Intelligent Energy Efficient Localisation in Wireless Sensor Networks

by

Muhammad Farooq-i-Azam

Submitted for the degree of
Doctor of Philosophy

School of Computing and Communications
Faculty of Science and Technology
Lancaster University
United Kingdom

November 2017

Abstract

A wireless sensor network comprises of tiny sensor nodes which communicate with each other through radio frequency communication links. In many applications of wireless sensor networks, the sensor nodes collect data or detect an event and report it to a central node for further processing. Location information of the node is sent with the data else it may not be useful. Therefore, a sensor node should know its geographic position coordinates. Localization solutions, such as GPS are not feasible due to their energy cost and size. Hence, sensor nodes estimate their positions using an algorithm. This thesis focuses on the localization of sensor nodes and related key issues, such as performance evaluation of localization algorithms, development of analytical model and analysis of localization error. Firstly, this thesis proposes three new novel metrics which can be used for the performance evaluation of three different aspects of localization algorithms. Alongside, we also present a comprehensive review of metrics which are used in literature for the measurement and characterization of localization errors. Secondly, we present an intelligent algorithm which we call ripple localization algorithm. The algorithm is distributed, energy efficient and does not require additional hardware for range estimation. The algorithm also provides control over localization granularity which makes it suitable for wide range of wireless sensor network and localization applications. Simulation results show that the algorithm gives good performance and localization accuracy even under irregular radio conditions. Thirdly, we give a new technique for solving multilateration equations and show that the overdetermined system of equations resulting from multilateration can be reduced to a pair of simultaneous equations which can then be solved using conventional techniques, such as Cramer's rule. Fourthly, we develop and present an analytical model of localization error resulting from trilateration.

The multilateration solution technique and analytical model are verified using simulation. Finally, we analyze trilateration errors in short range wireless networks, such as wireless sensor networks and internet of things where the distance estimation errors are sometimes comparable to the actual distances. We investigate the minimum and maximum values of localization errors in these networks. We also derive a number of other useful results. For example, we show that the localization error due to positive range estimation errors equal to the actual distances is 3 times the localization error resulting from the same magnitude of negative range estimation errors. All the results are tested and verified using a comprehensive set of simulation experiments.

Declaration

This thesis is a result of original research work undertaken by myself. Wherever work of others has been referred to, it has been appropriately referenced. The material in this thesis has not been submitted in part or full for the award of a higher degree elsewhere.

Muhammad Farooq-i-Azam
November 2017

Publications

1. M. Farooq-I-Azam, Q. Ni, E. A. Ansari, and H. Pervaiz,“Energy-efficient location estimation using variable range beacons in wireless sensor networks,” *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 1074 – 1079, 2015.

2. M. Farooq-I-Azam, Q. Ni, and E. A. Ansari, “Intelligent energy efficient localization using variable range beacons in industrial wireless sensor networks,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2206 – 2216, Dec 2016.

3. M. Farooq-I-Azam, and Q. Ni, “An analytical model of localization error due to trilateration,” *IEEE Transactions on Industrial Informatics*, Submitted

Acknowledgements

Foremost, my deepest thanks and gratitude to Allah, the Most Beneficent and the Most Merciful for His abundant blessings.

I would like to express my most sincere thanks to my supervisor Prof. Qiang Ni. I consider myself to be really lucky for getting a chance to work under his expert supervision and guidance. I have learned a lot from him. Throughout my PhD, whenever I reached him for advice about personal or research problems, I found him always there with a good piece of advice. He kept providing his unfailing support to me during my PhD pursuit.

There are many people in the InfoLab21 which helped me one way or the other in the course of my PhD endeavour. I thank all of them. In particular, I thank Dr. Philip Benachour, Dr. Ioannis Chatzigeorgiou, Dr Keivan Navaie, Prof. Zhiguo Ding and Dr. Leila Musavian for help in different matters and fruitful suggestions. I am also thankful to my friends and colleagues Haris, Arooj and Wenjuan in the Green Wireless Networking Laboratory. I must also thank our present and former research support and administrative staff Claire Anne, Heather, Debbie, Charlotte and Vicky for their help and support in various matters.

I owe a lot of thanks to my entire family. In particular, I would like to mention the selfless role of my parents, Muhammad Siddique and Karim Bibi (may she rest in peace) first during my early education and then during my career and postgraduate studies. Words are not enough to express my gratitude towards them for everything they have done for me. I am also thankful to my brothers, Akbar-i-Azam and Sohail Azam for their help and support. I would also like to thank my wife, Samara and children Omar, Ali and Amna for bearing with me during my PhD research work.

Dedication

Mother O' Mine

If I were hanged on the highest hill,
Mother o' mine, O mother o' mine!
I know whose love would follow me still,
Mother o' mine, O mother o' mine!

If I were drowned in the deepest sea,
Mother o' mine, O mother o' mine!
I know whose tears would come down to me,
Mother o' mine, O mother o' mine!

If I were damned of body and soul,
I know whose prayers would make me whole,
Mother o' mine, O mother o' mine!

Rudyard Kipling (1865 – 1936)

‘We are not lost. We are locationally challenged.’

John M. Ford

Contents

List of Tables

List of Figures

List of Abbreviations

5G	Fifth Generation
AHoLS	Ad hoc Localization System
APS	Ad hoc Positioning System
ARD	Average Relative Deviation
CAB	Concentric Anchor Beacon Localization
CDF	Cumulative Distribution Function
CPE	Convex Position Estimation
CPU	Central Processing Unit
CVD	Cumulative Vectorial Distance
dBm	Decibel milliwatts
DF	Direction Finding

DOI	Degree of Irregularity
DOLI	Degree of Location Intelligence
DOP	Dilution of Precision
DV	Distance Vector
GDE	Global Distance Error
GDOP	Geometric Dilution of Precision
GER	Global Energy Ratio
GHz	Giga Hertz
GLONASS	*Globalnaya Navigazionnaya Sputnikovaya Sistema*
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HDOP	Horizontal Dilution of Precision
IEEE	Institute of Electrical and Electronics Engineers

IoT	Internet of Things
LMAT	Localization with Mobile Anchor using Trilateration
LORAN	Long Range Navigation
LP	Linear Program
ns-2	Network Simulator version 2
PDOP	Position Dilution of Precision
RAM	Radio Access Memory
RED	Relative Euclidean Distance
RF	Radio Frequency
RLA	Ripple Localization Algorithm
RMS	Root Mean Square
ROCRSSI	Ring Overlapping based on Comparison of Received Signal Strength Indicator
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator

SDLC	System Development Life Cycle
TDOA	Time Difference of Arrival
TDOP	Time Dilution of Precision
TOSSIM	TinyOS Simulator
VDOP	Vertical Dilution of Precision
WAM	Wide Area Multilateration
WiFi	Wireless Fidelity
WPAN	Wireless Personal Area Networks
WSN	Wireless Sensor Network

List of Symbols

$\mathbf{F}$	Framework of ideas
$\mathbf{A}$	Area of concern
$\mathbf{M}$	Methodology
t	Localisation time
t_i	Time taken by a sensor node i for localisation
t_m	Mean localisation time
N	Number of sensor nodes
(x_a, y_a)	Actual position coordinates of a sensor node
(x_{ia}, y_{ia})	Actual position coordinates of a sensor node i
(x, y)	Estimated position coordinates of a sensor node

(x_i, y_i)	Estimated position coordinates of a sensor node i
P_i	Point representing the actual position of a sensor node i
$\hat{P}_i$	Point representing the estimated position of a sensor node i
d_{ij}	The distance between the actual positions P_i and P_j of sensor nodes i and j
$\hat{d}_{ij}$	The distance between the estimated positions $\hat{P}_i$ and $\hat{P}_j$ of sensor nodes i and j
$\mathbf{V}_{ij}$	The vector between the actual positions of sensor nodes i and j from P_i to P_j
$\hat{\mathbf{V}}_{ij}$	The vector between the estimated positions of sensor nodes i and j from $\hat{P}_i$ to $\hat{P}_j$
e_l	Localisation error which is the distance between the actual and estimated positions of a sensor node
μ_l	Mean absolute localisation error of N sensor nodes
e_{ln}	Normalised localisation error

R_s	Radio range of a sensor node
e_{li}	Absolute localisation error of sensor node i
e_{lni}	Normalised localisation error of sensor node i
e_{lrms}	Root mean square value of absolute localisation errors
e_{lnrms}	Root mean square value of normalised localisation error
μ_{ln}	Mean normalised localisation error of N sensor nodes
e_{lmax}	Maximum localisation error
e_{lmin}	Minimum localisation error
e_{ld}	Median of localisation error
e_{lo}	Mode of localisation error
σ_a	Standard deviation of absolute localisation error
e_g	Geometric mean of localisation error

E_G	Global energy ratio
E_A	Average relative deviation
E_D	Global distance error
A_r	Area of actual topology of sensor nodes
A_v	Area of estimated topology of sensor nodes
A_R	Area ratio between estimated and actual topologies
E_S	Sum of distance inconsistencies
E_Q	Quality of fit
R_{ij}	Actual distance between sensor nodes i and j
E_F	Frobenius error
E_C	Performance cost metric
γ	Weighing factor in the evaluation of performance cost metric
C	Performance cost as the average energy per node required for localisation

d_M	Manhattan distance between the actual and estimated positions of a sensor node
d_{Mi}	Manhattan distance between the actual and estimated positions of sensor node i
μ_M	Manhattan distance between the actual and estimated topologies of a wireless sensor network
d_C	Cosine distance between the actual and estimated topologies of a pair of nodes
d_{Cij}	Cosine distance between the actual and estimated topologies formed by a pair of nodes i and j
μ_C	Cosine distance between the actual and estimated topologies of a network
T_C	Tanimoto coefficient
d_T	Tanimoto distance between the actual and estimated topologies of a pair of nodes
d_{Tij}	Tanimoto distance between the actual and estimated topologies formed by a pair of nodes i and j

μ_T	Tanimoto distance between the actual and estimated topologies of a network
d_R	Relative Euclidean distance between a pair of nodes
d_{Rij}	Relative Euclidean distance between a pair of nodes i and j
μ_R	Relative Euclidean distance between the actual and estimated topologies of a network
μ_V	Cumulative vectorial distance between the actual and estimated topologies of a network
k_s	Elastic modulus of a tension spring
l	Length of a spring
x_s	Displacement in a spring
k_t	Rotational sensitivity constant of a torsion spring
θ_t	Angle due to rotation
k_d	Spring constant for type 1 spring

U_{T1}	Average potential energy stored in type 1 spring
U_{T2}	Average potential energy stored in type 2 spring
U_{T3}	Average potential energy stored in type 3 spring
θ_{tij}	Angle due to rotation between actual and estimated topologies of a pair of sensor nodes i and j
μ_s	Spring distance between the actual and estimated topologies of network
e_m	Error momentum
N_s	Number of settled nodes
η_l	Localisation efficiency
δ	Degree of location intelligence
r	Communication radius of a beacon node
r_i	Estimated range of an unknown node from ith beacon node

r_{ai}	Actual range of an unknown node from ith beacon node
e_i	Range estimation error of an unknown node from ith beacon node
B_i	Beacon node i
k	Number of beacon nodes
d_r	Beacon signal step which is the distance between two successive beacon signals in a ripple
t_o	Time stamp of a beacon message
(X_b, Y_b)	Position of a beacon node
P_{ti}	Transmission power used for the transmission of an ith beacon signal
R_i	Radio range of ith beacon signal
P_{min}	Minimum transmission power
P_{max}	Maximum transmission power
R_{min}	Minimum transmission radius

R_{max}	Maximum transmission radius
U_i	Unknown node i
P_t	Transmitted power
P_r	Received power
G_t	Gain of transmitter antenna
G_r	Gain of receiver antenna
d	Distance between transmitter and receiver antennas
λ	Wavelength of radio waves
α	Path loss exponent
n	Number of beacon signals in a ripple
P_T	Total power transmitted by a beacon node
P_S	Power saved by a beacon node
η_P	Energy efficiency of a beacon node
η_{Pmax}	Maximum energy efficiency of a beacon node

β	Bound on random error in the beacon signal radio range
r_{12}	Difference in the distance of receiver from the first pair of master and secondary stations
E_x	x component of localisation error
E_y	y component of localisation error
k_1	One of the two localisation constants in the x coordinate of position and localisation error
k_2	One of the two localisation constants in the x coordinate of position and localisation error
k_3	One of the two localisation constants in the y coordinate of position and localisation error
k_4	One of the two localisation constants in the y coordinate of position and localisation error
C_1	One of the two position constants
C_2	One of the two position constants
E_1	One of the two error factors in the determination of E_x and E_y

E_2	One of the two error factors in the determination of E_x and E_y
θ_l	Direction of localisation error
ζ_i	Error term given by $2r_{ai}e_i + e_i^2$
ζ_{imin}	Minimum value of error term ζ_i
ζ_{imax}	Maximum value of error term ζ_i
$\mathbf{H}$	Hessian matrix
E_{1sad}	E_1 at saddle point
E_{2sad}	E_2 at saddle point
E_{1min}	Minimum value of E_1
E_{1max}	Maximum value of E_1
E_{2min}	Minimum value of E_2
E_{2max}	Maximum value of E_2
E_{1+}	E_1 due to positive distance estimation errors
E_{1-}	E_1 due to negative distance estimation errors

E_{2^+}	E_2 due to positive distance estimation errors
E_{2^-}	E_2 due to negative distance estimation errors
E_{xsad}	E_x at saddle point
E_{ysad}	E_y at saddle point
E_{xmin}	Minimum value of E_x
E_{xmax}	Maximum value of E_x
E_{ymin}	Minimum value of E_y
E_{ymax}	Maximum value of E_y
E_{x^+}	E_x due to maximum positive distance estimation errors
E_{x^-}	E_x due to maximum negative distance estimation errors
E_{y^+}	E_y due to maximum positive distance estimation errors
E_{y^-}	E_y due to maximum negative distance estimation errors

E_{xd^+}	E_x due to positive and equal distance estimation errors
E_{xd^-}	E_x due to negative and equal distance estimation errors
E_{yd^+}	E_y due to positive and equal distance estimation errors
E_{yd^-}	E_y due to negative and equal distance estimation errors
$\mathbf{p}$	Proportion of distance estimation error
p	Absolute value of error proportion
E_{xp^+}	E_x due to positive and unequal but proportionate distance estimation errors
E_{xp^-}	E_x due to negative and unequal but proportionate distance estimation errors
E_{yp^+}	E_y due to positive and unequal but proportionate distance estimation errors
E_{yp^-}	E_y due to negative and unequal but proportionate distance estimation errors

e_{lp^+}	Localisation error when unequal but proportionate distance estimation errors are additive
e_{lp^-}	Localisation error when unequal but proportionate distance estimation errors are subtractive
ξ	Error coefficient

Chapter 1

Introduction

A wireless sensor network comprises of tiny sensor nodes which communicate using radio frequency links. A diverse range of applications for wireless sensor networks has been envisaged and many more applications are being proposed and developed [1–10]. Location information of sensor nodes is an important aspect in majority of these applications. For example, data collected by a sensor node may not be useful if the position from where it were gathered is not known. Therefore, the sensor node should send its location information along with the data. This is possible only when a sensor node knows its own position. Existing solutions, such as GPS, are not suitable for this purpose due to various factors including size and energy cost. Therefore, new localisation algorithms are being developed which sensor nodes can deploy to estimate their positions [11–15]. The localisation algorithm should be designed and its performance should be evaluated considering the characteristics, limitations and constraints of sensor nodes and the network they are part of. For example, due to volume, size, mass and energy constraints, it is not feasible to use specialised localisation hardware device in a sensor node. Therefore, it is desirable that the sensor node is able to estimate its location using radio connectivity and with the help of a set of beacon nodes and other sensor nodes. When the sensor nodes in a network have estimated their positions, they can use this information to either solve various network related issues, such as geographic position based routing, or help in the development of many applications of wireless sensor networks, such as area monitoring [16–21]. As energy is a scarce resource in a wireless sensor network, the localisation algorithm must be energy efficient.

1.1 Motivation

In many applications of industrial wireless sensor networks, sensor nodes need to determine their own geographic positions. Examples of such industrial applications include gas leakage detection, industrial IoT, industrial fire detection, underground pipeline inspection, target tracking, habitat monitoring and area surveillance [22–24]. In such cases, unknown sensor nodes need to employ an intelligent localisation algorithm to estimate their geographic position coordinates usually with the assistance of a few beacon nodes. The beacon nodes know their positions a priori either because these are placed at pre-determined locations or are equipped with location finding device, such as, global positioning system (GPS). Due to energy and size limitations, all unknown sensor nodes cannot be equipped with such extra piece of hardware.

Location information of industrial sensor nodes is important because of two major reasons. First, the sensor nodes must send their geographic position coordinates with the sensed data because data alone without location information may not be useful. For example, in the case of an industrial fire detection application, the sensor node should send the geographic coordinates along with the event information so that location of fire is known. Second, there are many services and protocols that use location information to work. For example, certain routing protocols, such as [25, 26], sensing coverage [27, 28], topology management [29] and clustering strategies [30] depend upon location information of sensor nodes.

Depending upon whether an algorithm uses range (distance) for location estimation, it may be classified as range based or range free. Range based algorithms, such as [31] determine distances or angle information to estimate their positions. In particular, range is determined using absolute point-to-point estimates. Range free algorithms, such as [32–35] estimate position without using range or angle information. Localisation algorithms may also be classified as anchor based or anchor free. In anchor free localisation algorithms, unknown nodes estimate their relative position coordinates without using any beacon nodes, such as in [36]. In anchor based localisation algorithms [31–33], unknown nodes estimate their absolute position coordinates using anchor or beacon nodes. Majority of localisation algorithms are anchor based. Another classification of localisation algorithms is single hop versus multi hop. Single hop algorithms [35] use only immediate neighbours at single hop distance for position estimation, whereas multi hop algorithms [37] use hop by hop communication with nodes at multi hop distance for this purpose. Localisation algorithm may be designed for outdoor unconstrained [32–38] or indoor constrained environment [39]. Similarly, localisation algorithm may use central

[40] or distributed processing [32–35]. When using central processing, network information is communicated to a central processor where node positions are calculated and then communicated back to sensor nodes. Localisation algorithm that employ central processing or multi hop communication usually involve communication overhead at the expense of energy.

1.2 Contributions

This thesis makes multiple contributions. Some of the core and major contributions include metrics for the performance evaluation of localisation algorithms, ripple localisation algorithm, new solution technique for multilateration, development of analytical model and analysis of the localisation error. An overview of each of these contributions is given in this section.

1. **Performance Evaluation and Metrics** : Performance evaluation is an important and significant step in the system development life cycle (SDLC) of localisation algorithms. This thesis details all the steps involved in the life cycle and also describes the criteria and parameters that can be used in the performance evaluation. Accuracy of estimated position is generally considered the most important and significant criteria. However, this thesis highlights that accuracy should be evaluated in conjunction with other criteria such as localisation cost, coverage, scalability and robustness. Only then are we able to get a complete picture of the overall performance of a localisation algorithm. Furthermore, the thesis reviews a number of important metrics that are used for the performance evaluation. This

chapter intends to serve as a guideline and reference for the performance evaluation of localisation algorithms. No comprehensive review on metrics and performance evaluation of localisation algorithms is available in the previously published literature. In addition, it proposes three novel metrics that evaluate three different aspects of a localisation algorithm. The newly proposed metrics measure localisation time and localisation error trade off, accuracy and coverage. To the best of our knowledge, no metrics are available in the existing literature which measure time and error trade off and coverage ability of localisation algorithms.

2. **Ripple Localisation Algorithm** : The thesis proposes a novel and intelligent localisation algorithm for wireless sensor networks deployed in the outdoor unconstrained environment. The proposed algorithm is called ripple localisation algorithm (RLA) as the beacon signals are transmitted in the fashion of a ripple by varying the transmission power. The proposed algorithm exploits the radio signal strength and information in the transmitted beacon messages for the estimation of distances. After estimating the distances, it then uses multilateration for estimation of position. The algorithm works in a distributed manner so that each individual node estimates only its own position. It then has a choice to transmit this information only to selective and trusted nodes thereby preserving its location privacy. As the algorithm uses radio frequency information for distance estimation, additional hardware is not required for ranging. This conserves energy. In addition, the communication overhead is also avoided due to the distributed nature of the algorithm. As a result, the algorithm is energy efficient. The algorithm is implemented using

MATLAB[1]. Performance of the proposed algorithm is verified and tested using a diverse set of metrics. Performance of the algorithm is also compared with two other algorithms. Results of the simulation experiments show that the proposed ripple localisation algorithm provides good localisation accuracy under varying radio conditions. We also analyse the energy utilisation and derive an analytical expression for the energy efficiency which has not been done previously. Furthermore, the ripple localisation algorithm provides control over localisation granularity thereby making it suitable for a wide range of applications requiring low to high degree of accuracy. This is a new feature and is not available in the previously developed localisation algorithms.

3. **Analytical Model of Localisation Error** : Multilateration and trilateration are important core techniques which can be used in an algorithm for estimation of position in different types of networks. A number of solution techniques are available for the multilateration equations. We show that the overdetermined system of equations which result from multilateration can be reduced to merely two equations in the variables x and y constituting the unknown coordinates (x, y). This set of two equations can then be solved simultaneously, for example using Cramers rule, for the estimation of unknown position (x, y). Compared to the previously used solution technique, our proposed solution does not require computation of an inverse, and hence greatly simplifies the solution to the multilateration equations. Using this approach of solution to multilateration equations, we develop an analytical model of trilateration localisation error. The

[1]The complete code of the implementation is open source and is available on the Internet and also from the authors.

analytical model is useful for the investigation and analysis of various aspects of localisation errors. This can eventually help in the better estimation of position. To the best of our knowledge, this thesis is the first attempt to present an accurate analytical model of trilateration localisation error.

4. **Trilateration Error Analysis** : The analytical model developed earlier in this thesis is used for the analysis of trilateration localisation error. The analytical model gives localisation error as a function of distance estimation errors. Therefore, we analyse the localisation error by varying and using different combinations of distance estimation errors as input to the analytical model. We derive a number of important results using this analysis. A summary of these results is provided in the relevant chapter. In particular, we determine the conditions under which the localisation error can be reduced to zero. We also determine the minimum and maximum values of trilateration localisation errors as a function of distance estimation errors. All these results derived using the analytical model are tested and verified using a comprehensive set of simulation experiments. To the best of our knowledge, this is the first attempt to derive precise analytical expressions for the minimum and maximum values of trilateration localisation error. The additional derived results are also unique and novel and are not available in the previously published literature.

1.3 Thesis Outline

This thesis is organised into six chapters whose details are as follows. In **Chapter 1**, we give an overview and background of the work presented in this thesis. A detail of the contributions made by this thesis is also provided.

Chapter 2 is about the performance evaluation of localisation algorithms which is an important step in the system development life cycle of localisation algorithms for wireless sensor networks. An overview of the system development life cycle of localisation algorithms is provided at the start of the chapter. The criteria and parameters used for the performance evaluation of localisation algorithms are then discussed and described. At the core of the chapter, a number of metrics used for the evaluation of different aspects of a localisation algorithm are described in detail. When evaluating the performance of a localisation algorithm, these metrics can serve as a reference and a subset can be chosen for use during the testing and performance evaluation of the algorithm.

A novel and intelligent localisation algorithm, that we call ripple localisation algorithm (RLA) is proposed and evaluated in **Chapter 3**. The algorithm uses only radio frequency information for localisation and does not use any additional hardware for this purpose. In addition, the algorithm works in a distributed fashion. In this way it preserves localisation privacy of individual nodes and avoids unnecessary communication overhead to a central localisation node. An analysis of the energy utilisation of the ripple localisation algorithm is also provided and it is shown that the algorithm is energy efficient. Performance of the algorithm is evaluated and compared with two other algorithms using simulation experiments. A comprehensive set of metrics including three of our own novel metrics are used for the testing and performance evaluation. The results of the simulation experiments are then analysed and discussed before concluding the chapter.

In **Chapter 4**, multilateration and trilateration are discussed and analysed which are important techniques used for estimation of position. In addition, we also analyse localisation errors resulting from position estimation using these techniques. We show that the overdetermined system of equations resulting from multilateration can be reduced to two linear equations which can be solved simultaneously using Cramers rule for the estimation of position. This solution technique is then exploited to develop an analytical model of the localisation error which results from trilateration.

In **Chapter 5**, the analytical model of the localisation error developed in the previous chapter is used for the investigation and analysis of the relationship between the errors in the distance estimates and error in the estimated position. The extreme values of localisation error as a function of distance estimation errors are derived. In particular, the localisation error is broken down into components. Each component is analysed for the minimum and maximum values. In addition, a number of useful results are derived which can be helpful in the design and development of localisation algorithms. Finally, all the analytical results are verified using a number of simulation experiments. The numerical results of the simulation experiments are analysed and compared with the results derived using the analytical model.

