE Trans. Parallel and Distributed Syst.*, vol. 15, no. 11, pp. 961 – 974, Nov. 2004.
- [71] T. M. Chan, S. Kwong, K. F. Man, K. S. Tang, "Hard handoff minimization using genetic algorithms," *Signal Processing*, vol. 83, no. 8, pp. 1047 – 1058, Jan. 2001.
- [72] S. H. Yeung, K. F. Man, W. S. Chan, "Optimized design of an ISM band antenna using a jumping genes methodology," *IET Microwaves, Antennas and Propagation*, vol. 2, no. 3, pp. 259 – 267, 2008.
- [73] S. X. D. Tan, C. J. R. Shi and J. C. Lee, "Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programming," *IEEE Trans. Computer-aided Design of Integrated Circuits and Syst.*, vol. 22, no. 12, pp. 1678-1684, Dec. 2003.
- [74] T. Wimalajeewa, and S. K. Jayaweera, "Optimal power scheduling for correlated data fusion in Wireless sensor networks via constrained PSO," *IEEE Trans. Wireless Comm.*, vol. 7, no. 9, pp. 3608-3618, Sep. 2008.
- [75] Y. Shi, Y. T. Hou, H. D. Sherali, "Cross-layer optimization for data rate utility problem in UWB-based adhoc networks," *IEEE Trans. Mobile Computing*, vol. 7, no. 6, pp. 764 – 777, Jun. 2008.
- [76] T. Jia, R. M. Buehrer, "On the optimal performance of collaborative position location," *IEEE Trans. Wireless Comm.*, vol. 9, no. 1, pp. 374 – 383, Jan. 2010.
- [77] J. Nocedal, S. J. Wright, *Numerical Optimization*, Springer, 2006, pp. 501-512
- [78] J. Semeter, M. Mendillo, "A nonlinear optimization technique for ground-based atmospheric emission tomography," *IEEE Trans. Geoscience and Remote Sensing*, vol. 35, no. 5, pp. 1102 – 1116, Sep. 1997.
- [79] R. Roy, A. B. Thompson, A. Godavarty, E. M. Sevick-Muraca, "Tomographic fluorescence imaging in tissue phantoms: A novel reconstruction algorithm and image recovery," *IEEE Trans. Medical Imaging*, vol. 24, no. 2, pp. 137 – 154, Feb. 2005.
- [80] R. Polyak, "Modified barrier function (theory and methods)," in *Math. Programming*, vol. 54, pp. 177 – 222, 1992.
- [81] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proc. Congr. Evolutionary Computation*, 1999, pp. 1945 – 1950.
- [82] K. S. Low and T. S. Wong, "A Mutltiobjective Generic Algorithm for Optimizing the Performance of Hard Disk Drive Motion Control System," *IEEE Trans. Ind. Electron.* vol. 54, no. 3, pp. 1716 – 1725, Jun. 2007.
- [83] T. Wimalajeewa, and S. K. Jayaweera, "Optimal power scheduling for correlated data fusion in Wireless sensor networks via constrained PSO," *IEEE Trans. Wirel. Commun.*, vol. 7, no. 9, pp. 3608 – 3618, Sep. 2008.
- [84] K. S. Tang, K. T. Ko, K. F. Man, and S. Wong, "Topology design and bandwidth allocation of embedded ATM networks using genetic algorithms," *IEEE Commun. Letters*, vol. 2, no.6, pp. 171 – 173, Jun. 1998.

- [85] T. M. Chan, K. F. Man, K. S. Tang, and S. Kwong, "A jumping-genes paradigm for optimizing factory WLAN network," *IEEE Trans. Ind. Informatics*, vol. 3, no. 1, pp. 33 – 43, Feb. 2007.
- [86] S. H. Yeung, K. F. Man, K. M. Luk, C. H. Chan, "A trapeizform U-slot folded patch feed antenna design optimized with jumping genes evolutionary algorithms," *IEEE Trans. Antenna and Propagation*, vol. 56, no. 2, pp. 571 – 577, Feb. 2008.
- [87] R. A. Krohling and L. D. S. Coelho, "Coevolutionary Particle Swarm Optimization using Gaussian distribution for solving constrained optimization problems," *IEEE Trans. Systems, Man and Cybernetics – Part B: Cybernetics*, vol. 36, no. 6, pp. 1407 – 1416, Dec. 2006.
- [88] Q. H. Nguyen, Y. S. Ong, and N. Krasnogor, "A study on the design issue of memetic algorithm," in *Proc. IEEE Congr. Evolutionary Computation*, Sep. 2007, pp. 2390 – 2397.
- [89] G. L. Xing, J. S. Xue, "A hybrid method for electromagnetic propagated resistivity logging data inversion," *IEEE Trans. Geoscience and Remote Sensing*, vol. 45, no. 3, pp. 649 – 655, Mar. 2007.
- [90] M. A. Spirito, "On the accuracy of cellular mobile station location estimation," *IEEE Trans. Veh. Technol.*, vol. 50, no. 3, pp. 674 – 685, May 2001.
- [91] *XBee ZNet 2.5/XBee-Pro ZNet 2.5 OEM RF Modules*, Digi International Inc., Minnetonka, MN, 2008.
- [92] T. P. Hong, G. N. Shiu, "Solving the K-of-N lifetime problem by PSO," *Int. J. Engineering, Science and Technol.*, vol. 1, no. 1, pp. 136 – 147, 2009.