This thesis concludes with **Chapter 6**. We reflect upon the work presented in the thesis in this final chapter. The areas of future work which can be based upon the work in the thesis are also identified.

Chapter 2

Performance Evaluation of Localisation Algorithms

Performance evaluation is an important step in the development life cycle of a localisation algorithm. In this chapter, we discuss its various aspects in the context of wireless sensor networks. In particular, we describe the performance metrics that are used for the testing, evaluation, comparison and analysis of localisation algorithms. These metrics are employed for the identification, characterisation and measurement of estimation errors due to localisation. Some of these metrics are particularly designed to measure and evaluate specific aspects of localisation errors. In addition to the performance metrics, we also give an overview of the system development life cycle, criteria and parameters for the performance evaluation of localisation algorithms for wireless sensor networks.

2.1 Introduction

The positions estimated by sensor nodes using a localisation algorithm usually have estimation errors. The characteristics of localisation errors depend upon the algorithm being used for position estimation. Therefore, when a sensor network application is being developed, various candidate algorithms for localisation may be evaluated and compared with respect to their error characteristics. Similarly, when a new localisation algorithm is developed, its performance may be tested, evaluated and analysed using simulation or any other techniques. The performance of the newly proposed method is also compared against other localisation algorithms.

Performance evaluation and testing of localisation algorithms is carried out using a set of metrics and parameters. Results of the performance evaluation may vary for different sets of metrics and parameters. For example, the performance of a localisation algorithm may look good when measured using a certain accuracy metric. However, it may not give desirable results when its performance is judged using coverage metrics. Therefore, choice of parameters and metrics plays an important role in the performance evaluation of a localisation algorithm. Some metrics may show greater sensitivity to certain types of errors. Hence, such metrics might be better suited to detect and analyse the errors to which they are sensitive.

The most widely used measure of localisation error is the distance between the estimated and actual positions. While this is a simple and easy to use metric, it has a drawback. It considers the node in isolation from the rest of the network for the measurement of error. It does not consider the effect of the estimated position on the topology i.e. the geometry formed by the positions of the node and its neighbour nodes. This is illustrated with the help of an example similar to the one given in [41].

Consider three unknown sensor nodes whose actual positions are represented by P_1, P_2 and P_3 as shown in Fig. 2.1. When the nodes estimate their positions as P_{1A}, P_{2A} and P_{3A} using an algorithm A, they have localisation errors e_{l1}, e_{l2} and e_{l3} calculated as the distance between the estimated and actual positions. The same nodes then estimate their positions as P_{1B}, P_{2B} and P_{3B} using another localisation algorithm B. The localisation errors for the three nodes are still the same as in algorithm A i.e. e_{l1}, e_{l2} and e_{l3}. However, the resultant topology formed by the estimated positions in both the cases is very different. While the Euclidean distance localisation errors between the estimated and actual positions are the same for both the algorithms, the topology formed by the positions estimated by Algorithm B is much closer to the actual topology than the one formed by the estimated positions using the Algorithm A. Therefore,

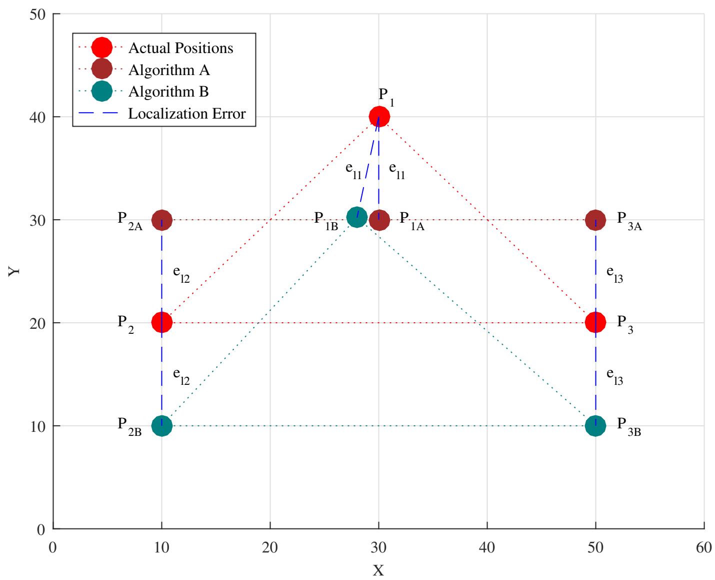


Figure 2.1: Different localisation algorithms may result in different estimated topologies.

it is important that performance metrics that consider the topology formed by the geometry of positions of nodes are developed and used for the performance evaluation of localisation algorithms. These metrics measure the closeness or distance between the estimated topology and actual topology of the sensor nodes. It is to be noted that the term topology in this thesis implies physical topology. We further define physical topology as the physical layout or map formed by the sensor nodes in the sensor network. Alternatively, it is the geometry formed by the positions of sensor nodes in the network.

In this chapter, we focus on some important aspects of performance evaluation of localisation algorithms for wireless sensor networks. First, we describe the system development life cycle of localisation algorithms for the purpose of giving the context. Second, we describe the criteria used for the performance evaluation of localisation algorithms. Though, localisation accuracy is a predominantly important criterion, there are other factors that should also be given consideration. Therefore, we describe the factors in the criteria that are relevant to the performance evaluation of localisation algorithms. Third, we also discuss the parameters that are used for performance evaluation of an algorithm against the described criteria. Fourth, we present a detailed study of the metrics used for the testing, evaluation and comparison of localisation algorithms.

Rest of this chapter is organised as follows. We give a summary of the related work in the next section. System development life cycle and criteria for the performance evaluation of localisation algorithms are described in Section 2.3 and Section 2.4 respectively. In Section 2.5, we discuss the parameters against which performance is evaluated. Performance evaluation metrics are described in detail in Section 2.6. We conclude with Section 2.8.

2.2 Related Work

Localisation is important in many areas of science and engineering. Therefore, localisation algorithms and their performance evaluation have been subjects of interest in different fields including wireless sensor networks. In [42], algorithms for workpiece localisation are investigated. In particular, reliability of localisation is analysed with respect to translational and rotational errors. A distributed anchor free localisation algorithm which uses mass spring relaxation and optimisation is presented in [43]. A new metric called global energy ratio (GER) is also proposed and used for the performance evaluation of the algorithm. The GER metric considers the entire topology for the computation of error. Key aspects and types of localisation algorithms are described in [44]. The authors also study and analyse the basic and important factors that need to be considered while evaluating the performance of a localisation algorithm.

Performance evaluation of localisation algorithms is discussed from different angles in [45]. Criteria, metrics, network topologies, and models for simulation, power consumption, radio propagation and communication, and development cycle are described. The authors argue that the performance evaluation criteria other than localisation accuracy are also important. In particular, the criteria should be based upon the requirements of the application of the sensor network being deployed. Even the requirements of localisation accuracy will vary from application to application. For example, scalability might be an important factor for some applications of sensor networks. For such applications, localisation algorithms using centralised processing approach may not be suitable even if they provide good localisation accuracy. A centralised localisation algorithm performs major processing at a central node and, therefore, may not scale well. Hence, despite their simplicity and easiness of implementation, centralised localisation algorithms may not suite large deployments of sensor networks.

Multiscale dead reckoning algorithm proposed in [46] uses force directed graph layout technique for localisation of sensor nodes. In addition, two new metrics are proposed and used for the performance evaluation of the algorithm. New metrics to measure the similarity of original and estimated topologies of sensor network are investigated in [41]. The topologies are shifted, rotated and randomly distorted, and different metrics are used to measure the change in topology. It is observed that a metric may be sensitive to one type of change in topology and insensitive to other types of changes. For example, cosine similarity metric is observed to be sensitive to rotation but insensitive to shift in topology. However, Euclidean and cumulative vectorial distance metrics are sensitive to shift but insensitive to rotation. Therefore, appropriate metrics can be selected for the performance evaluation of localisation algorithms depending upon the application of the wireless sensor network. Mass spring relaxation localisation technique with various optimisation steps is used in [47] for the evaluation of different performance metrics which consider the complete topology of a sensor network. Furthermore, a new area based metric is also proposed and evaluated.

2.3 System Development Life Cycle

We describe the system development life cycle (SDLC) of localisation algorithms in this section to give the context of performance evaluation. Investigation of localisation algorithms for wireless sensor networks is a research process which can be represented as a research cycle as shown in Fig. 2.2. SDLC is a subset of this process, and performance evaluation is, in turn, an important step in the SDLC of localisation algorithms. For an overview, some of the important steps in the development life cycle of a localisation algorithm are identification of objectives, design and implementation, performance evaluation and research methodology. We give a brief description of these steps below.

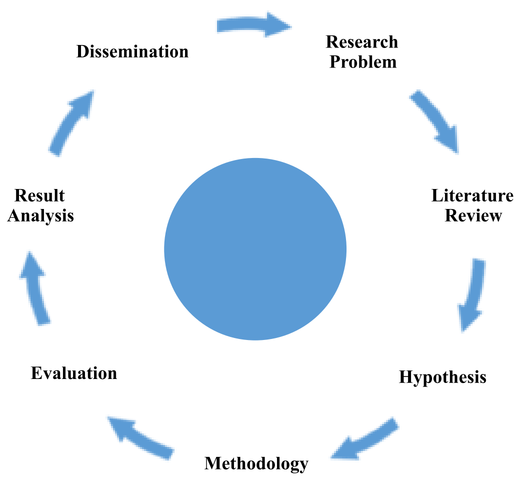


Figure 2.2: A typical research cycle.

2.3.1 Objectives

The objectives, challenges, constraints and characteristics that a localisation algorithm is expected to satisfy must first be identified. Some of the common objectives and characteristics are highlighted below.

i. As wireless sensor nodes communicate using radio frequency links, it is desirable that localisation algorithms exploit this aspect to estimate positions. In this way, the algorithm can be energy efficient as the size due to additional hardware and energy cost can be avoided.

ii. Wireless sensor networks are ad hoc networks. Therefore, a good localisation algorithm either exploits this ad hoc nature or keeps it into consideration while estimating position of a sensor node.

iii. A localisation algorithm should have a low response time so that the sensor node is able to estimate its position quickly in a small time. In this way, the sensor nodes can start functioning as soon as these are deployed.

iv. Different localisation algorithms may have different granularities of estimated positions. Therefore, a localisation algorithm should have localisation granularity matching the requirements of the application for which it is being selected or designed.

v. A good localisation algorithm is robust and maintains its level of accuracy against variations in the input parameters.

vi. Some applications of wireless sensor networks are dynamic in nature. Sensor nodes may be removed or added with the passage of time. Similarly, a sensor network may comprise of only a few nodes or at other times thousands of nodes. The localisation algorithm should scale well for all these variations in the network without compromising accuracy of estimated position.

vii. Design of any aspect of a wireless sensor network should consider to utilise energy efficiently as the sensor network is generally autonomous and the batteries may not be replaced during the entire life time of the deployment. Therefore, a localisation algorithm should be designed in such a manner that it is energy efficient. An energy efficient algorithm may also be energy aware in that it modifies its behaviour according to the remaining energy resource of the sensor node.

viii. Majority localisation algorithms use reference nodes, also known as beacon or anchor nodes, for position estimation. Performance of an algorithm may be affected with a change in the number of beacon nodes in the sensor field. A good localisation algorithm is adaptive to the change in the number of beacon nodes. It is able to estimate position with change in the number of beacon nodes without much effect on the accuracy of the estimated positions.

ix. A localisation algorithm is efficient if it is able to localise a large number of sensor nodes with the help of a minimum number of beacon nodes.

x. A good localisation algorithm is generic and universal in that it is able to estimate node positions under varying conditions. Particularly, the algorithm should be able to localise nodes in both indoor constrained and outdoor unconstrained environments.

Merely an ideal algorithm will be able to satisfy all these requirements. A practical localisation algorithm may not be able to meet all the aforementioned characteristics and may satisfy only a subset of these requirements.

2.3.2 Design and implementation

Design of a localisation algorithm depends upon the research technique being used. For example, the algorithm proposed in [1] considers both known and unknown transmission powers of sensor nodes. Furthermore, it uses both range estimates and angle measurements for position estimation. The design process may or may not yield an analytical model of the algorithm. Choice of a programming language and other implementation details depend upon the platform selected for the implementation of the algorithm.

2.3.3 Performance evaluation

A number of methods can be used for the testing and performance evaluation of an algorithm developed for any layer of the protocol stack of wireless sensor networks. These methods can generally be classified into the following categories:

1. Analytical modelling
2. Simulation
3. Emulation
4. Testbeds
5. Real deployment

A performance evaluation strategy may use any one or combination of methods from the above list for the testing of an algorithm.

2.3.3.1 Analytical modelling

The performance of an algorithm may be evaluated using an analytical model. It is the preferred method when quantitative analysis is desired. However, an analytical model may sometimes not yield accurate results for complex wireless sensor networks. A wireless sensor network is complex in nature due to a number of factors. For example, the number of nodes may range from a few nodes to thousands of nodes. Some of these nodes may be stationary and others may be mobile. Some of the nodes may leave the network and some new nodes may become part of the network with the passage of time. This can result in a dynamic topology of a sensor network. In a similar manner, the wireless channel characteristics depend upon the environment in which the network is deployed.

All the aforementioned factors contribute to the complexity of a wireless sensor network. When an analytical model of such a complex network is developed, certain assumptions and simplifications are usually made. Therefore, performance evaluation of an algorithm using the analytical model may lead to errors in results. This is in particular the case when the assumptions and simplifications are about the factors which are important in the performance of an algorithm being evaluated. It is for this reason that evaluation using an analytical model is usually followed up by another testing methodology, such as simulation.

2.3.3.2 Simulation

The performance of an algorithm can be evaluated by using simulation software. The algorithm is implemented using software code. Similarly, the necessary components of wireless sensor network, such as, layers of protocol stack, topology and radio channel are modelled in the simulation program. Performance of the algorithm is then evaluated by running simulation experiments. Simulation is the most widely used methodology for performance evaluation due to its flexibility. For example, the number of sensor nodes in the network can be easily scaled from a few nodes to thousands of nodes. Furthermore, a number of simulation experiments can be executed in a matter of minutes. Such options are time consuming and expensive in a real deployment of a sensor network. In particular, a localisation algorithm is usually evaluated for an ad hoc distributed wireless sensor network for various densities of sensor nodes. Simulation provides a flexible and feasible mechanism as the network can be easily scaled up or down.

One of the advantages provided by simulation is that various parameters can be easily modified. Hence, performance of an algorithm can be evaluated under a diverse set of parameters. For example, ratio of beacon and unknown nodes can be easily varied and the corresponding accuracy of a localisation algorithm can be measured. Similarly, many other parameters can be easily varied and response of the algorithm being evaluated can be recorded. Another advantage afforded by simulation is the modelling of future technology which is yet not available for real deployment. Therefore, an algorithm can be tested for deployment in such a technology which is yet to be developed. Obviously, the cost of simulation is also much lower when compared with real deployment.

A number of software tools for the simulation of wireless sensor networks are available. Many comparative studies and surveys of these simulation tools have been performed and are available in literature, such as [48–53]. Some of these simulation tools popular among the research community are MATLAB, network simulator ns-2 and OMNeT++. Apart from personal preferences, choice of simulation tool also depends upon the layer of the protocol stack for which the algorithm is being developed. Network simulator ns-2 is an open source simulator which is suitable for modelling lower level details of lower layers but is reported to have scalability issues when the number of sensor nodes is increased to a large value [52]. OMNeT++ is a discrete event simulator which is also open source. It has no scalability issues and can perform simulations involving a large number of sensor nodes [50, 52]. MATLAB has also been used extensively for the simulation of localisation algorithms. It provides a programming environment as well as many functions making it suitable for the simulation of many types of algorithms for wireless sensor networks [48, 54].

2.3.3.3 Emulation

Emulators are designed to run an algorithm in the environment of a particular hardware platform. Hence, the software code of the algorithm can be directly ported to the actual hardware after testing has been done in the emulator. Results of the performance testing are closer to a real deployment but are constrained to the hardware platform for which the emulation experiments have been conducted. However, once emulated, an algorithm can be easily and quickly deployed in a real environment. An example of wireless sensor network emulator is TOSSIM, which runs in TinyOS operating system and is designed primarily for Berkeley Mica Motes.

2.3.3.4 Testbeds

Another option for the testing and performance evaluation of algorithms for wireless sensor networks is the use of testbeds. A testbed is an actual deployment of a wireless sensor network. Hardware and software instruments are provided in the sensor network for the collection of data of interest. An interface to the testbed is usually made available for the research community to perform various tasks. Using the interface, a remote user can, for example, run programs and experiments and also collect the resulting data.

Various testbeds available to the research community are reviewed in [50, 51, 55, 56]. Using a testbed, the performance of an algorithm can be evaluated for the particular hardware platform used in the testbed. However, there are drawbacks involved as well. The testbeds are deployed indoors. Therefore, testing and performance evaluation of algorithms meant for outdoor environment cannot be carried out using a testbed, as the radio channel characteristics are vastly different for both the environments. Another important issue with the testbeds is scalability. The wireless sensor network in a testbed usually comprises of less than 100 sensor nodes. Therefore, testbeds are not suitable for the performance evaluation of algorithms which are designed for outdoor environment and where the number of nodes is scaled from a few nodes to hundreds of nodes in a number of experiments. When compared with simulation, the testbeds can only be used for the partial assessment and evaluation and to study only some aspects of an algorithm.

2.3.3.5 Real deployment

Performance evaluation of an algorithm using the real deployment of a wireless sensor network is not feasible as a first option as discussed in [49, 50]. It is not possible to change certain parameters and settings in an actual deployment of a sensor network. If possible at all, the process is difficult and time consuming. The number of sensor nodes in an actual deployment for the purpose of performance testing is usually limited because change of settings and parameters in a large number of nodes would be costly and time consuming. Therefore, the algorithm cannot be subjected to rigorous testing by scaling the number of nodes in the sensor field. Experimental results of the performance evaluation through actual deployment are restricted only to the hardware platform and topology deployed in the sensor network. Sometimes, it may not even be possible to implement an algorithm using the existing hardware and software platforms due to their limitations. This may especially be the case when an algorithm is being designed and tested for a future technology.

2.3.4 Methodology

Let a framework of ideas be represented by F, methodology by M and area of concern by A. Then, according to Checkland and Holwell, these three are related to each other as shown in Fig. 2.3 [57]. The methodology may be used either in a simple or complex engineering system [58].

As is evident from Fig. 2.3, the use of a methodology results in the creation of new knowledge about framework of ideas F, area of concern A, and the methodology M itself. The usefulness of the methodology can be assessed in the light of the new knowledge. As a result, the methodology can be modified or even replaced by a new methodology in the light of new results.

Let us now consider this in the context of localisation algorithms. Let localisation and its application be the area of concern A, an algorithm that

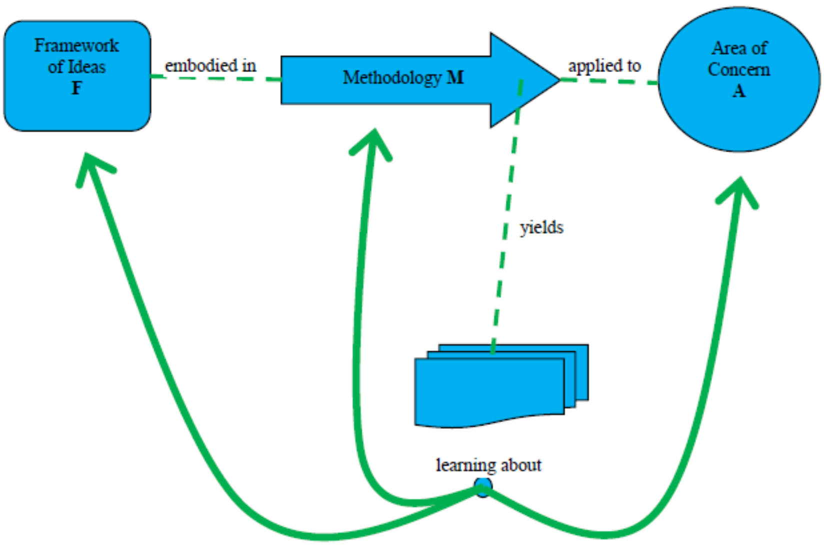


Figure 2.3: Role of methodology in research.

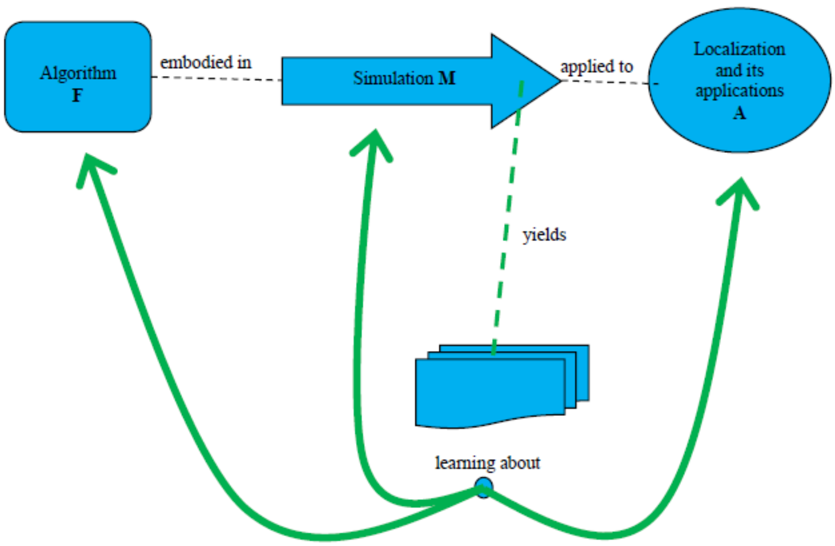


Figure 2.4: Role of simulation in the development and performance evaluation of a localisation algorithm.

our research may produce be the framework of ideas F, and the testing of this algorithm, say using simulation, be the methodology M. Like Fig. 2.3 this can be represented using Fig. 2.4. Hence, the performance evaluation methodology, say simulation, will create new knowledge about the algorithm, the area of concern which is localisation and the methodology of simulation process itself. Through an iterative process, improvements in all three entities can be achieved. Flow of the performance evaluation methodology and the iterative process are shown in Fig. 2.5.

The correctness, accuracy and application of new knowledge about the framework of ideas F, area of concern A and the methodology M may be decided at different levels. Before being accepted as valid, the new knowledge passes through various stages of approval. For example, if the findings and results are published after feedback and reviews by peers, the work has passed a checkpoint of approval and is considered as a valid work.

2.4 Criteria for Performance Evaluation

The criteria for the evaluation of a localisation algorithm generally depends upon the application for which it will be used. However, the categories of metrics given below are usually common for majority of applications.

1. Accuracy
2. Cost
3. Robustness
4. Coverage

5. Scalability

Among these, accuracy and cost are generally considered the most important and widely used criteria for the performance evaluation of localisation algorithms [41–45]. It is also to be noted that the particular set of metrics best

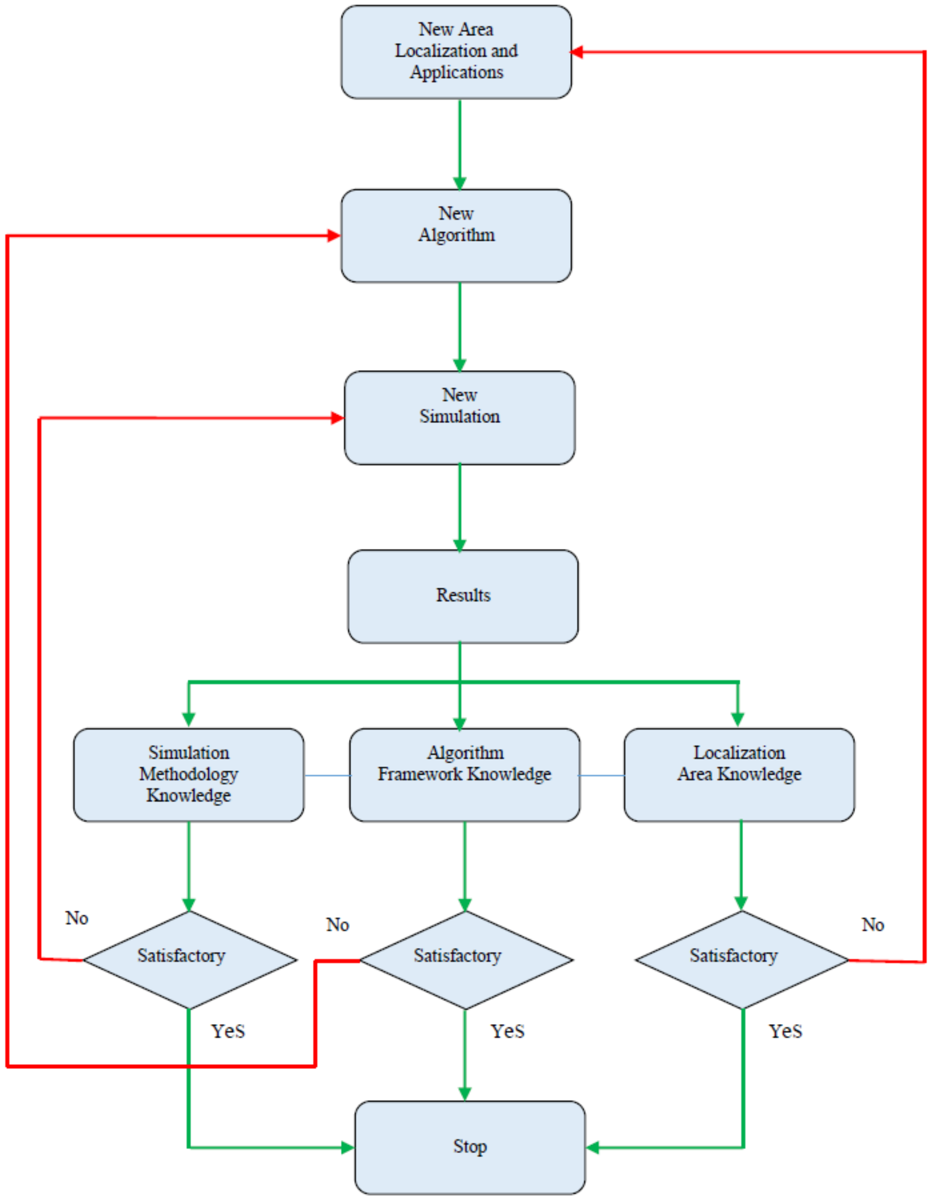


Figure 2.5: Performance evaluation of a localisation algorithm.

suited for the performance evaluation of a localisation algorithm depends upon the scope and application of the algorithm. For example, some applications of wireless sensor networks may require only coarse grained position estimation and strict accuracy may not be important. Similarly, some sensor networks may be deployed in a limited area and scalability may not be an important factor.

2.4.1 Accuracy

Accuracy metrics quantitatively measure the closeness of estimated and actual positions. This closeness is usually measured in terms of distance between the estimated and actual positions. Some accuracy metrics consider only individual nodes for determination of accuracy. However, some other metrics consider pairs of nodes or complete topology of a sensor network for the measurement of accuracy.

2.4.2 Cost

Cost metrics measure the amount of resources consumed for localisation. The cost may be calculated per node or for the entire network. Important cost factors include the following:

i. Energy

ii. Time

iii. Memory

iv. Hardware

v. Beacon nodes

The above categories of cost may be further subdivided. For example, the total energy cost comprises of the energy required for communication and the energy consumed by processing and computation. Hence, the total energy is sum of communication energy and processing energy. Similarly, the time cost comprises of processing time and communication time for the exchange of information. Processing time depends upon the computation complexity of localisation algorithm. Time cost and energy cost are related. Higher time cost implies higher energy cost. Memory is another factor. Localisation algorithms which require smaller amount of memory for processing and storage are usually preferable. Number of anchor nodes required to achieve a certain level of accuracy should also be taken into account while estimating the cost of a localisation algorithm. Some algorithms may need additional hardware to achieve localisation. Furthermore, the localisation cost may vary as the sensor network is scaled. Therefore, the aforementioned cost factors may be calculated as the number of sensor nodes and size of the sensor field is varied. If t_i is the time taken by a sensor node i for localisation, then the mean localisation time t_m for N sensor nodes is given by

$$t_m = \frac{1}{N} \sum_{i=1}^{N} t_i \tag{2.1}$$

2.4.3 Robustness

A localisation algorithm estimates position based on the input of certain parameters. For example, range based algorithms use range information between nodes for position estimation. For good performance, a localisation algorithm should be robust to changes in its input. Small errors in the input should not result in large errors in the estimated position.

2.4.4 Coverage

Ideally, a localisation algorithm should be able to help localise all the nodes in the sensor field. Therefore, an important aspect in the performance evaluation of a localisation algorithm is the coverage i.e. the number of sensor nodes it is able to localise. The coverage may be evaluated by varying other network parameters e.g. number of sensor nodes, beacon nodes, and size of the sensor field.

2.4.5 Scalability

A sensor network may scale in terms of number of nodes or area or both. The localisation algorithm being used for position estimation should be robust and retain its desired level of accuracy, time complexity and other traits as the sensor network is scaled up or down. Scalability of a localisation algorithm can be evaluated by measuring and comparing its performance as the size of the network is varied.

2.5 Parameters

The performance metrics are evaluated against certain variables or parameters of the sensor network. The metrics constitute only one axis of the plot. The other axis is the parameter against which the metric is being evaluated.

2.5.1 Number of beacon nodes

Majority of localisation algorithms depend upon anchor or beacon nodes for position estimation. Number and placement of beacon nodes may affect the performance of a localisation algorithm. Therefore, the metrics should be calculated and analysed as the number of beacon nodes in the sensor field is changed. A plot of the metrics against number of beacon nodes will then reflect performance of localisation algorithm as the number of beacon nodes is varied.

2.5.2 Number of sensor nodes

As the number of sensor nodes is varied, the distance between neighbour nodes varies. Similarly, the number of neighbours of a sensor nodes also varies. Both these factors may affect the localisation performance of an algorithm in terms of time and energy cost and localisation accuracy.

2.5.3 Size of sensor field

Performance of a localisation algorithm may be influenced by the size of the sensor field in which the network is deployed. Distance and connectivity among beacon and sensor nodes may be affected for different sizes of the sensor field. As a result, the performance of a localisation algorithm may also be affected. Therefore, performance of localisation algorithms should be evaluated for different sizes of sensor field.

2.5.4 Topology

Some localisation algorithms use single hop communication for estimation of position. Their performance does not vary with the change of topology. However, performance of localisation algorithms which use multi hop communication is affected by change in topology. For example, a localisation algorithm may not correctly estimate range using multi hop communication in a C shaped concave topology. Therefore, performance metrics of multi hop localisation algorithms should be evaluated for different topologies. Some of the irregular topologies used for performance evaluation are C, F, H, L and S shaped.

2.5.5 Mobile nodes

Depending upon the application of sensor network, part or all of the nodes may be mobile. Therefore, node mobility should be taken into account while evaluating the performance of localisation algorithms. In fact, some localisation algorithms take advantage of mobility of nodes to overcome certain problems like concave areas, coverage and obstacles.

2.5.6 Obstacles

Obstacles may be present in the localisation environment e.g. in the indoor environment. Presence of obstacles between nodes in the sensor field creates many challenges for a localisation algorithm. For example, a sensor node may not hear from a nearby beacon node due to an obstruction. Similarly, obstruction and obstacles may cause reflection which results in an inaccurate range estimate. Therefore, performance metrics for localisation algorithms may be evaluated with and without obstacles in the sensor field.

2.5.7 Signal type

Radio, acoustic, ultrasound or light signal can be used for range and position estimation by a localisation algorithm. Each of these signals has different propagation characteristics in different types of environments and channels. Therefore, appropriate metrics may be selected for the performance evaluation of a localisation algorithm depending upon the signal used.

2.5.8 Algorithm specific parameters

Some parameters may be specific to the localisation algorithm being evaluated. For example, some localisation algorithms model a sensor network using a graph with the sensor nodes considered as vertices and communication links between nodes as the edges of the graph. In this case, certain performance metrics may be evaluated against the degree of a vertex i.e. number of neighbours of a sensor node.

2.6 Performance Evaluation Metrics

In the following discussion, (x_a, y_a) is the actual position and (x, y) is the estimated position of an arbitrary sensor node. (x_{ia}, y_{ia}) and (x_{ja}, y_{ja}) are the actual positions of sensor nodes i and j and are represented by P_i and P_j respectively. (x_i, y_i) and (x_j, y_j) are their estimated positions which are represented by $\hat{P}_i$ and $\hat{P}_j$ respectively. The distance between the actual positions of sensor nodes i and j is denoted by d_{ij}, and the distance between their estimated positions is denoted by $\hat{d}_{ij}$. In other words, d_{ij} is the actual distance and $\hat{d}_{ij}$ is the estimated distance between sensor nodes i and j. $\mathbf{V}_{ij}$ is the vector between the actual positions of sensor nodes i and j from P_i to P_j. $\hat{\mathbf{V}}_{ij}$ is the vector between the estimated positions from $\hat{P}_i$ to $\hat{P}_j$.

2.6.1 Absolute localisation error

Absolute localisation error for an individual sensor node is defined as the distance between its actual and estimated positions. If (x_a, y_a) is the actual position and (x, y) is the estimated position of a sensor node, then the absolute localisation error e_l is given by

$$e_l = \sqrt{(x - x_a)^2 + (y - y_a)^2}. \tag{2.2}$$

2.6.2 Mean absolute localisation error

If there are N sensor nodes in a sensor field, then the mean absolute localisation error of the N sensor nodes is given by

$$\mu_l = \frac{1}{N}\sum_{i=1}^{N} e_l, \tag{2.3}$$

$$\mu_l = \frac{1}{N}\sum_{i=1}^{N} \sqrt{(x_i - x_{ia})^2 + (y_i - y_{ia})^2}, \tag{2.4}$$

where (x_{ia}, y_{ia}) is the actual position and (x_i, y_i) is the estimated position of an ith sensor node with e_l absolute localisation error.

2.6.3 Normalised localisation error

Normalised localisation error e_{ln} is obtained by dividing the absolute localisation error with the radio range R_s of a sensor node.

$$e_{ln} = \frac{e_l}{R_s}, \tag{2.5}$$

$$e_{ln} = \frac{1}{R_s}\sqrt{(x - x_a)^2 + (y - y_a)^2}. \tag{2.6}$$

This gives localisation error in units of radio range R_s of a sensor node. If a sensor node has an absolute localisation error e_l greater than R_s or normalised localisation error $e_{ln} > 1$, then the sensor node has estimated its position beyond its radio range. This implies that the sensor node is using an estimated position which is beyond its area of coverage.

2.6.4 Mean normalised localisation error

Some sensor nodes may have small while other nodes may have large localisation errors. Mean error of N sensor nodes in the sensor field gives a better estimate of performance of a localisation algorithm. Mean normalised localisation error μ_{ln} of N sensor nodes in a sensor field is given by

$$\mu_{ln} = \frac{1}{N}\sum_{i=1}^{N} e_{lni}, \tag{2.7}$$

$$\mu_{ln} = \frac{1}{NR_s}\sum_{i=1}^{N} e_{li}, \tag{2.8}$$

$$\mu_{ln} = \frac{1}{NR_s}\sum_{i=1}^{N} \sqrt{(x_i - x_{ia})^2 + (y_i - y_{ia})^2}, \tag{2.9}$$

where e_{li} is the absolute localisation error and e_{lni} is the normalised localisation error of an ith sensor node. Mean normalised localisation error only gives central tendency of error. It should be used along with other statistics e.g. median, mode, range and standard deviation to get a better view of the performance of a localisation algorithm. For example, lower value of standard deviation of error implies that majority of sensor nodes have localisation error near the mean value and that the error is predictable. On the other hand, a large value of standard deviation would mean that there is a large variation in estimation error and that the error is not predictable.

2.6.5 Statistics of localisation error

A single statistical measure such as mean error may not completely describe the performance of a localisation algorithm. Therefore, other statistical metrics given below along with the mean value of error may also be used to assess the performance of an algorithm.

2.6.5.1 Maximum localisation error

The maximum value of localisation error gives an indication of the worst performance of localisation algorithm.

$$\begin{aligned} e_{lmax} &= \max_{i=1 \ldots N} e_{li}, \\ e_{lmax} &= \max_{i=1 \ldots N} \sqrt{(x_i - x_{ia})^2 + (y_i - y_{ia})^2}. \end{aligned} \tag{2.10}$$

2.6.5.2 Minimum localisation error

The minimum value of localisation error gives an indication of the best performance of localisation algorithm.

$$\begin{aligned} e_{lmin} &= \min_{i=1 \ldots N} e_{li}, \\ e_{lmin} &= \min_{i=1 \ldots N} \sqrt{(x_i - x_{ia})^2 + (y_i - y_{ia})^2}. \end{aligned} \tag{2.11}$$

2.6.5.3 Median of localisation error

The middle value of localisation errors which are arranged in ascending order is the median of localisation error e_{ld}.

2.6.5.4 Mode of localisation error

The most frequently occurring value of the localisation error is the mode of the localisation error e_{lo}.

2.6.5.5 Standard deviation of localisation error

Standard deviation σ_a of localisation error is given by

$$\sigma_a = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_{li} - \mu_l)^2}, \tag{2.12}$$

where μ_l is the mean absolute localisation error of N unknown sensor nodes.

2.6.5.6 Geometric mean

If the standard deviation of the localisation error is large, it may sometimes be suitable to use the geometric mean.

$$e_g = \left(\prod_{i=1}^{N} e_{li}\right)^{\frac{1}{N}}, \tag{2.13}$$

where e_{li} is the localisation error of ith sensor node.

The mean, median and mode are measures of central tendency. Apart from other details, a comparison of these three statistics of localisation error also gives an insight into the type of distribution followed by the localisation error. For example, in a perfectly symmetric distribution, all the three statistics are equal.

$$\mu_l = e_{ld} = e_{lo}. \tag{2.14}$$

If frequency distribution of localisation error is positively skewed, then

$$\mu_l > e_{ld} > e_{lo}. \tag{2.15}$$

However, if the frequency distribution of localisation error is negatively skewed, then

$$\mu_l < e_{ld} < e_{lo}. \tag{2.16}$$

Furthermore, in a moderately skewed distribution, the distance between the mean and mode is approximately three times the distance between mean and median values i.e.

$$\mu_l - e_{lo} = 3(\mu_l - e_{ld}), \tag{2.17}$$

$$\begin{aligned} e_{lo} &= \mu_l - 3(\mu_l - e_{ld}), \\ e_{lo} &= 3e_{ld} - 2\mu_l. \end{aligned} \tag{2.18}$$

This gives mode of localisation error given the mean and median values. Again, from (2.17), the median value can be calculated given the mean and mode of localisation error.

$$\begin{aligned} \mu_l - e_{ld} &= \frac{1}{3}\mu_l - \frac{1}{3}e_{lo}, \\ e_{ld} &= \mu_l - \frac{1}{3}\mu_l + \frac{1}{3}e_{lo}, \\ e_{ld} &= \frac{2}{3}\mu_l + \frac{1}{3}e_{lo}, \\ e_{ld} &= \frac{1}{3}(2\mu_l + e_{lo}). \end{aligned} \tag{2.19}$$

Similarly, the mean value of localisation error can be calculated given the median and mode provided the frequency distribution is only moderately skewed. From (2.17)

$$\begin{aligned} 3(\mu_l - e_{ld}) &= \mu_l - e_{lo}, \\ 2\mu_l &= 3e_{ld} - e_{lo}, \\ \mu_l &= \frac{1}{2}(3e_{ld} - e_{lo}). \end{aligned} \tag{2.20}$$

Hence, knowing any two of the mean, median and mode, we can determine the third if the distribution is moderately skewed.

2.6.6 Root mean square error

Root mean square (RMS) error can be calculated for absolute or normalised localisation errors. For example, RMS value of the absolute localisation error is given by

$$e_{lrms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_{li}^2}, \tag{2.21}$$

$$e_{lrms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\sqrt{(x_i - x_{ia})^2 + (y_i - y_{ia})^2}\right]^2}, \tag{2.22}$$

$$e_{lrms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[(x_i - x_{ia})^2 + (y_i - y_{ia})^2\right]}. \tag{2.23}$$

RMS value of the normalised localisation error is given by

$$e_{lnrms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_{lni}^2}, \tag{2.24}$$

$$e_{lnrms} = \sqrt{\frac{1}{NR_s}\sum_{i=1}^{N}\left[(x_i - x_{ia})^2 + (y_i - y_{ia})^2\right]}. \tag{2.25}$$

Dividing (2.25) by (2.23), we get e_{lnrms} in terms of e_{lrms}.

$$e_{lnrms} = \frac{e_{lrms}}{\sqrt{R_s}}. \tag{2.26}$$

RMS error amplifies large errors. In calculation of RMS value of error, the error term is squared. Therefore, it gives more weight to larger values of error.

Therefore, high localisation errors of only a few sensor nodes can result in a high RMS value of error.

The relationship between the mean and the RMS values of localisation errors is given as below.

$$e_{lrms}^2 = \mu_l^2 + \sigma_a^2. \tag{2.27}$$

From the above relationship, it is evident that the RMS value of localisation error is always greater than or equal to the mean localisation error.

2.6.7 Cumulative distribution function

The plot of cumulative distribution function (CDF) gives cumulative distribution of sensor nodes having localisation error in a particular range. Therefore, it gives important information about the number of sensor nodes between a certain range of localisation error.

2.6.8 Global energy ratio

Global energy ratio (GER) is a metric proposed and used in [43]. It attempts to quantify the extent to which the layout of the sensor field constructed by the estimated positions matches the original layout. If (x_{ia}, y_{ia}) and (x_{ja}, y_{ja}) are the actual positions of sensor nodes i and j in the sensor field, then the actual distance d_{ij} between them is given by

$$d_{ij} = \sqrt{(x_{ia} - x_{ja})^2 + (y_{ia} - y_{ja})^2}. \tag{2.28}$$

Similarly, if (x_i, y_i) and (x_j, y_j) are the estimated positions of unknown sensor nodes i and j, then the distance $\hat{d}_{ij}$ between their estimated positions in the sensor field reconstructed by the localisation algorithm is given by

$$\hat{d}_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{2.29}$$

The global energy ratio is then given by

$$E_G = \frac{2}{N(N-1)} \sqrt{\sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(\frac{\hat{d}_{ij} - d_{ij}}{d_{ij}} \right)^2}. \tag{2.30}$$

As distance is a quantity measured between two nodes, the total number of distance elements between N sensor nodes is calculated by considering combination of N things taken 2 at a time. Therefore, the total number of distances is $\frac{N!}{(N-2)!2!} = \frac{N(N-1)}{2}$. Square root of the sum of squares of the difference in distances is divided by this number in (2.30) to compute the error. The GER is a global quality metric which attempts to measure the ability of localisation algorithm to reconstruct the layout of the entire sensor field.

2.6.9 Average relative deviation

The authors of [59] devise a new metric to reflect the error in the reconstruction of the layout of the sensor field. The average relative deviation (ARD) is given by

$$E_A = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{\left| \hat{d}_{ij} - d_{ij} \right|}{\min(\hat{d}_{ij}, d_{ij})}. \tag{2.31}$$

The ARD calculates the average of the difference in estimated and actual distances normalised to the shorter of the two distances. In the ARD metric, the difference in the distance term $\hat{d}_{ij} - d_{ij}$ is not squared. Therefore, it gives equal weight to all the errors including the outliers. If the errors are uniformly distributed, it gives a plot similar to that of RMS error [47].

2.6.10 Global distance error

In [60], the authors note that GER does not reflect a true RMS value. Therefore, they modify GER and propose a new metric called global distance error (GDE).

$$E_D = \frac{1}{R_s}\sqrt{\frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(\frac{\hat{d}_{ij} - d_{ij}}{d_{ij}}\right)^2}. \tag{2.32}$$

The GDE calculates the RMS error over the entire network. The result is then normalised to the radio range R_s of a sensor node. Investigation by the authors of [47] also reveal that GER does not perform consistently when the number of nodes in the sensor field is varied. GER decreases as the number of sensor nodes is increased which is not according to the expected results. Therefore, GER is not independent of the number of sensor nodes in the network. The GDE metric, however, provides expected and consistent results according to [47].

2.6.11 Area based metric

A metric based upon the areas of estimated and actual topologies is proposed in [47]. For many localisation algorithms, the number and extent of localisation errors for sensor nodes at the corners and boundaries of the sensor network are higher than nodes at and around the centre. This is due to the reason that the degree of connectivity of the nodes at and around the centre of the sensor network is higher and is uniform all around a sensor node. However, the sensor nodes located at the boundary and corners usually have low connectivity from the side facing the boundary. Therefore, the estimated position may be flipped resulting in a flip ambiguity [47, 61]. In such a scenario, the area ratio better depicts the performance of localisation algorithm. If A_r is the area of the actual topology and A_v is the area of the estimated topology, then the area ratio A_R is given by

$$A_R = \frac{A_v}{A_r}. \tag{2.33}$$

If $A_R \neq 1$, the areas of the estimated and actual topologies are not equal. This implies that the boundary of the estimated topology is not constructed accurately. However, it is to be noted that the areas of the estimated and actual topologies can be equal even if the boundary is not accurately estimated. Therefore, if $A_v = A_r$ so that $A_R = 1$, this does not necessarily mean that the boundary is accurately realised. Hence, A_R gives only a vague impression of the performance of the localisation algorithm being evaluated. When $A_v = A_r$, it merely implies that the areas of the estimated and actual topologies are equal regardless of the accuracy of estimated positions of individual sensor nodes. Therefore, area ratio should be used in conjunction with other metrics for the localisation performance evaluation and analysis.

2.6.12 Sum of distance inconsistencies

The authors in [62] use the sum of the differences between the actual and estimated distances between pairs of sensor nodes as a metric to judge the performance of localisation algorithm. The sum of distance inconsistencies is given by

$$E_S = \sum_{i=1}^{N} \sum_{j=1}^{i-1} \left| d_{ij} - \hat{d}_{ij} \right|, \tag{2.34}$$

where d_{ij} is the measured distance between the actual positions and $\hat{d}_{ij}$ is the distance between the estimated positions of nodes i and j. If the estimated positions of nodes i and j are (x_i, y_i) and (x_j, y_j) respectively, then

$$\hat{d}_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \tag{2.35}$$

so that from (2.34)

$$E_S = \sum_{i=1}^{N} \sum_{j=1}^{i-1} \left| d_{ij} - \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \right|. \tag{2.36}$$

The sum of distance inconsistencies varies with the number of sensor nodes even if the average localisation error remains unchanged. Therefore, this metric may not be suitable for performance comparison of different sizes of sensor networks.

2.6.13 Quality of fit

An accuracy metric for three dimensional sensor networks is defined in [63]. The quality of fit metric measures the average difference between the measured and estimated ranges between pairs of nodes, and is given by

$$E_Q = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(R_{ij} - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \right), \tag{2.37}$$

where R_{ij} is the actual distance and the term under the square root gives the distance between the estimated positions of nodes i and j. As the absolute value of the difference between the estimated and actual distances is not used, some of the differences will be positive and some will be negative. As a result, some of the difference values will cancel out. Therefore, the end result may not correctly depict the average of the difference between the distances.

2.6.14 Frobenius error

Like GER, the Frobenius error proposed and used in [46], captures the error in the construction of the geometry of the sensor field using the estimated positions. The metric uses the node to node distances to compute the error. The Frobenius error for a sensor field with N sensor nodes, is given by

$$E_F = \sqrt{\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\hat{d}_{ij} - d_{ij})^2}. \tag{2.38}$$

It should be noted that unlike GER and ARD, computation of Frobenius error in (2.38) does not require that $i < j$. As a result, each distance between two nodes is used twice in the summation. Hence the total number of distance elements being summed is $N(N-1)$, which is twice the number of total distance elements between all the nodes. However, the summation is divided by N^2.

Hence, the term inside the square root does not represent a mean value.

2.6.15 Performance cost metric

This is a hybrid metric proposed in [60]. The performance cost metric is evaluated by assigning weight γ to global distance error E_D and weight $(1-\gamma)$ to performance cost C, which is the average energy per node required for localisation.

$$E_C = \gamma(E_D) + (1-\gamma)C. \tag{2.39}$$

The weighing factor γ can be assigned any arbitrary value depending upon the relative importance of E_D and C in the sensor network being evaluated.

2.6.16 Manhattan distance

Another metric that has been used in wireless networks is the Manhattan distance [41, 64, 65]. Manhattan distance between two points is the sum of horizontal and vertical distances to get from one point to the other. In other words, it is the sum of distances between two points in a coordinate plane if you move only parallel to the x-axis and y-axis. Therefore, it is not the shortest path between the two points. Manhattan distance d_M between actual position (x_a, y_a) and estimated position (x, y) is given by

$$d_M = |x - x_a| + |y - y_a|. \tag{2.40}$$

In [41], it is proposed that the Manhattan distance can be used for the evaluation of the similarity of original and estimated topologies by calculating its average for the estimated positions of all the nodes.

$$\mu_M = \frac{1}{N}\sum_{i=1}^{N} d_{Mi}, \tag{2.41}$$

$$\mu_M = \frac{1}{N}\sum_{i=1}^{N}(|x_i - x_{ia}| + |y_i - y_{ia}|), \tag{2.42}$$

where (x_{ia}, y_{ia}) is the actual position, (x_i, y_i) is the estimated position of an ith node while d_{Mi} is the Manhattan distance between the actual and estimated positions. Investigation results in [41] show that the metric based upon Manhattan distance is sensitive to rotation, shift and distortion in network topology and shows a linear response against each of these changes. If a sensor node has estimated its position without any error such that $(x, y) = (x_a, y_a)$, then from (2.40) $d_M = 0$. If all the sensor nodes in the network estimate their positions accurately, then from (2.41) and (2.42), $\mu_M = 0$.

2.6.17 Cosine distance

Cosine similarity is used in the information retrieval domain. Authors of [41] propose to adapt this metric for measuring the similarity of original and estimated topologies of sensor network. To further explain, consider a pair of sensor nodes i and j with actual positions P_i and P_j and estimated positions $\hat{P}_i$ and $\hat{P}_j$. Let $\mathbf{V}_{ij}$ be the vector joining the actual positions from P_i to P_j and $\hat{\mathbf{V}}_{ij}$ be the vector joining the estimated positions from $\hat{P}_i$ to $\hat{P}_j$. The cosine similarity is the cosine of the angle θ between the vectors $\mathbf{V}_{ij}$ and $\hat{\mathbf{V}}_{ij}$.

$$\cos\theta = \frac{\mathbf{V}_{ij} \cdot \hat{\mathbf{V}}_{ij}}{|\mathbf{V}_{ij}||\hat{\mathbf{V}}_{ij}|}. \tag{2.43}$$

Using the cosine similarity, the cosine distance d_C between a pair of nodes is defined as

$$d_C = \frac{1 - \cos\theta}{2}. \tag{2.44}$$

A metric μ_C based upon the cosine similarity to measure the difference in original and estimated topologies comprising of N sensor nodes is given by averaging the cosine distances among the nodes.

$$\mu_C = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} d_{Cij}, \tag{2.45}$$

where d_{Cij} is the cosine distance between node i and node j. Cosine similarity measures the similarity in the directions of original and estimated positions of a pair of nodes. As $-1 \leq \cos\theta \leq +1$, the cosine similarity has a range from -1 to $+1$. For example, if the estimated positions are such that the direction of vector between them is exactly the same as the direction between actual positions, then $\theta = 0$ and $\cos\theta = 1$. The corresponding cosine distance is zero. It has been shown in [41], that the cosine distance is insensitive to shift in topology. However, it is sensitive to rotation and shows a logarithmic response to distortion in topology. It is to be noted that if a pair of nodes estimates the positions such that there is only a shift in the positions in the same direction so that the magnitude and direction of $\hat{\mathbf{V}}_{ij}$ remains unaffected, then $\hat{\mathbf{V}}_{ij} = \mathbf{V}_{ij}$, $\theta = 0$, $\cos\theta = 1$, so that from (2.44), $d_C = 0$. Therefore, the cosine distance can be zero even when the sensor nodes have errors in their estimated positions. If all the sensor nodes in the network have the same amount of shift in position in the same direction, then from (2.45), $\mu_C = 0$.

2.6.18 Tanimoto coefficient distance

Tanimoto coefficient is an important similarity metric and is used in a number of applications including text and image matching problems. It is proposed in [41] to use it for measuring the difference in original and estimated topologies of a sensor network. Using the vector $\mathbf{V}_{ij}$ between the actual positions and $\hat{\mathbf{V}}_{ij}$ between the estimated positions of a pair of sensor nodes, the Tanimoto coefficient is defined as:

$$T_C = \frac{\mathbf{V}_{ij} \cdot \hat{\mathbf{V}}_{ij}}{|\mathbf{V}_{ij}|^2 + |\hat{\mathbf{V}}_{ij}|^2 - \mathbf{V}_{ij} \cdot \hat{\mathbf{V}}_{ij}}. \tag{2.46}$$

The Tanimoto distance d_T between the original and estimated topologies of two nodes i and j is defined as:

$$d_T = \frac{1 - T_C}{2}. \tag{2.47}$$

The difference in original and estimated topologies of the sensor network comprising of N sensor nodes is measured by the average Tanimoto distance among the sensor nodes which is given by:

$$\mu_T = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} d_{Tij}, \tag{2.48}$$

where d_{Tij} is the Tanimoto distance between the original and estimated positions of two sensor nodes i and j. The behaviour of Tanimoto coefficient distance against change in topology is similar to that of cosine distance. It is insensitive to shift but is sensitive to rotation in topology. Its response to distortion in topology is logarithmic. Similar to cosine distance, Tanimoto distance does not report shift in the topology. When there is a shift in the estimated positions of a pair of sensor nodes i and j such that the estimated topology is the same as the original topology so that $(x_i, y_i) \neq (x_{ia}, y_{ia})$ and $(x_j, y_j) \neq (x_{ja}, y_{ja})$ but $\hat{\mathbf{V}}_{ij} = \mathbf{V}_{ij}$ and $\theta = 0$, then from (2.46), $T_C = 1$ and from (2.47) $d_T = 0$. Hence, the Tanimoto distance is zero despite error in the estimated position. If all the sensor nodes have the same amount of localisation error with a shift in the same direction, then from (2.48), $\mu_T = 0$.

2.6.19 Relative Euclidean Distance

This metric is proposed and investigated in [41]. Relative Euclidean distance (RED) is based upon the observation that the Euclidean distance alone between the actual and estimated positions of a sensor node does not reflect the change in the relative topology or geometry of sensor nodes with respect to each other. Therefore, the RED metric considers sensor nodes in pairs so as to capture the relative geometry of positions of nodes. To further elaborate the RED metric, consider a topology comprising of a pair of nodes i and j. Let the actual positions of nodes be P_i and P_j and the estimated positions be $\hat{P}_i$ and $\hat{P}_j$. Let $\mathbf{V}_{ij}$ be the vector from P_i to P_j and $\hat{\mathbf{V}}_{ij}$ be the vector from $\hat{P}_{ij}$ to $\hat{P}_{ij}$. Hence, $\mathbf{V}_{ij}$ represents the actual topology of sensor nodes and $\hat{\mathbf{V}}_{ij}$ represents the estimated topology of the nodes. The relative Euclidean distance (RED) between the actual and estimated topologies of the pair of sensor nodes is then equal to the magnitude of the vector connecting the end points of $\mathbf{V}_{ij}$ and $\hat{\mathbf{V}}_{ij}$, and is given by

$$d_R = \left(|\mathbf{V}_{ij}|^2 + |\hat{\mathbf{V}}_{ij}|^2 - 2\mathbf{V}_{ij} \cdot \hat{\mathbf{V}}_{ij} \right)^{\frac{1}{2}}. \tag{2.49}$$

To measure the change in topology of the entire sensor field comprising of N sensor nodes, the average value of RED for all the possible combinations of pairs of sensor nodes is calculated as below:

$$\mu_R = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} d_{Rij}, \tag{2.50}$$

where d_{Rij} is the RED between pair of nodes i and j computed using (2.49). It is to be noted that $\hat{\mathbf{V}}_{ij} = \mathbf{V}_{ij}$ and $\theta = 0$ when the estimated positions are the same as the actual positions i.e. $\hat{P}_{ij} = P_{ij}$. Using this condition in (2.49),

$$d_R = \left(|\mathbf{V}_{ij}|^2 + |\mathbf{V}_{ij}|^2 - 2|\mathbf{V}_{ij}||\mathbf{V}_{ij}| \right)^{\frac{1}{2}} = 0. \tag{2.51}$$

Hence, $\mu_R = 0$ when positions of all the sensor nodes are estimated without error and estimated topology of the sensor field is the same as the actual topology. Like cosine distance and Tanimoto distance, the RED metric remains unaffected by shift if the relative topology among the sensor nodes is not changed. As is obvious from (2.49), $d_R = 0$ if $\hat{\mathbf{V}}_{ij} = \mathbf{V}_{ij}$ and $\theta = 0$.

2.6.20 Cumulative vectorial distance

This is another metric proposed in [41]. For each sensor node, the difference in the x-coordinates of the estimated and actual positions are calculated. Similar difference is calculated for the y-coordinates. The differences in the x-coordinate values of all the sensor nodes are summed together to result in a vector along x-axis starting at the origin. Similarly, the differences in the y-coordinate values are summed together to result in a vector along the y-axis. The cumulative vectorial distance (CVD) is then defined as the distance between the terminal points of the perpendicular vectors averaged over the number of sensor nodes.

$$\mu_V = \frac{1}{N}\Big(\Big[\sum_{i=1}^{N}(x_i - x_{ia})\Big]^2 + \Big[\sum_{i=1}^{N}(y_i - y_{ia})\Big]^2\Big)^{\frac{1}{2}}. \tag{2.52}$$

The distance between the terminal points of the vectors is the same as the magnitude of the resultant vector of the sum of these vectors. It is to be noted that $x_i - x_{ia} \geq 0$ or $x_i - x_{ia} \leq 0$ and $y_i - y_{ia} \geq 0$ or $y_i - y_{ia} \leq 0$ for individual sensor nodes depending upon the estimated coordinate values. Therefore, some of the differences in the summation terms in (2.52) may be positive and some may be negative. This may result in smaller values of the component vectors along x-axis and y-axis. As a result, the CVD metric may not sometimes depict a true picture of the performance of a localisation algorithm. This is also evident from the performance results of CVD in [41] against random distortion in the sensor field.

2.6.21 Spring distance

To evaluate spring distance between the estimated and actual topologies, the sensor network is modelled as an elastic object as proposed in [41]. Actual topology of the sensor network is considered as the original elastic object and the estimated topology is considered as the deformed version. The spring distance is the difference between the potential energies of the original elastic object and the deformed object.

To model the sensor network as an elastic object, three types of springs are assumed. Type 1 springs are tension springs connecting each node to the rest of the nodes in the network. Hence, if there are N sensor nodes in the network, each sensor node is connected through $N-1$ type 1 springs to the rest of the $N-1$ nodes. Hence, there are a total of $\frac{N(N-1)}{2}$ type 1 springs. These springs store potential energy as a result of change in the Euclidean distance. In addition, each node is connected to the ground through a type 2 tension spring. For N sensor nodes, there are N type 2 springs. A type 2 spring stores potential energy due to change of position of an individual node. In addition to a type 1 spring, a pair of sensor nodes is connected to each other through a type 3 torsion spring. A torsion spring stores potential energy due to rotation and is not responsive to expansion or compression. Each sensor node is connected to $N-1$ type 3 springs and there are $\frac{N(N-1)}{2}$ type 3 springs in total. When the sensor nodes are at their actual positions, the springs are relaxed under equilibrium and store no potential energy. However, in the model corresponding to the estimated topology, the springs are displaced and hence store potential energy. This potential energy is a measure of the difference between the estimated and actual topologies of the network. The potential energy stored in a tension spring of length l is $\frac{k_s x_s^2}{2l}$ where x_s is the displacement in the spring and k_s is the elastic modulus of the tension spring. Similarly, the potential energy stored in torsion spring is $\frac{k_t \theta_t^2}{2}$ where θ_t is the angle due to rotation between a pair of sensor nodes and k_t is a constant for torsion spring.

For a pair of sensor nodes i and j, the equilibrium length of the type 1 tension spring is $l = \mathbf{V}_{ij}$. Its deformed length is $\hat{\mathbf{V}}_{ij}$. Therefore, displacement of the spring is $d_s = \Big||\mathbf{V}_{ij}| - |\hat{\mathbf{V}}_{ij}|\Big|$. Hence, the average potential energy stored in a type 1 spring is given by

$$U_{T1} = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{k_d \Big||\mathbf{V}_{ij}| - |\hat{\mathbf{V}}_{ij}|\Big|^2}{2|\mathbf{V}_{ij}|}, \tag{2.53}$$

where k_d is the spring constant for type 1 spring. For the type 2 spring, the displacement is the localisation error e_l which is the distance between the estimated and actual positions. If k_s is the spring constant for a type 2 spring, then the average potential energy stored in a type 2 spring is given by

$$U_{T2} = \frac{1}{N}\sum_{i=1}^{N}\frac{k_s e_{li}^2}{2}, \tag{2.54}$$

where e_{li} is the localisation error for the sensor node i. Similarly, the average potential energy stored in a type 3 torsion spring is given by

$$U_{T3} = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\frac{1}{2}k_t\theta_{tij}^2, \tag{2.55}$$

where θ_{tij} is the angle between the actual and estimated topologies of a pair of sensor nodes i and j. This is measured by the angle between the vectors $\mathbf{V}_{ij}$ and $\hat{\mathbf{V}}_{ij}$, and is given by

$$\theta_{tij} = \cos^{-1}\left(\frac{\mathbf{V}_{ij}\cdot\hat{\mathbf{V}}_{ij}}{|\mathbf{V}_{ij}||\hat{\mathbf{V}}_{ij}|}\right). \tag{2.56}$$

The spring distance between the estimated and actual topologies is given by

$$\mu_s = U_{T1} + U_{T2} + U_{T3}. \tag{2.57}$$

The spring constant k_d for type 1 spring defines the displacement sensitivity of the spring distance. Similarly, spring constant k_s for type 2 spring controls the shift sensitivity and spring constant k_t for type 3 spring controls the rotational sensitivity of the spring distance. By assigning different values to shift sensitivity and rotational sensitivity, four different types of spring distances are defined in [41]. These are named as Spring A, Spring B, Spring

Table 2.1: Four types of spring distance

Spring distance	Shift Sensitivity k_s	Rotational sensitivity k_t
Spring A	0.5	0.5
Spring B	1	0
Spring C	0	1
Spring D	0	0

C and Spring D. Values assigned to shift sensitivity k_s and rotational sensitivity k_t for the four types of the spring distance are given in Table 2.1. The displacement sensitivity is kept constant at $k_d = 1$ for all the four types of spring distance.

2.7 Review

Performance of a localisation algorithm can be evaluated either in general or for its use in a specific application. In the case of a general performance evaluation, the performance evaluation will usually aim at testing the algorithm for a representative and simple strategy. On the other hand, performance evaluation for a specific application may be planned in such a manner so that it focuses more on testing a particular aspect, such as coverage ability of the algorithm. In either case, it may not be possible to evaluate its performance using the entire set of criteria, parameters and metrics that we have discussed in the previous sections. Therefore, a balanced and optimum set of criteria, parameters and metrics should be selected for this purpose. In the case of a general performance evaluation, it is sufficient that a basic and primary set of metrics and parameters are considered so that the general characteristics of the algorithm can be identified and evaluated. However, performance evaluation of an algorithm for deployment in a specific application may consider a more detailed set of parameters and metrics suited to the requirements of the application. For example, it is sufficient that a basic accuracy metric such as absolute localization error or normalized localization error is used for the general performance evaluation as is done in the case of Centroid [66] and CAB [34] localisation algorithms. However, one or more of the topology metrics may be used if the performance of the algorithm is being evaluated for an application where accurate reconstruction of sensor field topology is important as is done in [43]. In particular, the topology metrics are used for the performance evaluation of anchor free localization algorithms where the reconstructed topology may be shifted or rotated [43, 46, 59, 60].

For our proposed anchor based localization algorithm in Chapter 3, we use a basic and general set of metrics for the performance testing and evaluation. Furthermore, we propose three new metrics to test aspects of algorithms which cannot be done using the metrics available in the published literature.

2.8 Summary

In this work we have reviewed the system development life cycle, criteria, parameters and metrics for the performance evaluation of localisation algorithms for wireless sensor networks. Position estimation by sensor nodes in a wireless sensor network using a localisation algorithm results in estimation errors. Therefore, the error characteristics of a localisation algorithm are an important deciding factor in its selection for a sensor network application. The error characteristics are studied during the performance evaluation of a localisation algorithm. In this chapter, we first presented the system development life cycle of a localisation algorithm to highlight the context of performance evaluation. Then the criteria and parameters used for the performance evaluation of localisation algorithms for wireless sensor networks are described. Finally, we describe and discuss in detail various types of metrics used in the literature for the measurement of various aspects of localisation algorithms.

Chapter 3

Ripple Localisation Algorithm

In many applications of industrial wireless sensor networks, sensor nodes need to determine their own geographic position coordinates so that the collected data can be ascribed to the location from where it were gathered. We propose a novel intelligent localisation algorithm which uses variable range beacon signals generated by varying the transmission power of beacon nodes. The algorithm does not use any additional hardware resources for ranging and estimates position using only radio connectivity by passively listening to the beacon signals. The algorithm is distributed, so each sensor node determines its own position and communication overhead is avoided. As the beacon nodes do not always transmit at maximum power and no transmission power is used by unknown sensor nodes for localisation, the proposed algorithm is energy efficient. It also provides control over localisation granularity. Simulation results show that the algorithm provides good accuracy under varying radio conditions.

3.1 Introduction

In this chapter, we present a localisation algorithm which is distributed so that each unknown node can localise itself passively by just listening to beacon nodes. The algorithm employs multiple power levels [32–34] and annular rings around beacon nodes and uses multilateration in sensor nodes to achieve a novel but simple localisation technique. The algorithm is intelligent in two aspects. First, the proposed algorithm enables a node to estimate position in an intelligent manner by using passive information from beacon nodes. Second, it provides sensing intelligence to the sensor network. For example, after estimating positions, the nodes can make many intelligent decisions, such as routing based on location information. The algorithm does not require any extra piece of hardware to estimate range and position. Based on the fact that the algorithm sends out a *ripple* of beacon signals, we call it ripple localisation algorithm (RLA) for convenience of reference. We also show quantitatively that the algorithm is energy efficient compared to localisation techniques which transmit beacon signals at fixed radio range. Approximately 92% of the upper limit of energy efficiency can be attained by using 10 magnetisation levels of transmission power.

This chapter makes four major contributions. First, we present a novel, intelligent, distributed and energy efficient localisation algorithm which gives good localisation accuracy. Second, we give a quantitative analysis of energy efficiency of the proposed algorithm. Third, we implement and simulate our intelligent localisation algorithm with practical irregular radio conditions. Our localisation algorithm is also compared with two related algorithms – Centroid [66] and CAB [34]. Fourth, we propose and use three new metrics, error momentum, degree of location intelligence (DOLI) and localisation efficiency, for the evaluation of localisation algorithms. Simulation results demonstrate that our proposed localisation algorithm achieves localisation error which is much lower than both Centroid and CAB. Time complexity of our ripple localisation algorithm is much lower than that of CAB and only marginally higher than that of Centroid.

The rest of this chapter is organised as follows. We summarise previous related work in Section 3.2. In Section 3.3, we describe the ripple localisation algorithm and analyse its energy efficiency in Section 3.4. Performance metrics are discussed in Section 3.5. We evaluate performance of RLA, compare it with two closely related algorithms and describe and simulation results in Section 3.6. We conclude with Section 3.7.

3.2 Related Work

A number of techniques have been used to solve the problem of localisation in wireless sensor networks. Their details are available in various literature surveys. In this section, we describe a few prominent localisation algorithms related to our work.

A distributed, range free, anchor based localisation technique which uses ring overlapping based on comparison of received signal strength indicator (ROCRSSI) is proposed in [33]. In this algorithm, beacon signals transmitted by a beacon node are received and sampled by neighbour beacon nodes. The neighbour beacon nodes calculate the RSSI of the beacon signal and then broadcast it to the network. In this way, RSSI is propagated in the sensor network. Using RSSI and location information of various beacon nodes, an unknown node constructs a set of annular rings around these beacon nodes. It then determines the area of intersection using a grid scan. The sensor node then localises itself at the centroid of this area. Propagation of RSSI increases communication overhead of the algorithm.

In another distributed algorithm, localisation with mobile anchor using trilateration (LMAT) proposed in [67], a moving beacon node is used to help unknown nodes estimate their positions. A single beacon node moves in a zig zag fashion along a trajectory following the shape of interconnected equilateral triangles in the sensor field. If communication radius of the beacon node is r, authors show that the coverage area is maximum if side of each equilateral triangle is $\sqrt{3}r$. While following the path of interconnected equilateral triangles, the beacon node broadcasts its location when it is at each vertex of a triangle. After an unknown node has received three beacon packets, it can estimate its distances from three reference points and then estimate its own position.

In the simple algorithm proposed in [38] for outdoor unconstrained environment, an unknown node starts by initialising its estimated position to be the entire sensor field. When it receives a beacon message, it assumes an annular ring around the beacon node based upon the RSSI. It then updates its position estimate by performing an intersection between existing position estimate and the annular ring around the node, from which, it received the beacon message. This process is iteratively repeated for all neighbour beacon nodes. When an unknown node has estimated its position, it can help other unknown nodes to localise by acting as a beacon node and sending beacon messages.

In another algorithm proposed in [66], and commonly known as Centroid algorithm, an unknown sensor node determines its connectivity with neighbour beacon nodes and estimates its position at their centroid. If coordinates of neighbour beacon nodes B_1, B_2, B_3, ..., B_k are (x_1, y_1), (x_2, y_2), (x_3, y_3), ... (x_k, y_k), then estimated position (x, y) of the sensor node is given by:

$$(x, y) = (\frac{x_1 + x_2 + ... + x_k}{k}, \frac{y_1 + y_2 + ... + y_k}{k}) \tag{3.1}$$

In concentric anchor beacon (CAB) localisation scheme proposed in [34], beacon nodes use two transmission levels. An unknown node chooses three most distant beacon nodes by calculating the areas of triangles of all possible combinations. Next, it calculates points of intersection of annular rings around these beacons by selecting two beacon nodes at a time. It then isolates valid points of intersection which are points that fall within all the annular rings. The unknown node localises itself at the centroid of the region bounded by the points of intersection.

In [37], a distance vector (DV) based positioning system, called ad-hoc positioning system (APS) is proposed. The algorithm uses multi hop communication for localisation. It is assumed that beacon nodes have the same radio range as the unknown nodes. Authors propose three techniques to obtain distance estimates to beacon nodes using multi hop communication. In DV-hop approach, an unknown node estimates its distance to a beacon node by calculating the number of hops to the beacon node and multiplying with average hop length. In DV-distance propagation, hop-to-hop distances between all nodes connecting the unknown node and beacon node are estimated and propagated to the unknown node. It can then estimate its distance from the beacon node. In Euclidean propagation method, an unknown node estimates its direct distance to a beacon node with the help of distance estimates to its immediate neighbour nodes and their corresponding estimates to the beacon node. When an unknown node has obtained distance estimates to three or more beacon nodes by using any of the three methods, it then estimates its position.

Ad hoc localisation system (AHLoS) proposed in [68] uses settled nodes as beacon nodes. Unknown nodes which are immediate neighbours of beacon nodes can estimate their distances to three or more beacon nodes and estimate their geographic positions. These settled nodes then act as beacon nodes and start broadcasting their positions using their own beacon messages. The remaining unknown nodes can use the new beacon nodes alongside the old ones to estimate their positions. The process continues in an iterative manner until all nodes have settled. Major problem associated with this technique is accumulation of error after each iteration.

Using a centralised processing approach, convex position optimisation (CPE) algorithm proposed in [40] formulates sensor node localisation as an optimisation problem. Linear programming is then used to solve the problem and estimate node positions. Connectivity information of beacon and unknown nodes serves as a constraint. For example, if node A has a communication radius of 10 m and another node B can receive communication from A, then it implies that nodes A and B are within 10 m distance. Similarly, if another node C does not hear from node A, it then lies beyond 10 m distance from node A. Each sensor node stores its connectivity information by maintaining a list of nodes from which it receives communication. This connectivity information serves as a constraint for the node localisation linear program (LP) problem. *Given* the positions of beacon nodes, linear program *finds* positions of unknown nodes *subject to* the connectivity constraints of beacon and unknown nodes. As central processing is used, positions of beacon nodes and connectivity information of all unknown nodes is sent to the central processor. After processing the information, central processor sends estimates back to the unknown nodes.

We have discussed selected localisation algorithms for outdoor unconstrained environment and related to our work in this section. Centroid and CAB algorithms, which are also discussed above, are used for comparison with proposed ripple localisation algorithm in Section 3.6.

3.3 Ripple Localisation Algorithm

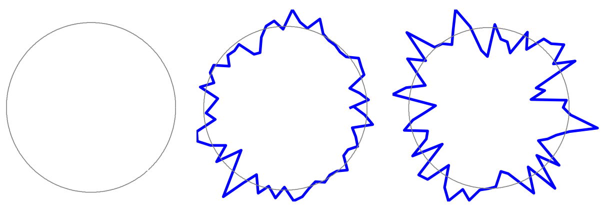
In this section, we start with assumptions of the sensor field and then describe the ripple localisation algorithm. The algorithm consists of two parts – one part is executed by beacon nodes and the other by the unknown nodes.

3.3.1 Sensor Field

We consider an outdoor wireless sensor network in a two dimensional unconstrained sensor field with finite geographic boundaries in which the sensor and beacon nodes are deployed. The sensor nodes are not equipped with any extra piece of hardware to determine range from other nodes. Radio range of unknown nodes is longer compared to their sensing range so that sensing granularity of nodes is higher and sensed data can be transmitted to longer distances. Communication range of beacon nodes is longer than that of unknown sensor nodes. As a result, the beacon signal reaches a large number of unknown sensor nodes at greater distances and a fewer number of beacon nodes are required to localise a large number of unknown sensor nodes. We assume that all nodes are equipped with omnidirectional antennas, designed for sensor networks such as the one described in [69], so that nodes communicate equally in all directions. We also assume that orthogonality of beacon signals is handled by a medium access control protocol. To discuss and explain the algorithm, we assume a perfectly circular radio range. However, for performance evaluation and simulation, we use a more practical irregular radio model [35] as shown in Fig. 3.1. Degree of irregularity (DOI) is used to denote the extent of irregularity in radio pattern and is defined as the maximum radio range variation per unit degree change in the direction of propagation.

3.3.2 RLA Part 1 for Beacon Nodes

In many wireless networks, beacon nodes transmit multiple beacon signals at regular intervals using the same transmission power. As a result, all of these



(a) DOI = 0 (b) DOI = 0.1 (c) DOI = 0.2

Figure 3.1: Irregular radio pattern and degree of irregularity.

beacon signals have the same fixed transmission radius. An example of such a network is the one considered by the Centroid localisation algorithm [66]. In the Centroid algorithm, an unknown node receives multiple beacon signals from each of the reference nodes before it can determine the connectivity metric and estimate its position. In the ripple localisation algorithm, a beacon node transmits beacon signals at different power levels corresponding to different transmission radii so that these radii fall into certain pre-determined quantized intervals[1]. The beacon nodes are tested and calibrated so that transmission radii corresponding to different power levels are recorded for embedding in beacon messages. Hence, unknown nodes receive more information each time they receive a beacon message. The unknown nodes use this information to achieve better accuracy in location estimation.

Transmission of successive beacon signals with incremental values of transmission power and hence different transmission radii is shown in Fig. 3.2. This is analogous to a ripple in water. It emanates from the centre and travels outwards. In the same manner, each beacon node generates a ripple of beacon signals. A beacon node sends its first beacon signal with some set minimum transmission power. For each successive beacon signal it increments the transmission power in such manner that the transmission radius of the beacon signal is longer by a step d_r from the previous beacon signal. Beacon node increments transmission power with each successive beacon signal until maximum transmission power is reached, at which point, the beacon node resets and starts this process all over again. A typical beacon message is shown in Fig. 3.3. In this beacon message, t_0 is the time stamp. (X_b, Y_b)

[1]Many sensor node platforms allow the transmission power to be set dynamically. For example, when using CC2420, an IEEE-802.15.4 compliant RF transceiver, transmission power for each packet can be set using CC2420PacketC.setPower() command under TinyOS.

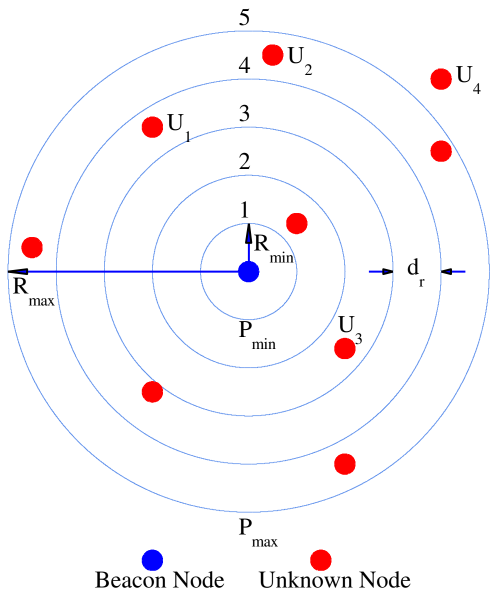


Figure 3.2: A ripple of beacon signals.

ID	t_0	(X_b, Y_b)	P_{ti}	R_i	d_r	R_{min}	R_{max}

Figure 3.3: A typical beacon message.

are the position coordinates of the beacon node. P_{ti} is the transmission power used. R_i is the corresponding radio range. d_r is the beacon signal step. R_{min} is the minimum transmission radius corresponding to the minimum transmission power P_{min}. R_{max} is the maximum transmission radius corresponding to the maximum transmission power P_{max}.

3.3.3 RLA Part 2 for Unknown Nodes

Multiple unknown nodes lying within the communication range of a beacon node receive its beacon signals as shown in Fig. 3.2. By extracting information from all the beacon messages that an unknown node receives from a particular

Algorithm 1 Algorithm for beacon nodes

```
while (true) do
    transmit power = minimum transmit power
    while (transmit power ≤ maximum transmit power) do
        prepare beacon message
        transmit beacon message
        increment transmit power
    end while
end while
```

beacon node, it can determine the radii of the inner and outer circles of the annular ring around the beacon node in which it lies. For example, the first beacon signal that unknown node U_1 receives is beacon signal number 4. Therefore, it can ascertain that outer radius of the annular ring in which it lies is the same as that of beacon signal 4. Note that, of all the beacon messages that unknown node U_1 receives from that particular beacon node, first beacon signal has the smallest radius. Knowing the beacon signal step d_r from the beacon message, and by subtracting it from the radius of the outer circle, it can also determine the radius of the inner circle. Next, it estimates its distance from the beacon node by calculating average of the radii of the inner and outer circles around the beacon node. The unknown node then constructs and solves a set of multilateration equations to estimate its own position.

To further explain position estimation, consider an unknown sensor node having k neighbour beacon nodes with position coordinates (x_1, y_1), (x_2, y_2), ..., (x_k, y_k). The unknown node gets current transmission radius R_i and beacon signal step d_r of a neighbour beacon node from the beacon message and estimates its range r_i from the beacon node as under:

$$r_i = \frac{R_i + R_{i-1}}{2} \tag{3.2}$$

where R_{i-1} is calculated as below:

$$R_{i-1} = R_i - d_r \tag{3.3}$$

R_i and R_{i-1} are radii of the outer and inner circles of the annular ring around the beacon node in which the unknown node lies. Using position coordinates of beacon nodes as centres and range estimates as calculated above as radii, a set of following equations of circles around beacon nodes can be obtained:

$$\begin{bmatrix} (x-x_1)^2+(y-y_1)^2 \\ (x-x_2)^2+(y-y_2)^2 \\ \vdots \\ (x-x_k)^2+(y-y_k)^2 \end{bmatrix} = \begin{bmatrix} r_1^2 \\ r_2^2 \\ \vdots \\ r_k^2 \end{bmatrix} \tag{3.4}$$

Expanding square terms on the left side and rearranging, we get:

$$\begin{bmatrix} x^2+y^2-2x_1x-2y_1y \\ x^2+y^2-2x_2x-2y_2y \\ \vdots \\ x^2+y^2-2x_kx-2y_ky \end{bmatrix} = \begin{bmatrix} r_1^2-x_1^2-y_1^2 \\ r_2^2-x_2^2-y_2^2 \\ \vdots \\ r_k^2-x_k^2-y_k^2 \end{bmatrix} \tag{3.5}$$

Subtracting last row from each of the rows above it:

$$\begin{bmatrix} 2(x_k - x_1)x + 2(y_k - y_1)y \\ 2(x_k - x_2)x + 2(y_k - y_2)y \\ \vdots \\ 2(x_k - x_{k-1})x + 2(y_k - y_{k-1})y \end{bmatrix} \begin{bmatrix} r_1^2 - r_k^2 + x_k^2 - x_1^2 + y_k^2 - y_1^2 \\ r_2^2 - r_k^2 + x_k^2 - x_2^2 + y_k^2 - y_2^2 \\ \vdots \\ r_{k-1}^2 - r_k^2 + x_k^2 - x_{k-1}^2 + y_k^2 - y_{k-1}^2 \end{bmatrix} \tag{3.6}$$

Separating the unknowns (x, y), this can be rewritten in matrix form as below:

$$\begin{bmatrix} (x_k - x_1) & (y_k - y_1) \\ (x_k - x_2) & (y_k - y_2) \\ \vdots & \vdots \\ (x_k - x_{k-1}) & (y_k - y_{k-1}) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2} \begin{bmatrix} r_1^2 - r_k^2 + x_k^2 - x_1^2 + y_k^2 - y_1^2 \\ r_2^2 - r_k^2 + x_k^2 - x_2^2 + y_k^2 - y_2^2 \\ \vdots \\ r_{k-1}^2 - r_k^2 + x_k^2 - x_{k-1}^2 + y_k^2 - y_{k-1}^2 \end{bmatrix} \tag{3.7}$$

Using matrix notation, this can be written as:

$$\mathbf{Az} = \mathbf{R} \tag{3.8}$$

where $\mathbf{z} = \begin{bmatrix} x & y \end{bmatrix}^T$ and

$$\mathbf{A} = \begin{bmatrix} (x_k - x_1) & (y_k - y_1) \\ (x_k - x_2) & (y_k - y_2) \\ \vdots & \vdots \\ (x_k - x_{k-1}) & (y_k - y_{k-1}) \end{bmatrix} \tag{3.9}$$

$$\mathbf{R} = \frac{1}{2} \begin{bmatrix} r_1^2 - r_k^2 + x_k^2 - x_1^2 + y_k^2 - y_1^2 \\ r_2^2 - r_k^2 + x_k^2 - x_2^2 + y_k^2 - y_2^2 \\ \vdots \\ r_{k-1}^2 - r_k^2 + x_k^2 - x_{k-1}^2 + y_k^2 - y_{k-1}^2 \end{bmatrix} \tag{3.10}$$

Using least squares approximation, we get the following closed-form and unique solution to (3.8).

$$\mathbf{z} = \mathbf{A}^{+}\mathbf{R} \tag{3.11}$$

where

$$\mathbf{A}^{+} = (\mathbf{A^T A})^{-1}\mathbf{A^T} \tag{3.12}$$

and is pseudoinverse of matrix $\mathbf{A}$.

Algorithm 2 Algorithm for unknown nodes

```
/* Step 1 */
construct a list of neighbour beacon nodes
/* Step 2 */
for all neighbour beacon nodes do
    sort beacon signal radii of received signals
    signal with smallest radius is the outer circle
    radius of inner circle = radius of outer circle – signal step
    estimated range = average of radii of inner and outer circles
end for
/* Step 3 */
estimated position = multilaterate using all neighbour beacon nodes
```

Number of beacon nodes in a sensor field is only a small percentage of the total number of unknown sensor nodes N. Beacon nodes are an expensive resource in terms of both size and energy as these are usually equipped with some location finding device such as a GPS receiver. Furthermore, the beacon nodes are distributed across the entire sensor field. Hence, an unknown sensor node listens only to a fraction of the limited number of beacon nodes which are in its neighbourhood. Therefore, we do not limit the number of neighbour beacon nodes k in the ripple localisation algorithm for unknown sensor nodes. In fact, some unknown sensor nodes may not be able to localise due to insufficient number of neighbour beacon nodes. This depends upon the coverage ability of a localisation algorithm and is measured by our proposed localisation efficiency metric. For practical deployments, a limit on the maximum number of neighbour beacon nodes used by an unknown node for location estimation may be implemented depending upon the memory resources of the node.

3.3.4 Proof of Concept

To give a proof of the basic concept used in ripple localisation algorithm, we perform a simple experiment which can be easily replicated. We use D-Link DIR-605 IEEE 802.11 wireless router operating at 2.4 GHz and a Huawei Ascend Y300 Android smartphone for our experiment. The wireless router allows its transmission power to be set at different levels through its administrative interface and is equipped with omnidirectional antennas. The WiFi Analyzer application is downloaded and installed on the smartphone from the Google Play Store. We place the wireless router in the centre of a large and unobstructed open field. The power is supplied by an uninterrupted

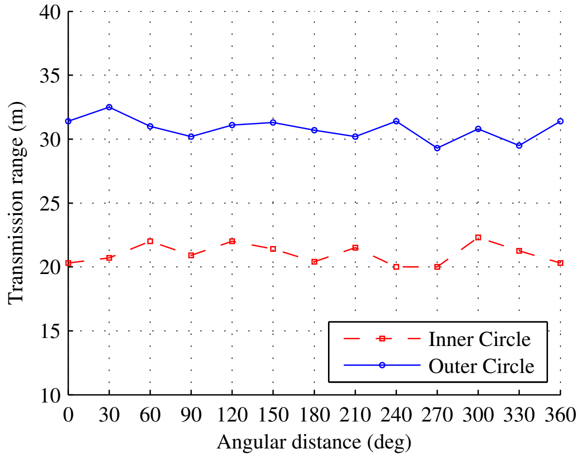


Figure 3.4: Experimental verification of radio connectivity.

power source. Using the administrative interface, we set the transmission power of the wireless router to 15% which is the minimum level it allows. The smartphone is moved away from the wireless router until the WiFi Analyzer on the smartphone reads -95 dBm. This is the threshold power for which the smartphone provides connectivity. Further decrease in RSS results in loss of connectivity. The distance between the router and the smartphone is recorded. We repeat the same experiment every 30^o making a total of 12 recordings circling 360^o around the wireless router. We then set the transmission power of the wireless router to 35% which is the next level that it allows. We repeat the experiment described above and take another set of 12 readings each 30^o apart. We plot both sets of data in Fig. 3.4. The first set of data gives the inner circle corresponding to 15% transmission power whilst the second set of data constitutes the outer circle corresponding to 35% transmission power. It can be seen that the radio coverage is nearly circular. It is also evident that practical irregularity in radio is comparable to DOI used in our simulation experiments.

In the second part of our experiment, we download and install the WiFi Alarm application on the smartphone and set an alarm to sound when the WiFi network provided by the wireless router is detected. We place the smartphone in the area bounded by the transmission circles corresponding to 15% and 35% power levels. We first set the transmission power to 15%. As the smartphone is not within range, the alarm does not sound. Next, we set the transmission power to 35%. The smartphone receives and detects the wireless signal and sounds the alarm. This is repeated by placing the smartphone at 10 different positions in the annular region bounded by the two transmission circles. The alarm did not sound only once when the smartphone was placed at the outer periphery of the annular region. Experiment verifies the basic concept used in the ripple localisation algorithm.

3.3.5 Algorithm Complexity

The proposed localisation algorithm is not computationally expensive. Part 1 of the algorithm executed by the beacon nodes has a linear time complexity $O(k)$. Similarly, the second part executed by the unknown nodes also has linear time complexity $O(k)$ with respect to both the number of beacon nodes and the number of beacon signals in a ripple. As a result, the processing energy required by the algorithm is also small.

Like any practical algorithm, ripple localisation algorithm has certain limitations. The proposed algorithm might not work as good in the indoor environment as in an outdoor environment. In an outdoor environment, signal propagation is fairly circular as is concluded in [38] and [66] and also verified by our experiment. In an indoor environment, shadowing, reflection, absorption and noise would severely affect the circular radio pattern and result in low localisation accuracy.

3.4 Energy Efficiency

We give a quantitative analysis of the energy efficiency of the proposed ripple localisation algorithm in this section. We show that a beacon node achieves 92% of the upper limit of energy efficiency with 10 quantization levels of transmission power. The unknown sensor node expends *zero* transmission energy for localisation.

We assume that the relationship between transmitted power P_t and received power P_r between two nodes in the outdoor unconstrained sensor field is governed by the following path loss model:

$$\frac{P_t}{P_r} = Kd^{\alpha} \tag{3.13}$$

where

$$K = \frac{1}{G_t G_r}(\frac{4\pi}{\lambda})^2 \tag{3.14}$$

G_t and G_r are gains of transmitter and receiver antennas respectively, λ is the wavelength of radio waves, d is the distance between transmitter and receiver antennas, and α is the path loss exponent.

Let us now successively increment the transmitted power from its minimum value P_{min} to maximum value P_{max} corresponding to beacon signal minimum radio range R_{min} and beacon signal maximum radio range R_{max} respectively so as to generate a ripple of beacon signals as shown in Fig. 3.2. The power is increased in such a manner that with each increment of power, increase in beacon signal radio range remains the same i.e. the difference between radii of two consecutive beacon signals remains constant. We call this beacon signal step and denote it by d_r. Furthermore, let us also assume that beacon signal minimum radio range R_{min} is equal to beacon signal step d_r for simplicity. Let the transmitted power of an ith beacon signal be denoted by P_{ti} and the corresponding radio range of the beacon signal be R_i. As the difference between the radii of two consecutive beacon signals d_r is constant, therefore

$$R_i = i \times d_r \tag{3.15}$$

If the total number of beacon signals in the ripple generated by the beacon node is n, then

$$R_{max} = n \times d_r \tag{3.16}$$

According to (3.13), transmitted power P_{ti} for ith beacon signal is given as:

$$\frac{P_{ti}}{P_r} = KR_i^{\alpha} \tag{3.17}$$

Similarly, maximum transmitted power P_{max} is given by

$$\frac{P_{max}}{P_r} = KR_{max}^{\alpha} \tag{3.18}$$

Received power P_r is the same in (3.17) and (3.18). Substituting (3.15) in (3.17) and (3.16) in (3.18), we get:

$$\frac{P_{ti}}{P_r} = K(id_r)^{\alpha} \tag{3.19}$$

$$\frac{P_{max}}{P_r} = K(nd_r)^\alpha \tag{3.20}$$

Dividing (3.19) by (3.20), we get:

$$P_{ti} = (\frac{i}{n})^\alpha P_{max} \tag{3.21}$$

The above relation gives us the transmission power required to transmit ith beacon signal with radius R_i in a ripple. To get the upper bound on the energy saved, we use $\alpha = 2$. Therefore, total power P_T transmitted by a beacon node for sending a ripple of n beacon signals is given by:

$$P_T = \frac{P_{max}}{n^2} \sum_{i=1}^{n} i^2 \tag{3.22}$$

Summation term on the right is the sum of squares of first n natural numbers, which is given by:

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \tag{3.23}$$

Substituting this in (3.22), we get

$$P_T = \frac{(n+1)(2n+1)}{6n} P_{max} \tag{3.24}$$

If a beacon node transmits 5 beacon messages, all at maximum power, transmit power used is $5P_{max}$. However, if a ripple of 5 beacon messages is transmitted by varying the transmit power, so that $n = 5$, the total transmitted power, as calculated using (3.24) is $2.2P_{max}$, which is less than half of the power required to transmit usual beacon messages at maximum power. Power[2] saved is $5P_{max}$ – $2.2P_{max} = 2.8P_{max}$ and energy efficiency of $100 \times 2.8/5 = 56\%$ is achieved. In

[2]Time required to transmit beacon signals in both cases is the same. Therefore, power implies energy and vice versa.

general, transmit power saved P_S in transmitting a ripple of n beacon signals is given by:

$$P_S = nP_{max} - \frac{(n+1)(2n+1)}{6n} P_{max} \tag{3.25}$$

This can be simplified to arrive at the following result:

$$P_S = \frac{(4n+1)(n-1)}{6n} P_{max} \tag{3.26}$$

This gives us the energy saved when a ripple of beacon messages is sent instead of sending beacon signals at fixed radio range. If n beacon messages are transmitted at fixed power, the transmitted power used is nP_{max}. Percentage of power saved or energy efficiency η_P achieved is given by:

$$\eta_P = \frac{P_S}{nP_{max}} \times 100 = \frac{(4n+1)(n-1)}{6n^2} \times 100 \tag{3.27}$$

Note that for $n = 1$, i.e. beacon messages with only one power level, (3.26) and (3.27) result in zero implying that no energy is saved. For $n = 5$, η_P is 56% which is the same as calculated earlier using (3.24). A plot of (3.27) for the interval $0 \leq n \leq 10$ is shown in Fig. 3.5. As can be seen, greater the number of beacon signals n, greater is the energy saved. In the limit, when a beacon node transmits an infinite number of beacon signals, maximum energy efficiency η_{Pmax} is achieved and is given by:

$$\eta_{Pmax} = \lim_{n \to \infty} \frac{(4 + \frac{1}{n})(1 - \frac{1}{n})}{6} \times 100 = 66.67\% \tag{3.28}$$

This shows that the upper bound on the energy saved by a beacon node is 66.67% when the number of beacon signals in a ripple approaches ∞.

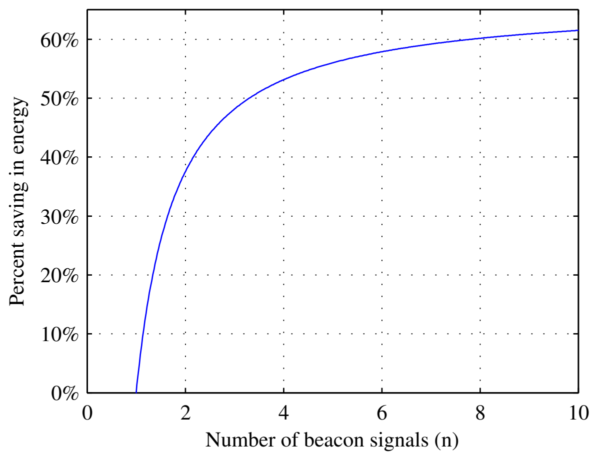


Figure 3.5: Energy saving with increase in beacon signals in a ripple.

Using (3.27), we calculate that for 60% and 65% energy saving, the number of beacon signals in a ripple is approximately 8 and 30 respectively. For $n = 10$ in a ripple, we get 61.50% energy saving i.e. we attain approximately 92% of the upper limit of energy efficiency.

The above calculations show the energy that is being saved by beacon nodes only. The proposed algorithm does not require an unknown node to transmit anything. It estimates the position *passively* by merely receiving and processing information from beacon nodes. Therefore, unknown nodes utilise *zero* transmission energy for the purpose of localisation. We assert that the only energy an unknown node expends for localisation is the processing energy. However, note that a complete assessment of total energy consumption of a sensor node can only be made after analysis of all the related information. This includes energy spent on receiving and how duty cycling is done for the particular application for which the sensor network has been deployed. It is further added that we are concerned only with energy used by the localisation algorithm and not the overall energy used by a sensor node for various other tasks.

3.5 Proposed Performance Metrics

To evaluate performance of the proposed algorithm, it is simulated using MATLAB. In this section, we describe metrics used for performance evaluation and comparison of ripple localisation algorithm. We also propose our own three novel metrics – error momentum, localisation efficiency and degree of location intelligence. We later use them for the performance evaluation of localisation algorithms. The error momentum is a hybrid metric which measures the effectiveness with which a localisation algorithm trades localisation error with localisation time. Localisation efficiency is a metric that measures the coverage ability of a localisation algorithm. Degree of location intelligence (DOLI) assigns a number between 0 and 1 to a sensor node depending upon the level of accuracy of its estimated position. To the best of our knowledge, no previous metrics in the published literature are available to measure these aspects of a localisation algorithm.

We evaluate performance of the ripple localisation algorithm for all the criteria discussed in Chapter 2 – accuracy, cost, coverage, scalability and robustness. As the algorithm is anchor based and it is a general performance evaluation which is not aimed at any specific application, we employ a basic set of accuracy metrics for this purpose. Cost is evaluated in terms of localisation time. Coverage is tested using the localisation efficiency metric. We evaluate performance against scalability by varying the size of the sensor field and hence density of the sensor nodes. Robustness of the algorithm is tested by varying DOI, the beacon signal step and number of beacon signals.

3.5.1 Localisation error

Localisation error is the distance between actual and estimated positions. If a sensor node having actual position coordinates (x_s, y_s) estimates its position to be at (x, y), then localisation error is given by:

$$e_l = \sqrt{(x - x_s)^2 + (y - y_s)^2} \tag{3.29}$$

Localisation error is normalised to sensor node radio range R_s for the purpose of uniform comparison of results.

3.5.2 Localisation time

Localisation time is the time taken by a localisation algorithm for position estimation of a sensor node. Localisation time depends upon computational complexity of the localisation algorithm and is also an indicator of the amount of energy used by the algorithm.

3.5.3 Error momentum

Error momentum is a hybrid metric. Higher localisation time implies higher amount of processing and hence higher expenditure of energy. To quantify the combined effect of localisation time and localisation error, we introduce a new metric which we call *time momentum of localisation error* or simply *error momentum*. We define error momentum as the product of localisation error and localisation time. If t is the time taken by a node for localisation and e_l is the resulting localisation error, then the error momentum e_m is given by

$$e_m = e_l \times t. \tag{3.30}$$

If localisation time of 1 second results in a localisation error of 1 meter, we have unit error momentum of 1 meter-second. As localisation times are usually in the range of milliseconds, meter-second is a large unit for our purpose and we, therefore, use milli meter-second instead.

We further explain error momentum with the help of its two components i.e. localisation error and localisation time. Localisation error is a measure of accuracy of a localisation algorithm. Lower error implies higher accuracy and vice versa. Similarly, localisation time is a measure of the cost of localisation. We buy localisation accuracy with localisation time. Therefore, higher amount of spent time should result in higher accuracy i.e. lower error. Higher the localisation time, higher is the amount of processing and higher is the amount of energy used. Multiplying cost i.e. localisation time with the output of algorithm i.e. localisation error for a given position we get a combined view of the performance of the algorithm in terms of cost and accuracy. Higher value of error momentum implies that either localisation time or localisation error or both have high values. On the other hand, a lower value means value of either or both of these parameters is small. Error momentum is further explained with the help of Table 3.1.

Table 3.1: Error momentum

Localisation error	Localisation time	Error momentum
LOW	LOW	LOW
LOW	HIGH	Unknown
HIGH	LOW	Unknown
HIGH	HIGH	HIGH

At one extreme both localisation error and localisation time have low values resulting in a low error momentum. This is desirable result under most circumstances. At the other extreme, localisation error and localisation time have high values resulting in high error momentum. This is usually not desirable as localisation error is high despite using higher cost in terms of localisation time. Between these two extremes, values of localisation error and time might be low or high and the resulting error momentum may lie anywhere between the extreme values. If localisation error or time is appropriately traded off, value of error momentum will be more closer to the low value. A localisation algorithm may buy better location accuracy i.e. reduction in localisation error by spending more localisation time. For example, a localisation algorithm may combine a number of techniques using more time to achieve better location accuracy. However, more time means more expenditure of energy. Therefore, location accuracy may be traded off with time to conserve energy. As localisation error is a measure of location accuracy, error momentum can be used to compare localisation algorithms for effective use of time and accuracy trade off.

3.5.4 Localisation Efficiency

This metric measures the coverage ability of a localisation algorithm. A localisation algorithm may some times be not able to help all the unknown nodes localise and settle down. For example, if an unknown nodes does not have more than two neighbour beacon nodes, it may not be able to estimate its location. These nodes are called *unsettled* nodes. The nodes which are able to localise are called *settled* or *location intelligent* nodes. If number of settled nodes is represented by N_s and number of total unknown nodes by N, we can define localisation efficiency η_l as following:

$$\eta_l = \frac{N_s}{N} \times 100. \tag{3.31}$$

Localisation efficiency quantifies the ability of a localisation algorithm to help unsettled nodes in estimating their positions.

3.5.5 Degree of Location Intelligence

This metric assigns a value between 0 and 1 to a sensor node based upon the degree of accuracy of its estimated position. A sensor node which does not know its position is a dumb node or unknown node. When it has estimated its position with the help of a localisation algorithm, it has acquired *location intelligence*. Degree of location intelligence (DOLI) of a sensor node depends upon the extent of accuracy with which it has estimated its position. An unknown or dumb node which does not know its position has a DOLI of 0. A node which knows its exact position with zero estimation error has a DOLI of 1. We represent DOLI using Greek symbol δ and formally define it as:

$$\delta = \begin{cases} 0, & \text{if } \frac{e_l}{R_s} \geq R_s \\ \frac{R_s - \frac{e_l}{R_s}}{R_s}, & \text{otherwise,} \end{cases} \tag{3.32}$$

where R_s is the radio range and e_l is the localisation error of the unknown sensor node. If an unknown node has normalised localisation error greater than or equal to its radio range, $\delta = 0$ for such cases and for all other cases $0 \leq \delta \leq 1$.

A drawback of the proposed metrics is that each of these metrics evaluates performance of a localisation algorithm for a single parameter only. The error momentum evaluates ability of an algorithm to trade off accuracy with localisation time, localisation efficiency evaluates coverage and DOLI evaluates localisation accuracy. A localisation algorithm should be evaluated using different sets of metrics to get a complete picture of its performance.

3.6 Performance Evaluation

We begin this section with a description of simulation settings. We then present, describe and analyse results of experiments carried out for the performance evaluation of the ripple localisation algorithm (RLA) and its comparison with two other algorithms – Centroid [66] and CAB [34]. Both Centroid and CAB are designed for unconstrained environment and are closely related to our work. In particular, CAB also uses multiple power levels. Moreover, both the compared algorithms are described in detail in literature, and therefore, they can be easily implemented and simulated.

A square sensor field of size 100 $m \times 100$ m with 100 randomly deployed sensor nodes is used for the simulation experiments. Transmission power of beacon nodes is changed such that they have a minimum 10 m and maximum 100 m transmission radius with a beacon signal step of 10 m. As a result, the beacon nodes use 1 to 10 transmission power levels. A practical irregular radio model as depicted in Fig. 3.1 is used for performance evaluation. Degree of irregularity (DOI) is used to denote the extent of irregularity and noise in radio pattern and is defined as the maximum radio range variation per unit degree change in the direction of propagation. To simulate irregular radio range of a beacon node, we use a Gaussian random variable with mean (μ) equal to the current transmission radius (R_i) of the beacon node and standard deviation (σ) equal to the product of DOI and mean. In other words, $\mu = R_i$ and $\sigma = \mu \times DOI$. For performance evaluation and comparison, number of beacon signals in a ripple, number of beacon nodes in the sensor field and DOI of beacon signals are varied in a number of simulation experiments and results are recorded and plotted. We use a computer with Intel Core i3-3110M CPU @2.40 GHz processor and 4 GB RAM to run the simulations. We use the same platform to simulate all the compared algorithms.

3.6.1 Localisation Error

Localisation error is a measure of accuracy of localisation. The lower the localisation error, the better is the estimated position with a higher localisation accuracy. We record localisation error in simulation experiments with the results described below.

3.6.1.1 Localisation granularity and number of beacon signals in a ripple

We vary the number of waves in the ripple i.e. the number of beacon signals from 1 to 10 and record the localisation error. The number of the beacon nodes used is 20%. Results are plotted in Fig. 3.6. As more and more beacon signals are added in the ripple, the width of annular ring becomes smaller. As a result, the estimate of the range between an unknown node and a beacon node, calculated as the average of radii of inner and outer circles of annular ring, becomes better thereby resulting in a smaller localisation error. This gives ripple localisation algorithm ability to control granularity of position estimation. Smaller number of beacon signals in the ripple implies coarse granularity and higher number of beacon signals in the ripple implies finer granularity. For example, in simulation result shown in Fig. 3.6, under adverse radio conditions with DOI $= 0.2$, mean error is below $0.75R_s$ when $n = 5$, and approximately $0.5R_s$ when $n = 10$. Therefore, depending upon application of deployed sensor network, localisation granularity can be controlled by varying the number of the beacon signals in a ripple. The higher the number of beacon signals n, the finer the location granularity.

3.6.1.2 Number of beacon nodes

Under adverse radio conditions, with DOI $= 0.2$, localisation error is recorded while number of beacon nodes in the sensor field is varied from 3% to 30%. Results, plotted in Fig. 3.7, show that RLA performs better than both Centroid

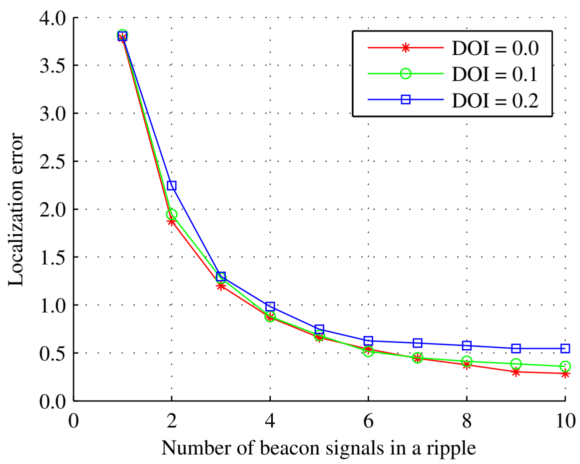


Figure 3.6: Effect of number of beacon signals in a ripple on localisation error.

and CAB over the entire range of the number of beacon nodes. At 10% beacon nodes, RLA has a localisation error of approximately $0.75R_s$, whereas it is approximately $3.75R_s$ for Centroid and $1.45R_s$ for CAB. Addition of beacon nodes beyond 15% does not result in significant improvement in localisation accuracy of any of the three localisation algorithms. Better performance of RLA can be attributed to three other reasons in addition to its usage of multiple power levels. First, the only source of error in RLA is due to least squares approximation, which in turn is due to error in estimation of distances. In the case of CAB, there are two sources of error – estimation of distances in terms of concentric rings and estimation of node position at the centroid of intersected region which is only a guess. Second, RLA constructs a mathematical model and solves a set of equations for position estimation. Third, instead of making a selection, it uses all neighbour beacon nodes for localisation of sensor node resulting in a better position estimate.

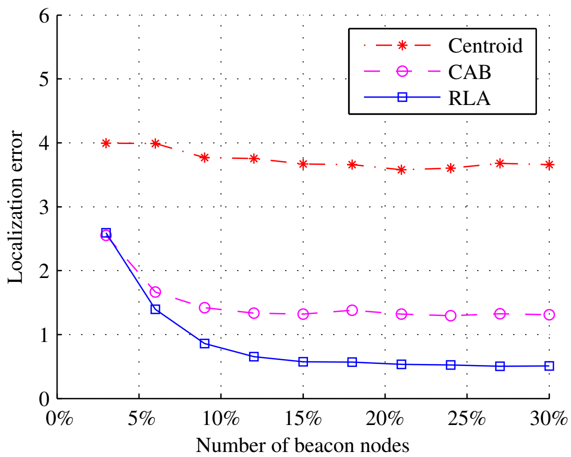


Figure 3.7: Localisation error of Centroid, CAB and RLA.

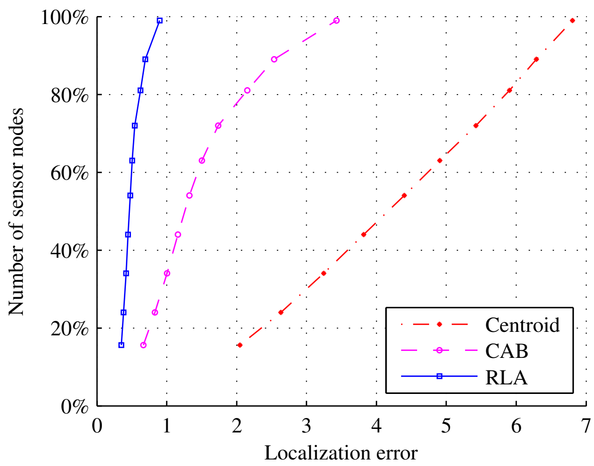


Figure 3.8: Comparison of cumulative error distribution.

3.6.1.3 Cumulative error distribution

CDF of the localisation error of the three compared algorithms is plotted in Fig. 3.8 for DOI = 0.1 using 10% beacon nodes in the sensor field. There are four observations that we can make from these plots. First, there is a large difference in the error distribution of the three algorithms. Second, the error distribution of Centroid is spread over large values. When using Centroid, almost 50% nodes have localisation error between $2R_s$ and $4.5R_s$, and remaining 50% have error between $4.5R_s$ and $7R_s$ which are quite large values. Third, CDF plot for RLA algorithm is relatively vertical compared to the other two algorithms. It means that there is comparatively smaller spread in localisation error and accuracy of localisation can be predicted with more certainty in the case of RLA algorithm. 100% nodes are able to localise with error below R_s and approximately 90% nodes have error below $0.75R_s$ using RLA algorithm. Fourth, in the case of CAB algorithm, approximately 40% nodes have localisation error below R_s. All other nodes using CAB have localisation error greater than R_s. Localisation errors above R_s imply that the node has localised itself beyond its area of radio coverage and the estimated position may lie in the area of coverage of another sensor node.

3.6.2 Localisation Time

Time taken by a node to localise itself is a measure of the computational complexity of the localisation algorithm. It is also a measure of energy used for localisation as higher localisation time implies more processing time and hence higher consumption of energy.

3.6.2.1 Number of beacon signals in a ripple

Change in localisation time with increase in number of beacon signals for RLA algorithm is reflected in Fig. 3.9. There is a linear increase in localisation time implying time complexity of $O(k)$ with respect to the number of beacon signals in a ripple. This linear increase is independent of degree of radio irregularity, as there is only marginal difference in localisation time for different values of

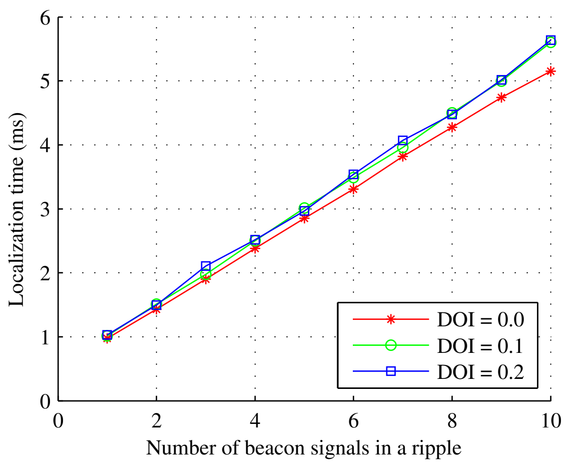


Figure 3.9: Effect of number of beacon signals in a ripple on localisation time.

DOI. This implies that localisation time of RLA is robust to changes in radio pattern and can be more accurately predicted for a given number of beacon nodes.

3.6.2.2 Number of beacon nodes

In Fig. 3.10, we compare average localisation times of Centroid, CAB and RLA algorithms when DOI = 0. The plot in Fig. 3.11 adds a third axis and shows

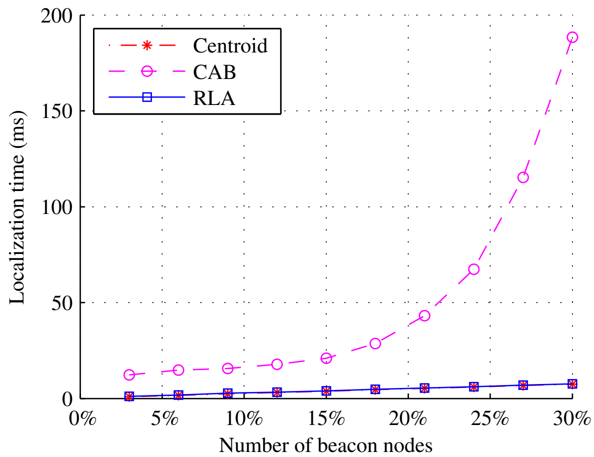


Figure 3.10: Comparison of localisation times.

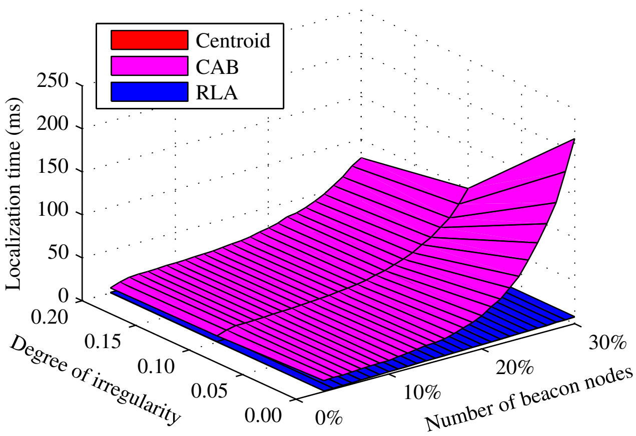


Figure 3.11: Change in localisation time with DOI and number of beacon nodes.

change in localisation time with change in DOI and number of beacon nodes. Localisation time of CAB algorithm causes large values along vertical axis and the marginal difference in localisation times of Centroid and RLA is not visible in Fig. 3.10. Plot for Centroid is not visible and overlapped by RLA curve in Fig. 3.11 due to the same reason. Therefore, localisation times of only Centroid and RLA are plotted in Fig. 3.12. It is evident that the localisation times of Centroid and RLA vary linearly with change in the number of beacon nodes and have linear computational complexity $O(k)$ with respect to neighbour beacon nodes. The average localisation times of both these algorithms are almost identical and much shorter than the average localisation time required by the CAB algorithm. The time required by CAB algorithm increases rapidly in a nonlinear fashion with an increase in the number of the beacon nodes. As is shown in Fig. 3.7, localisation error reduces for all three algorithms as the number of beacon nodes is increased. However, in the case of CAB, this reduction is at the cost of higher localisation time and processing energy. CAB requires repetitive and extensive computation for the selection of three most distant neighbour beacon nodes and calculation and isolation of valid points of intersection. This results in longer localisation time and processing energy. On the other hand, RLA gives better location accuracy without any significant increase in time, as it does not need repetitive computation for position estimation. It is to be noted that, while CAB requires significantly higher amount of processing energy, it needs the same amount of transmission energy as used by RLA when transmitting beacon signals using the same number of power levels.

3.6.2.3 Cumulative time distribution

In. Fig. 3.13, we plot cumulative time distributions of Centroid, CAB and RLA for DOI = 0 with $n = 10$ using 20% beacon nodes. CDF curves for Centroid and RLA algorithm are almost identical and vertical. This means that there is only a small variance in localisation times of sensor nodes and almost all nodes take the same amount of time for localisation using these algorithms. However, in the case of CAB, localisation time required by sensor

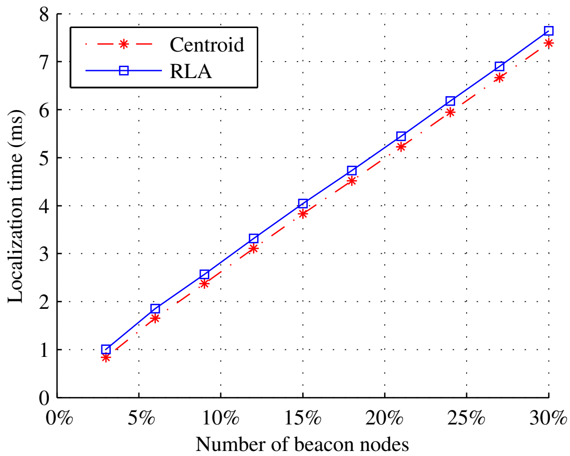


Figure 3.12: Localisation times of Centroid and RLA.

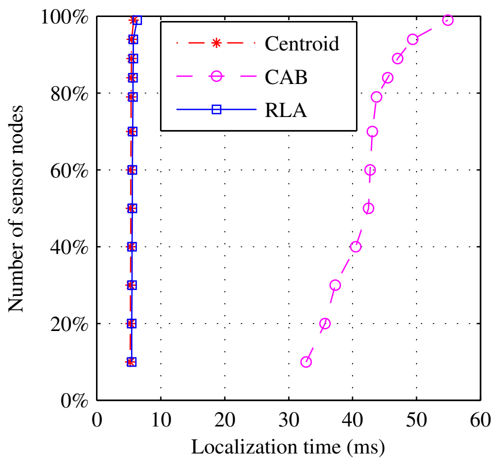


Figure 3.13: Comparison of cumulative time distribution.

nodes varies from 32 to 55 milliseconds – 6 to 10 times larger than the time required by either Centroid or RLA. Lower localisation time implies that the sensor node uses smaller amount of energy for localisation and the localisation algorithm is energy efficient. Smaller variation in localisation time implies that the sensor network reaches settled state in shorter time and energy is also conserved. Furthermore, due to small deviation, it is easier to predict the probable localisation time at the deployment of sensor nodes which can help in scheduling other tasks for the sensor node.

3.6.3 Error Momentum

In Fig. 3.10 three algorithms are compared with respect to localisation time, and in Fig. 3.7, the same algorithms are compared with respect to localisation error. In terms of localisation error, CAB algorithm performs better than the Centroid. However, with respect to localisation time, performance of Centroid algorithm is better than CAB. We can use error momentum to determine which of the two algorithms makes better use of localisation time and error trade off.

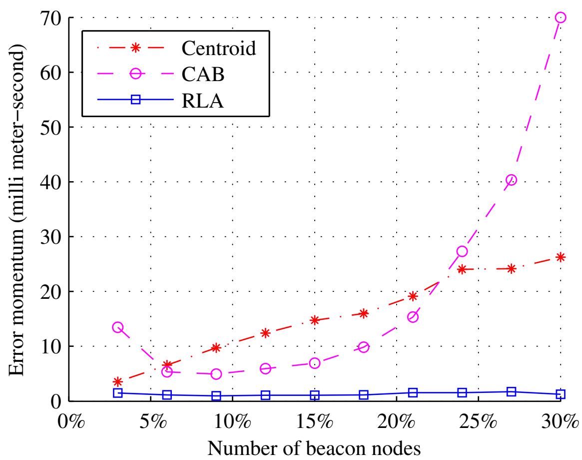


Figure 3.14: Performance comparison using error momentum.

In Fig. 3.14, we plot error momentum of the three algorithms as the number of beacon nodes in the sensor field is increased. Fig. 3.15 depicts change in error momentum with DOI on the third axis. CAB performs better when the number of beacon nodes is between 5% to 23%. However, Centroid algorithm makes better use of time and error trade off when the number of beacon nodes is below 5% or is greater than 23%. RLA performs better than both Centroid and CAB. It has lower error momentum than the other two algorithms. In addition, the error momentum remains almost constant for the entire range of beacon nodes. This implies that RLA scales better when the number of beacon nodes in the sensor field is decreased or increased.

3.6.4 Localisation Efficiency

In Fig. 3.16, we plot localisation efficiency of the three compared algorithms for DOI = 0.2. Centroid and RLA algorithms achieve 100% localisation efficiency when the number of beacon nodes is increased beyond 5%. CAB algorithm also

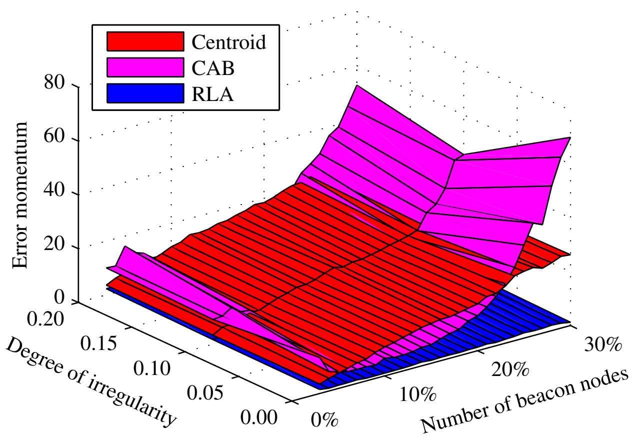


Figure 3.15: Change in error momentum with DOI and number of beacon nodes.

achieves almost 100% efficiency beyond this point with occasional exceptions. In the case of CAB algorithm, a node has to first calculate points of intersection of transmission circles of neighbour beacon nodes and then isolate *valid* points of intersection. Due to irregular radio pattern, isolation of valid points of intersection may not be possible sometimes thereby resulting in lower localisation efficiency.

3.6.5 Degree of Location Intelligence

DOLI for the three algorithms is plotted in Fig. 3.17 for DOI = 0.2. Sensor nodes have better DOLI when using RLA compared to either Centroid or CAB for the entire range of beacon nodes. Beyond 10% beacon nodes, DOLI of RLA remains above 0.95.

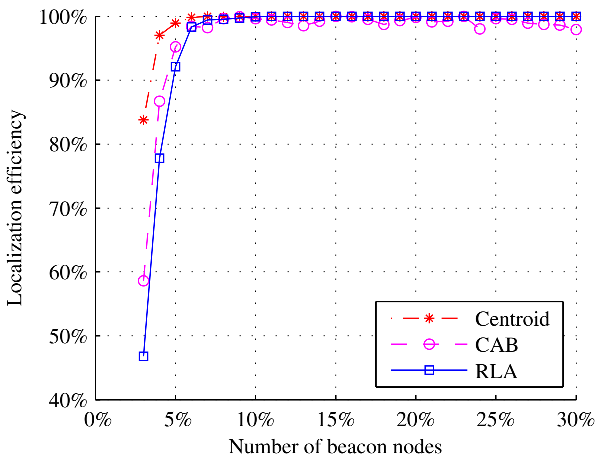


Figure 3.16: Localisation efficiency.

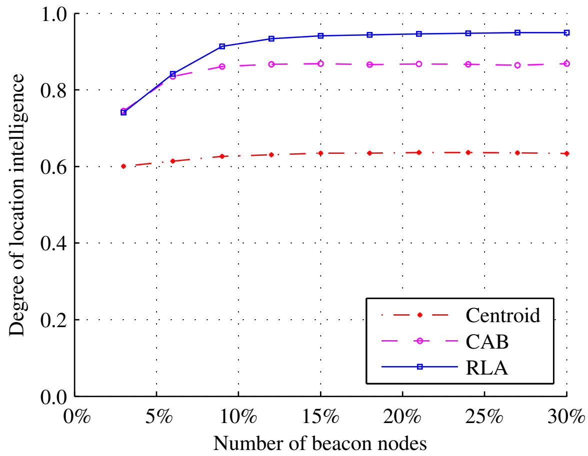


Figure 3.17: Performance Comparison using DOLI.

3.6.6 Node Density

The performance of a localisation algorithm is affected as the size of the sensor field is varied thereby varying density of both sensor and beacon nodes. We perform simulation experiments by deploying 100 sensor nodes in sensor fields of different sizes. The results are described below.

In Fig. 3.18, we plot localisation error as side of the square sensor field is increased from 50 m to 200 m while using 100 sensor nodes, 20% beacon nodes and DOI = 0.2. As the size of the sensor field increases, the beacon node density decreases. Consequently, a sensor node has fewer and distant neighbour beacon nodes, and as a result, higher distance estimation and localisation error while using any of the three compared algorithm.

In Fig. 3.19 and Fig. 3.20, we plot localisation error and localisation efficiency respectively using 100 sensor nodes and DOI = 0.2 as the number of beacon nodes in a large sensor field of size 200 $m \times 200$ m is varied. If the radio range of a sensor node is 10 m and its sensing range is 5 m, then the radio and

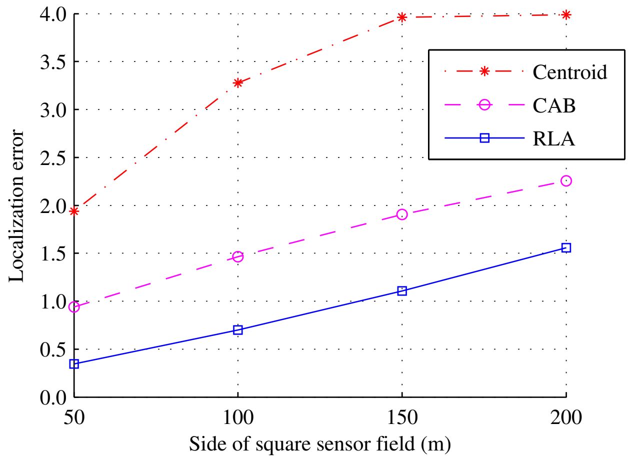


Figure 3.18: Localisation error using varying sizes of sensor field.

sensing areas covered by 100 nodes are $10000\pi\ m^2$ and $2500\pi\ m^2$ respectively. If the coverage areas of individual nodes do not overlap, the maximum radio coverage of the 40000 m^2 sensor field is 79% while its maximum sensing coverage is only 20%. Hence it is a sparse sensor field. When the number of beacon nodes is below 7%, localisation error using RLA is higher than both the Centroid and CAB. When the number of beacon nodes is further increased, there is only marginal improvement in localisation accuracy for Centroid and CAB. However, the improvement using RLA is significant and is much better than either of the two compared algorithms as explained in Section 3.6.1.2. Furthermore, as shown in Fig. 3.20, Centroid and RLA achieve better localisation efficiency compared to CAB as the number of beacon nodes in the sparse sensor field is increased for reasons explained in Section 3.6.8.

3.6.7 Variation in Beacon Signal Step

In a practical situation, it is possible that beacon signal step (d_r) does not remain constant as the transmission power is increased successively. We add

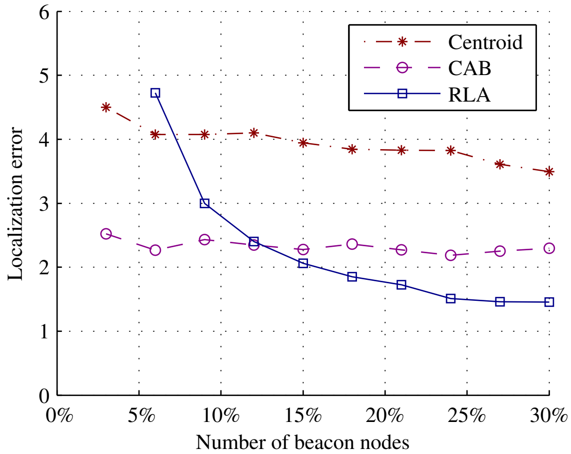


Figure 3.19: Localisation error using 200 $m \times 200$ m sensor field.

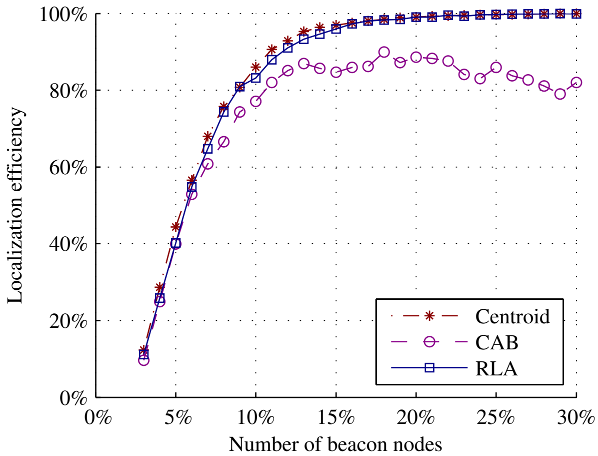


Figure 3.20: Localisation efficiency using 200 $m \times 200$ m sensor field.

an error to the beacon signal step d_r in each wave in a ripple in a 100×100 sensor field with 100 sensor nodes and 20% beacon nodes using $n = 10$. The error is random over the interval $[-\beta d_r \ +\beta d_r]$, where β specifies bound on the random error. For example, if $d_r = 10$ and $\beta = 50\%$, then the error added to d_r in a wave in a ripple is random over the interval $[-0.5 \times d_r + 0.5 \times d_r]$ i.e. bound on the random error is ± 5. We vary error bound β from 0% to 100%, and record localisation error. The results are plotted in Fig. 3.21. Localisation error increases as the error in beacon signal step increases. The addition of the

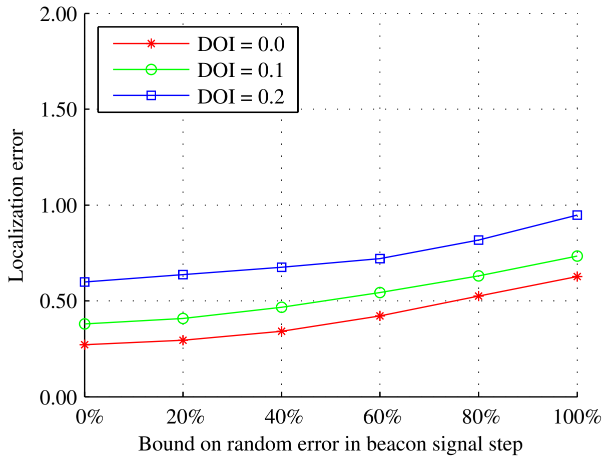


Figure 3.21: Effect of variations in beacon signal step on localisation error.

random error in d_r may cause a sensor node to incorrectly estimate itself lying in a wrong annular ring around a neighbour beacon node, thereby yielding a faulty distance estimate, and hence, higher localisation error. As error in the beacon signal step increases, a sensor node may estimate itself lying in wrong annular rings around more and more neighbour beacon nodes. As a result, the sensor node has incorrect distance estimates to a higher number of neighbour beacon nodes, thereby resulting in an increase in the localisation error. Transmission power of sensor node can be calibrated to minimise error in the beacon signal step.

3.6.8 Performance Comparison with Improved CAB

It may be considered to modify CAB to use a higher number of transmission levels to give improved performance. We describe the possible modified forms of CAB and then present their performance comparison with ripple localisation algorithm.

CAB can be modified in two ways. First, it can either use only three farthest beacon nodes or select all neighbour beacon nodes for position estimation. When using all neighbour beacon nodes, an unknown node does not test for the farthest three nodes. it simply selects all neighbour beacon nodes and then estimates position as in original CAB. Second, the number of transmission power levels can be kept as in CAB or increased to such levels as proposed in RLA. This gives us following four possibilities of CAB. One of these is the original CAB algorithm and the other three are the modified forms of CAB, which we call CAB-A, CAB-B and CAB-C.

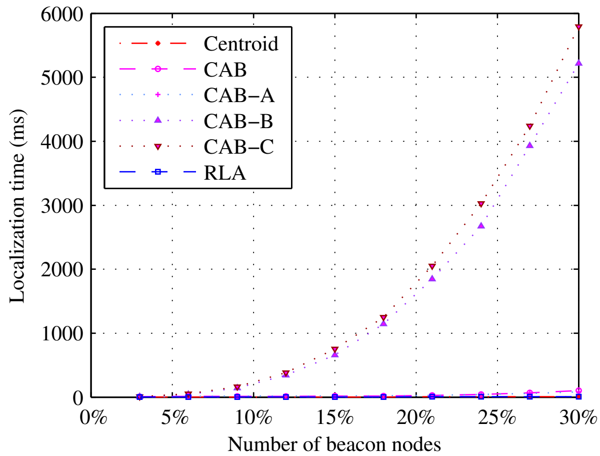
1. CAB: Use the same transmission power level as in CAB and select three farthest beacon nodes for position estimation. This is original CAB algorithm.

2. CAB-A: Use a higher number of transmission levels but only three farthest beacon nodes.

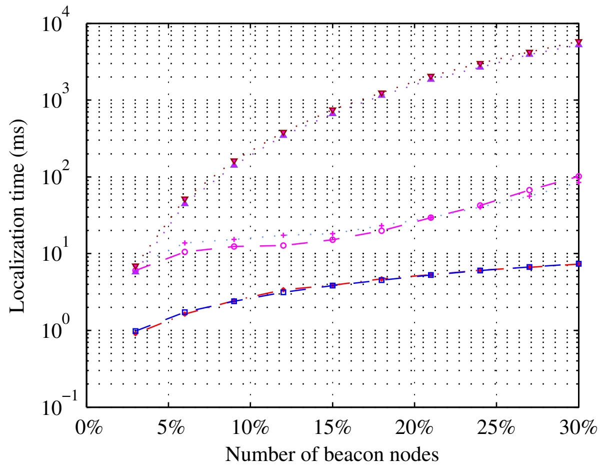
3. CAB-B: While keeping the number of power levels same as in CAB, use all neighbour beacon nodes for position estimation instead of the farthest three.

4. CAB-C: With a higher number of power levels, use all neighbour beacon nodes for localisation.

We investigate all these forms for performance comparison. We plot localisation times in Fig. 3.22 and localisation error in Fig. 3.23 against the number of beacon nodes for RLA and the original and the modified form of CAB using $n = 10$ for DOI $= 0.2$ with 100 sensor nodes deployed in 100×100 sensor field. Accuracy of modified versions of CAB is better as localisation



(a) Linear scale



(b) Semilog scale

Figure 3.22: Localisation time of modified forms of CAB.

error is lower. Accuracy of CAB-A is as good as that of RLA. CAB-B and CAB-C have lower localisation error than RLA. The two factors i.e. inclusion of all neighbour beacon nodes for position estimation and the usage of higher number of transmission levels result in smaller area bounded by points of intersection, and hence, lower localisation error. However, both these factors also result in increased localisation times which are plotted in Fig. 3.22, and lower localisation efficiency which is plotted in Fig. 3.24. From Fig. 3.12 and Fig. 3.22, it is evident that localisation times of CAB-B and CAB-C are very

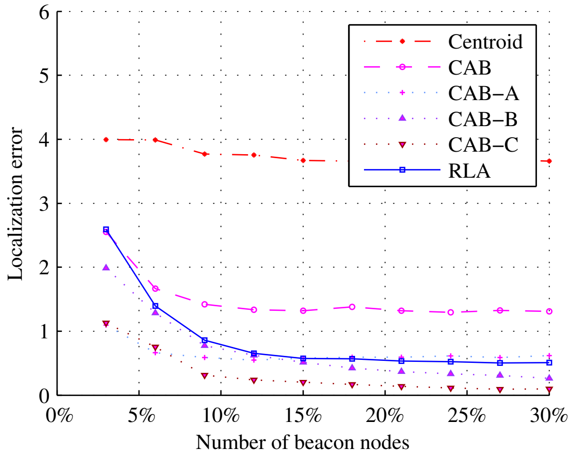


Figure 3.23: Localisation error of modified forms of CAB.

high and of the order of 1000 ms compared to that of RLA which is of the order of 1 ms. When all the neighbour beacon nodes are used for position estimation in the modified version of CAB, points of intersection among the annular rings of all these nodes are calculated. Then each point of intersection is tested to satisfy the criteria of falling within the annular rings of all other neighbour nodes. This results in increased time complexity. Similarly, using higher number of transmission levels results in a higher number of annular rings, more points of intersection and increased time complexity. Localisation times are plotted in Fig. 3.22 using both linear and semilog scales due to large variance in localisation times of compared algorithms.

Complexity of CAB-B and CAB-C increases as the number of beacon nodes increases because a sensor node has to calculate a higher number of points of intersection and then determine valid points among them. To calculate number of points of intersection, let there be k beacon nodes. It takes 2 beacon nodes for circles around them to intersect. Combination of k beacon nodes taken 2 at a time is $\frac{k!}{(k-2)!2!} = \frac{k(k-1)}{2}$. Each combination of 2 beacon nodes contributes

2 annular rings and hence 4 circles and 8 points of intersections. Therefore, maximum number of points of intersection contributed by k beacon nodes is $\frac{8k(k-1)}{2} = 4k(k-1)$ resulting in time complexity of the order of $O(k^2)$. Each of these points of intersection is tested against k annular rings around k beacon nodes. Hence, total number of tests performed by an unknown node to determine valid points is $4k^2(k-1)$ giving a time complexity of $O(k^3)$.

All modified versions of CAB are unable to localise a large number of unknown nodes, as shown in Fig. 3.24. This can be attributed to incorrect estimation of annular ring around a beacon node by an unknown node. Due to irregular radio, a sensor node may incorrectly determine itself lying in a different annular ring than the one it actually lies in around a beacon node. This annular ring is included in the calculation of points of intersection and also in the test to isolate valid points. As required by CAB, a point of intersection must fall within the annular rings of *all* participating neighbour beacon nodes for it to be a valid point of intersection. Due to a single incorrect annular ring, no point of intersection can satisfy this condition resulting in failure to isolate valid

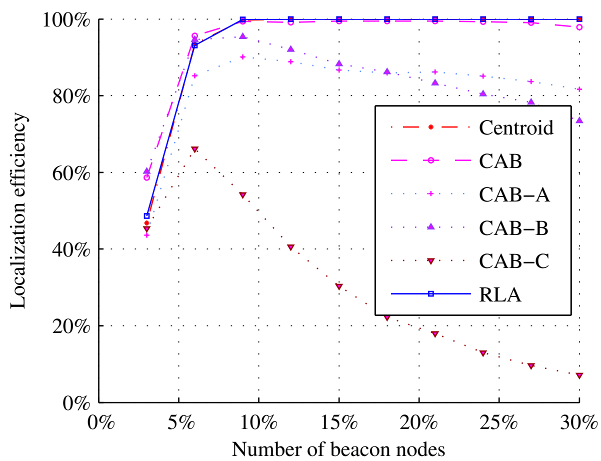


Figure 3.24: Localisation efficiency of modified forms of CAB.

points. Therefore, node is unable to localise and remains unsettled. There are two factors which can contribute to this. Firstly, when all the neighbour beacon nodes are included in position estimation, probability for an unknown node to include an incorrect annular ring is higher. This probability further increases as the number of beacon nodes is increased. Secondly, the higher the number of transmission levels, the smaller the width of annular ring and the higher the chance for an unknown node to estimate itself lying in a wrong annular ring. It is for this reason that CAB-C, which includes both these factors i.e. all neighbour beacon nodes along with a high number of transmission levels, has very low localisation efficiency. However, CAB-A and CAB-B, which use only either one of these factors, have better localisation efficiency. Ripple localisation algorithm, however, has the advantage that it absorbs an incorrect annular ring during position estimation. When using Centroid or RLA, an incorrect annular ring results in a wrong range estimate, and hence it merely contributes to localisation error. However, it does not block a node from localisation.

In this section, performance of proposed ripple localisation algorithm has been evaluated and compared with Centroid and CAB. The results show that overall performance of the ripple localisation algorithm is better than the compared algorithms.

3.7 Summary

We have presented an intelligent, energy efficient and distributed localisation algorithm for industrial wireless sensor networks. The algorithm is able to estimate position without using any additional piece of hardware thereby saving cost, size and energy. The algorithm saves 66.67% energy in transmission of beacon signals compared to those algorithms that transmit beacons at fixed maximum power level. Approximately 92% of this energy efficiency can be achieved using 10 discrete power levels. The algorithm does not require unknown nodes to expend any transmission energy and they can localise passively in an intelligent manner using only processing energy by merely listening to the beacon messages. It achieves 100% localisation efficiency with a high degree of location intelligence within a short localisation time by using only 5% beacon nodes. It also provides control over localisation granularity. Performance of the algorithm is evaluated using simulation and the results show that the algorithm provides good localisation accuracy. We have also proposed and used three new novel metrics for the performance testing and evaluation of localisation algorithms.

Chapter 4

Analytical Model of Localisation Error

Trilateration and multilateration are important localisation techniques used in a diverse range of networks and applications. In this chapter, we use optimisation with calculus and least squares approximation to show that the overdetermined system of equations yielded by multilateration can be reduced to a set of two equations which can be solved simultaneously using conventional methods, such as Cramer's rule. Based upon this result, we also develop and present a novel and unique analytical model for the localisation error resulting from trilateration due to inaccurate range estimates. Using the analytical model, we analyse trilateration errors for localisation applications in short range wireless networks, such as wireless sensor networks and internet of things where the distance estimation errors are comparable to the actual distances. We also determine the minimum and maximum values of localisation error in these networks. In addition, we derive a number of important and useful results which can be used for the development and analysis of localisation algorithms and applications. For example, we show that the localisation error due to the positive distance estimation errors equal to the actual distances is 3 times the localisation error resulting from the same magnitude of negative distance estimation errors. The analytical model and results are verified using simulation.

4.1 Introduction

Location information is important in many applications of science and engineering. For example, in wireless sensor networks, data are ascribed to the location from where it is gathered. For this to be possible, a randomly deployed sensor node should be able to estimate its position [70–72] using a localisation algorithm. These sensor nodes may be deployed in the outdoor [71], indoor [73–76] or underwater [77] environment. Spectrum sensing is an important technique and enabling factor for dynamic spectrum sharing in future 5G communications. Location estimation of the primary user results in reliable spectrum sensing and cognitive enhancement [78]. In context aware and pervasive computing, context or location information of a user is used for the adaption and provision of computing services without explicit intervention of the user [79, 80]. Many large shopping malls track the position information of their visitors to analyse buying patterns, to determine frequency of visits of customers in different areas of the store and other analytics. WiFi signals broadcast by a cell phone carried by a customer are received by the access points and routers and then processed to estimate the real time location of its user [81, 82]. In space and moon exploration missions, an unmanned rover should be able to estimate its position so that it can continue towards its desired destination [83].

Multilateration is an important technique to estimate position of an object given its distances to known positions. It is widely used in location finding applications. For estimation of position in two dimensions, distances to three or more known locations are required. In the case of three dimensional position estimation, distance estimates to four or more locations with known position coordinates are required. When the distance estimates to only three known positions in two dimensions or four known positions in three dimensions are used, then it is termed as trilateration. Global navigation satellite systems (GNSS), such as GPS, GLONASS, Galileo and BeiDou use distance estimates to four or more satellites to estimate position in three dimensions using multilateration [84]. In cellular networks, a cell phone can estimate its position using multilateration if it has distance estimates to three or more base stations [85]. Similarly, multiple base stations can collaborate and use multilateration to estimate the position of a cell phone [86]. In robotics, a moving robot should know its current position for it to be able to decide its next move. In many robotics applications, multilateration is used for this purpose [87]. In navigation applications, an object determines its position using distance estimates from three known locations so that it is able to keep itself on track towards its destination [88, 89]. For example, three or more synchronised transmitters can be used to transmit beacon signals. The object can then estimate its distances from the transmitters using time difference of arrival (TDOA) information of these beacon signals. Position is then determined using multilateration [84, 90]. In surveillance and tracking applications, the position information of an unknown object under observation is determined [91]. If the object is moving, real time position information is estimated to track the object [92]. Three or more transceivers estimate their distances from the unknown object to estimate its position. For example, three or more synchronised receivers can be used to receive the signals transmitted or reflected by an unknown object. Using the TDOA, the receivers can then estimate their distances and position of the unknown object [93]. Multilateration is also used in aircraft surface surveillance and detection equipment [94, 95]. Traffic controllers use this system to track movement of aircrafts and vehicles on the airport surface. This allows them to detect any incursions and potential conflicts on and around runways and taxiways. This is specially helpful in low visibility operations. In a technique, known as wide area multilateration (WAM), a sensor network is deployed around the airport, mountain ranges or other areas where location of an aircraft is to be identified [96]. The sensors query the aircraft transponders and estimate their distances from the aircraft by analysing the response. Multilateration is then used to estimate position of the aircraft. WAM is useful in particular in areas where there is no radar coverage. Furthermore, WAM is much cheaper than systems comprising of radars which are much more expensive. Another use of multilateration is to localise artillery fire using sound ranging [97]. Multilateration is also used to estimate the epicentre of an earthquake [98]. Seismostations located around the world record any seismic activity on seismograms. Energy released during an earthquake travels in the form of different types of waves. The seismograms record the time difference of arrival of longitudinal primary waves (P-waves) and transverse secondary waves (S-waves). P-waves travel 1.7 times faster than the S-waves. Knowing their speeds and the time difference of arrival of both types of waves, distance to the epicentre of an earthquake can be estimated. With the help of seismograms from three or more seismic stations, the distances of these seismic stations from the epicentre can be estimated. Next multilateration is used to estimate position of the epicentre of an earthquake. Multilateration has also been proposed for the early prediction of time and location of earthquakes, precision surveying, plate tectonics and orbital applications [99, 100]. The system uses pulsed laser and airborne retroreflectors to determine range among ground stations with a precision of 1 cm at regular intervals. An analysis of change in the range along with multilateration is used for the prediction of time and location of earthquakes.

There are many historical navigation systems that use multilateration. These include Gee, deployed by the British royal air force, LORAN (long range navigation) by US navy, Decca navigation system by the British royal navy, Omega built and operated by US in partnership with six other nations, CHAYKA and Alpha by the former Soviet Union [101]. All these navigation systems deploy chains of ground stations comprising of master and secondary stations which transmit signals at regular intervals. Any object in air, sea or land carrying a receiving station can estimate its position using hyperbolic navigation. In these historical radio navigation systems, the multilateration problem is solved and the position is fixed using hyperbolic curves. Hence, in radio navigation, the terms multilateration and hyperbolic navigation are used synonymously. To further explain, let there be a master station at (x_1, y_1) with two secondary stations at (x_2, y_2) and (x_3, y_3). Let the distances of these stations from a receiving station at (x, y) be represented by r_1, r_2, and r_3 respectively. We can now write equations of circles around these ground stations as under:

$$
\begin{aligned}
(x - x_1)^2 + (y - y_1)^2 &= r_1^2 \\
(x - x_2)^2 + (y - y_2)^2 &= r_2^2 \\
(x - x_3)^2 + (y - y_3)^2 &= r_3^2
\end{aligned} \tag{4.1}
$$

The distances of these stations from the receiver are given by

$$
\begin{aligned}
r_1 &= \sqrt{(x - x_1)^2 + (y - y_1)^2} \\
r_2 &= \sqrt{(x - x_2)^2 + (y - y_2)^2} \\
r_3 &= \sqrt{(x - x_3)^2 + (y - y_3)^2}
\end{aligned} \tag{4.2}
$$

The receiving station listens to the signals transmitted by the master and secondary stations and records the time difference in the receiving of signals from each pair of master and secondary station. From the time difference, the difference in distance of these stations from the receiver can be calculated using $d = v \times t$. From (4.2), the difference in the distance of the receiver from the first pair of master and secondary stations is given by

$$
\begin{aligned}
r_{12} &= r_1 - r_2 \\
r_{12} &= \sqrt{(x - x_1)^2 + (y - y_1)^2} - \sqrt{(x - x_2)^2 + (y - y_2)^2}
\end{aligned} \tag{4.3}
$$

Similarly, the difference in distance of the receiver from the second pair of master and secondary stations is given by

$$
\begin{aligned}
r_{13} &= r_1 - r_3 \\
r_{13} &= \sqrt{(x - x_1)^2 + (y - y_1)^2} - \sqrt{(x - x_3)^2 + (y - y_3)^2}
\end{aligned} \tag{4.4}
$$

In (4.3) and (4.4), the distances r_{12} and r_{13} are known and positions of ground stations (x_1, y_1), (x_2, y_2) and (x_3, y_3) are also known. Hence, (4.3) and (4.4) can be solved for the position (x, y) of the receiver. At the time of development of these navigation systems, the digital processing and computing resources were not easily available. Therefore, curves from (4.3) and (4.4) for different possible values of time differences or distance differences for chains of master and secondary stations were plotted in the form of charts. Using the TDOA of signals received from different pairs of master and secondary stations, the position of the receiver was fixed at the intersection of hyperbolic curves corresponding to the time delays using the charts. The shape of the curves is in the form of hyperbola and hence this is termed as hyperbolic navigation. After the development of global navigation satellite systems (GNSS), these historical navigation systems are being phased out.

As is evident from the previous discussion, trilateration and multilateration are used in a wide variety of systems ranging from GNSS where distance estimates are in thousands of kilometres to wireless sensor networks where the distances are in the range of a few meters. Different applications employ different techniques for estimation of distance between the unknown object and the known position [102]. For example, in many navigation systems, synchronised clocks and time difference of arrival (TDOA) are used to estimate distances. Examples of such systems include GNSS, such as GPS, GLONASS, Galileo and BeiDou. In such systems, the distances between the unknown object, such as GPS receiver, and the known positions i.e. the GPS satellites are in thousands of kilometres. However, due to tightly synchronised atomic clocks, the range estimation errors are very small and are only a minor fraction of the actual distances. In short range wireless networks, such as wireless sensor networks (WSN), internet of things (IoT) and wireless personal area networks (WPAN), the distances are in the range of a few meters. Due to energy, size and cost limitations, additional synchronisation hardware may not be used for range estimation in these networks. Even if used, only inexpensive clocks may be used which do not give accurate range estimates. Therefore, alternate means such as received signal strength (RSS) are used to estimate range in short range wireless networks [103]. While the range estimates from the RSS can be used for localisation, these are not precise. The range estimation errors resulting from distance estimation using RSS are comparable to the actual distances between the sensor nodes [103, 104]. Consequently, range estimation errors in such systems have pronounced effect on the estimated position and may result in large localisation errors. Therefore, exploration of localisation error as a function of range estimation errors in wireless sensor networks constitutes an interesting research investigation.

This and next chapter of the thesis make several contributions. First, we show that the overdetermined system of equations resulting from multilateration can be reduced to a set of two equations which can be solved simultaneously using conventional techniques, such as Cramer's rule. Second, using the result, we give an accurate analytical model of localisation error as a function of distance estimation errors, when position is estimated using trilateration. Given the actual distances from the beacon nodes and the distance estimation errors, our model is able to accurately calculate the localisation error, without needing to first estimate position of the unknown node. The model establishes the relationship between localisation error, the distance estimation errors and the geometry formed by the positions of beacon nodes. Third, using the analytical model, we analyse localisation error and determine its extreme values for systems where the range estimation errors are comparable to the actual distances. Fourth, we derive more than thirty important and useful results which can be used in the design, development and analysis of localisation solutions. Summary of these results is given in Section 5.1.6. Fifth, we verify the results using simulation. To the best of our knowledge, this is the first attempt to develop an accurate analytical model for the localisation error. This is also the first attempt to determine the minimum and maximum values of localisation error and its various components.

The solution to multilateration using Cramer's rule developed in Section 4.3 and the analytical model of the localisation error developed in Section 4.4 are applicable to all applications wherever trilateration and multilateration are used irrespective of the size of the distance estimation errors. Error analysis carried out in Section 5.1 and numerical results in Section 5.2 are in the context of wireless sensor networks. However, the error analysis and the results are applicable to trilateration systems in any network where the distance estimation errors are comparable to the actual distances. Examples of such networks are short range wireless networks, such as wireless sensor networks (WSN), internet of things (IoT) and wireless personal area networks (WPAN).

4.2 Related Work

Major work to investigate errors due to trilateration and multilateration has either been done in the context of historical hyperbolic navigation systems or GNSS. The problem of localisation error in wireless sensor networks has also been investigated. However, it is not in the context of trilateration and multilateration. In the following, we give a summary of the related work in both these areas.

Different types of errors in a hyperbolic multilateration system are described in [105]. The errors are classified as systematic, correlated and random. Systematic errors are attributed to delays caused by deviation of wave propagation from the ideal propagation and imperfect antenna position measurement. Correlated errors are errors due to the multipath signal propagation and the actual troposphere conditions along the path of signal propagation. The random error results from the additive noise at receiver and quantization of TDOA measurements. The effect of noisy distances on the estimated position using two different localisation techniques is presented in [106]. The author in [107] shows that if the error in estimated distances from the direction finding (DF) ground stations in a hyperbolic navigation system follows a normal error distribution, then the localisation error in the estimated position follows a normal elliptical error distribution. The author also proposes that the reciprocal of rms localisation error should be used as a measure for the reliability of estimated position. In [108], the lengths and direction cosines of the multilateration error ellipsoids semiaxes are calculated in terms of rms error and direction cosines of estimated distances. The lengths and directions of ellipsoid axes give a measure of the GDOP of the estimated position. Characteristics of GPS positioning errors are investigated in [109] and an analytical formula for the covariance of the positioning error is derived.

It is observed in [84] that relative geometry of positions of the unknown object and the satellites plays an important role in the accuracy of estimated position. The unknown object and the GNSS satellites may be positioned such that the area of the intersected region from multilateration is relatively small or their relative positions may be such that the area of the intersected region is quite large. Given the same error in the estimated distances, the localisation error will be higher in the latter case. In navigation, the effect of geometry due to relative positions of beacon nodes and the unknown object is quantified using dilution of precision (DOP). DOP is further categorised as horizontal dilution of precision (HDOP), vertical dilution of precision (VDOP), position dilution of precision (PDOP) and time dilution of precision (TDOP). The lesser the DOP, the better it is. DOP values less than 5 are considered good. In general, when the beacon nodes are close together, the resulting geometry is not optimum for position estimation and the DOP is high. On the other hand, when the beacon nodes are distributed all around the unknown node and are at a distance from each other, the geometry is good and the DOP is low.

In the case of wireless sensor networks, a Cramer-Rao lower bound of the expected localisation error for a single sensor node is derived in [110] assuming that the sensor nodes measure RSS or TOA. The fundamental limits of localisation in indoor environment using signal strength of IEEE 802.11 wireless local area networks is explained in [111]. The authors show that a median localisation error of 10 feet can be expected over a range of localisation algorithms. Authors also suggest that simple and computationally inexpensive localisation algorithms are preferable as complexity and additional computation do not result in significantly improved performance. In [112], a number of simulation experiments are performed and based upon the results, an empirical formula for localisation error for a centralised wireless sensor network is presented. The formula is a function of average distance between beacon nodes, the number of distance estimations performed by the beacon nodes and certain network parameters. A lower bound on the localisation error for a network of sensor nodes deployed randomly according to Poisson point process is derived in [113]. The authors show that the lower bound is a function of the distance measurements between sensor and beacon nodes and density of beacon nodes. Using a log-normal shadow-fading radio model for wireless sensor networks, the authors of [114] derive an expression for the distribution of lower bound on localisation error for any range free localisation algorithm. The authors of [115, 116] investigate the possibility of upper bounds of localisation error for range based algorithms. Assuming that positive distance estimation errors are always present, a highly probable upper bound on localisation error is proposed. The upper bounds are formulated as nonconvex optimisation problem and then a relaxation technique is used to obtain convex problems which are easier to solve.

4.3 Multilateration Solution

In this section, we use optimisation with calculus and least squares approximation to arrive at a simplified solution of multilateration. We also provide an algebraic method to arrive at the same result.

To explain position estimation using multilateration, consider an unknown node with actual position coordinates (x_a, y_a) having k neighbour beacon nodes with position coordinates (x_1, y_1), (x_2, y_2), ..., (x_k, y_k). If estimated position of the unknown node is (x, y), then a set of following equations of circles around beacon nodes can be obtained by using the position coordinates of beacon nodes as centres and range estimates $r_1, r_2, ..., r_k$ as radii:

$$\begin{bmatrix} (x-x_1)^2+(y-y_1)^2 \\ (x-x_2)^2+(y-y_2)^2 \\ \vdots \\ (x-x_k)^2+(y-y_k)^2 \end{bmatrix} = \begin{bmatrix} r_1^2 \\ r_2^2 \\ \vdots \\ r_k^2 \end{bmatrix} \tag{4.5}$$

Expanding square terms on the left side and rearranging:

$$\begin{bmatrix} x^2+y^2-2x_1x-2y_1y \\ x^2+y^2-2x_2x-2y_2y \\ \vdots \\ x^2+y^2-2x_kx-2y_ky \end{bmatrix} = \begin{bmatrix} r_1^2-x_1^2-y_1^2 \\ r_2^2-x_2^2-y_2^2 \\ \vdots \\ r_k^2-x_k^2-y_k^2 \end{bmatrix} \tag{4.6}$$

Subtracting last row from each of the rows above it:

$$\begin{bmatrix} 2(x_k-x_1)x+2(y_k-y_1)y \\ 2(x_k-x_2)x+2(y_k-y_2)y \\ \vdots \\ 2(x_k-x_{k-1})x+2(y_k-y_{k-1})y \end{bmatrix} = \begin{bmatrix} r_1^2-r_k^2+x_k^2-x_1^2+y_k^2-y_1^2 \\ r_2^2-r_k^2+x_k^2-x_2^2+y_k^2-y_2^2 \\ \vdots \\ r_{k-1}^2-r_k^2+x_k^2-x_{k-1}^2+y_k^2-y_{k-1}^2 \end{bmatrix} \tag{4.7}$$

Separating the unknowns (x, y), this can be rewritten in matrix form as below:

$$\begin{bmatrix} (x_k - x_1) & (y_k - y_1) \\ (x_k - x_2) & (y_k - y_2) \\ \vdots & \vdots \\ (x_k - x_{k-1}) & (y_k - y_{k-1}) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2} \begin{bmatrix} r_1^2 - r_k^2 + x_k^2 - x_1^2 + y_k^2 - y_1^2 \\ r_2^2 - r_k^2 + x_k^2 - x_2^2 + y_k^2 - y_2^2 \\ \vdots \\ r_{k-1}^2 - r_k^2 + x_k^2 - x_{k-1}^2 + y_k^2 - y_{k-1}^2 \end{bmatrix} \tag{4.8}$$

Using matrix notation, this can be written as:

$$\mathbf{A}\mathbf{z} = \mathbf{R} \tag{4.9}$$

where $\mathbf{z} = \begin{bmatrix} x & y \end{bmatrix}^T$ and

$$\mathbf{A} = \begin{bmatrix} (x_k - x_1) & (y_k - y_1) \\ (x_k - x_2) & (y_k - y_2) \\ \vdots & \vdots \\ (x_k - x_{k-1}) & (y_k - y_{k-1}) \end{bmatrix} \tag{4.10}$$

$$\mathbf{R} = \frac{1}{2} \begin{bmatrix} r_1^2 - r_k^2 + x_k^2 - x_1^2 + y_k^2 - y_1^2 \\ r_2^2 - r_k^2 + x_k^2 - x_2^2 + y_k^2 - y_2^2 \\ \vdots \\ r_{k-1}^2 - r_k^2 + x_k^2 - x_{k-1}^2 + y_k^2 - y_{k-1}^2 \end{bmatrix} \tag{4.11}$$

To further investigate a solution, let us consider (4.5) which can be formulated as least squares approximation problem as under:

$$\hat{z} = \underset{z}{\operatorname{argmin}} \sum_{i=1}^{k} e_i(z)^2 \tag{4.12}$$

where k is number of neighbour beacon nodes, $\mathbf{z} = \begin{bmatrix} x & y \end{bmatrix}^T$ and is a vector of estimated and best-fit values of position coordinates, and $e_i(z)$ is a residual function given by

$$e_i(z) = r_i^2 - \{(x - x_i)^2 + (y - y_i)^2\} \text{ where } i = 1, 2, ..., k \tag{4.13}$$

The problem in (4.12) relates to nonlinear least squares and is categorised as unconstrained nonlinear optimisation problem. We reduce the problem from nonlinear least squares in (4.5) to linear least squares in (4.8) by subtracting the last row from the rows above. Based upon (4.8), the residual function becomes

$$e_i(z) = b_i - \{a_{i1}x + a_{i2}y\} \text{ where } i = 1, 2, ..., k - 1 \tag{4.14}$$

where

$$a_{i1} = x_k - x_i \tag{4.15}$$

$$a_{i2} = y_k - y_i \tag{4.16}$$

$$b_i = \frac{1}{2}[r_i^2 - r_k^2 + x_k^2 - x_i^2 + y_k^2 - y_i^2] \text{ where } i = 1, 2, ..., k - 1 \tag{4.17}$$

Sum of squares of the residuals $S(x, y)$ is given by

$$S = \sum_{i=1}^{k-1} e_i^2 = \sum_{i=1}^{k-1} [b_i - (a_{i1}x + a_{i2}y)]^2 \tag{4.18}$$

Our objective is to minimise this function. To determine the values of x and

y for which S is minimum, we evaluate

$$\frac{\partial S}{\partial x} = 0 \tag{4.19}$$

$$\frac{\partial S}{\partial y} = 0 \tag{4.20}$$

Differentiating (4.18) with respect to x and setting the result equal to 0, we get

$$\frac{\partial S}{\partial x} = 2\sum_{i=1}^{k-1}[b_i - (a_{i1}x + a_{i2}y)](-a_{i1}) = 0 \tag{4.21}$$

$$\sum_{i=1}^{k-1}(a_{i1}x + a_{i2}y)a_{i1} = \sum_{i=1}^{k-1} a_{i1}b_i \tag{4.22}$$

Differentiating (4.18) with respect to y and setting the result equal to 0, we get

$$\frac{\partial S}{\partial y} = 2\sum_{i=1}^{k-1}[b_i - (a_{i1}x + a_{i2}y)](-a_{i2}) = 0 \tag{4.23}$$

$$\sum_{i=1}^{k-1}(a_{i1}x + a_{i2}y)(a_{i2}) = \sum_{i=1}^{k-1} a_{i2}b_i \tag{4.24}$$

Equations (4.22) and (4.24) can be combined in matrix form to arrive at the normal equation as under

$$\mathbf{A^T A z} = \mathbf{A^T R} \tag{4.25}$$

We can arrive at the same result using algebraic method as well. Having subtracted last row from the rows above it, the order of matrix $\mathbf{A}$ in (4.8)-(4.10) is $(\mathbf{k-1}) \times \mathbf{2}$ if an unknown node has $\mathbf{k}$ neighbour beacon nodes. As matrix $\mathbf{A}$ is not necessarily square, we cannot use

$$\mathbf{z} = \mathbf{A^{-1}R} \tag{4.26}$$

to determine $\mathbf{z}$. Therefore, multiplying both sides of (4.9) by $\mathbf{A^T}$, we get

$$\mathbf{A^T A z} = \mathbf{A^T R} \tag{4.27}$$

which is the same result as found earlier in (4.25) using calculus.

It is to be noted that the matrix $\mathbf{A}$ has an order of $(k-1) \times 2$ if the unknown node has k neighbour beacon nodes. It has $k-1$ rows as the last row is subtracted from the rows above it. It has 2 columns due to position coordinates x and y. Consequently, the order of matrix $\mathbf{A^T}$ is $2 \times (k-1)$ i.e. it always has 2 rows. Hence, we can combine (4.22) and (4.24) in matrix form in (4.25). Moreover, order of the matrix $\mathbf{A^T A}$ is 2×2. Hence matrices $\mathbf{A^T A}$ and $\mathbf{z}$ are conformable for multiplication. Similarly, the matrix $\mathbf{A^T R}$ has an order of 2×1. Therefore, (4.25) can also be rewritten as below

$$\mathbf{H z} = \mathbf{T} \tag{4.28}$$

where $\mathbf{H} = \mathbf{A^T A}$ is a 2×2 matrix, $\mathbf{z}$ is a 2×1 column matrix and $\mathbf{T} = \mathbf{A^T R}$ is also a 2×1 column matrix. Hence (4.28) can be solved using Cramer's rule. It is to be noted that (4.25) or (4.27) can also be rewritten as below to give another closed-form and unique solution to (4.9):

$$\mathbf{z} = (\mathbf{A^T A})^{\mathbf{-1}} \mathbf{A^T R} \tag{4.29}$$

$$\mathbf{z} = \mathbf{A^+ R} \tag{4.30}$$

where

$$\mathbf{A^+} = (\mathbf{A^T A})^{\mathbf{-1}} \mathbf{A^T} \tag{4.31}$$

is Moore-Penrose pseudoinverse of $\mathbf{A}$. Solution to (4.5) using Cramer's rule on its transformed form in (4.28) or using pseudoinverse in (4.30) or any other least squares approximation such as Newton's method yields the same values of $\mathbf{z} = \begin{bmatrix} x & y \end{bmatrix}^T$. We choose to localise an unknown node only when it has three or more neighbour beacon nodes. Therefore, $k - 1 \geq 2$ i.e. number of rows in matrix $\mathbf{A}$ is always equal to or greater than the number of columns. The number of columns in the matrix is always 2 and is less than or equal to the number of rows. As the matrix elements are derived from the independent and random positions of beacon nodes, the rows and columns are linearly independent and the matrix $\mathbf{A}$ is full rank. As a result, the matrix $\mathbf{A}$ is nonsingular and invertible and the achieved solution in (4.28) or (4.30) is unique. This gives us an exact solution if it exists and an approximate solution otherwise. The only exception to this is the case when all the neighbour beacon nodes of an unknown node are collinear. In this case, the matrix $\mathbf{A}$ is singular and the solution does not exist. However, the possibility of all neighbour beacon nodes being collinear is remote if these are deployed randomly.

4.4 Analytical Model

Let an unknown node have k neighbour beacon nodes. Let the estimated distances between the unknown node and the neighbour beacon nodes be $r_1, r_2, ..., r_k$ and the actual distances be $r_{a1}, r_{a2}, ..., r_{ak}$. If the errors in the estimated distances are $e_1, e_2, ..., e_k$, then

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{bmatrix} = \begin{bmatrix} r_{a1} + e_1 \\ r_{a2} + e_2 \\ \vdots \\ r_{ak} + e_k \end{bmatrix} \tag{4.32}$$

Hence, (4.8) can be rewritten as

$$\begin{bmatrix} (x_k - x_1) & (y_k - y_1) \\ (x_k - x_2) & (y_k - y_2) \\ \vdots & \vdots \\ (x_k - x_{k-1}) & (y_k - y_{k-1}) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
$$= \frac{1}{2} \begin{bmatrix} (r_{a1} + e_1)^2 - (r_{ak} + e_k)^2 & +x_k^2 - x_1^2 \\ & +y_k^2 - y_1^2 \\ (r_{a2} + e_2)^2 - (r_{ak} + e_k)^2 & +x_k^2 - x_2^2 \\ & +y_k^2 - y_2^2 \\ \vdots & \\ (r_{ak-1} + e_{ak-1})^2 - (r_{ak} + e_k)^2 & +x_k^2 - x_{k-1}^2 \\ & +y_k^2 - y_{k-1}^2 \end{bmatrix} \tag{4.33}$$

Let us now consider a simple case where an unknown node has three neighbour beacon nodes so that $k = 3$ and for this specific case (4.33) becomes:

$$\begin{bmatrix}(x_3 - x_1) & (y_3 - y_1) \\ (x_3 - x_2) & (y_3 - y_2)\end{bmatrix}\begin{bmatrix}x \\ y\end{bmatrix} = \frac{1}{2}\begin{bmatrix}(r_{a1} + e_1)^2 - (r_{a3} + e_3)^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2 \\ (r_{a2} + e_2)^2 - (r_{a3} + e_3)^2 + x_3^2 - x_2^2 + y_3^2 - y_2^2\end{bmatrix} \tag{4.34}$$

As explained in the section 4.3, this can be solved using Cramer's rule. Let

$$\begin{aligned}\Delta &= \begin{vmatrix}(x_3 - x_1) & (y_3 - y_1) \\ (x_3 - x_2) & (y_3 - y_2)\end{vmatrix} \\ &= (x_3 - x_1)(y_3 - y_2) - (x_3 - x_2)(y_3 - y_1) \\ &= x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \\ &= y_1(x_3 - x_2) + y_2(x_1 - x_3) + y_3(x_2 - x_1)\end{aligned} \tag{4.35}$$

$$
\begin{aligned}
\Delta_1 = \frac{1}{2}&\begin{vmatrix} \begin{matrix}(r_{a1}+e_1)^2-(r_{a3}+e_3)^2\\ +x_3^2-x_1^2+y_3^2-y_1^2\end{matrix} & (y_3-y_1) \\ \\ \begin{matrix}(r_{a2}+e_2)^2-(r_{a3}+e_3)^2\\ +x_3^2-x_2^2+y_3^2-y_2^2\end{matrix} & (y_3-y_2) \end{vmatrix}\\
&= \frac{1}{2}(y_3-y_2)[r_{a1}^2+\underbrace{2r_{a1}e_1+e_1^2}-r_{a3}^2\underbrace{-2r_{a3}e_3-e_3^2}\\
&+x_3^2-x_1^2+y_3^2-y_1^2]\\
&-\frac{1}{2}(y_3-y_1)[r_{a2}^2+\underbrace{2r_{a2}e_2+e_2^2}-r_{a3}^2\underbrace{-2r_{a3}e_3-e_3^2}\\
&+x_3^2-x_2^2+y_3^2-y_2^2]\\
&= \frac{1}{2}(y_3-y_2)[r_{a1}^2-r_{a3}^2+x_3^2-x_1^2+y_3^2-y_1^2]\\
&-\frac{1}{2}(y_3-y_1)[r_{a2}^2-r_{a3}^2+x_3^2-x_2^2+y_3^2-y_2^2]\\
&+\frac{1}{2}(y_3-y_2)[2r_{a1}e_1+e_1^2-2r_{a3}e_3-e_3^2]\\
&-\frac{1}{2}(y_3-y_1)[2r_{a2}e_2+e_2^2-2r_{a3}e_3-e_3^2]
\end{aligned}
\tag{4.36}
$$

Similarly

$$
\begin{aligned}
\Delta_2 = \frac{1}{2}&\begin{vmatrix} x_3 - x_1 & \begin{array}{c}(r_{a1}+e_1)^2-(r_{a3}+e_3)^2\\ +x_3^2-x_1^2+y_3^2-y_1^2\end{array} \\ \\ x_3 - x_2 & \begin{array}{c}(r_{a2}+e_2)^2-(r_{a3}+e_3)^2\\ +x_3^2-x_2^2+y_3^2-y_2^2\end{array}\end{vmatrix} \\
&= \frac{1}{2}(x_3-x_1)[r_{a2}^2+\underbrace{2r_{a2}e_2+e_2^2}-r_{a3}^2\underbrace{-2r_{a3}e_3-e_3^2} \\
&+x_3^2-x_2^2+y_3^2-y_2^2] \\
&-\frac{1}{2}(x_3-x_2)[r_{a1}^2+\underbrace{2r_{a1}e_1+e_1^2}-r_{a3}^2\underbrace{-2r_{a3}e_3-e_3^2} \\
&+x_3^2-x_1^2+y_3^2-y_1^2] \\
&= \frac{1}{2}(x_3-x_1)[r_{a2}^2-r_{a3}^2+x_3^2-x_2^2+y_3^2-y_2^2] \\
&-\frac{1}{2}(x_3-x_2)[r_{a1}^2-r_{a3}^2+x_3^2-x_1^2+y_3^2-y_1^2] \\
&+\frac{1}{2}(x_3-x_1)[2r_{a2}e_2+e_2^2-2r_{a3}e_3-e_3^2] \\
&-\frac{1}{2}(x_3-x_2)[2r_{a1}e_1+e_1^2-2r_{a3}e_3-e_3^2]
\end{aligned}
\tag{4.37}
$$

From (4.35) and (4.36), the estimated value of x coordinate is given by

$$
x = \frac{\Delta_1}{\Delta} \tag{4.38}
$$

$$
\begin{aligned}
x &= \frac{1}{2\Delta}(y_3-y_2)[r_{a1}^2-r_{a3}^2+x_3^2-x_1^2+y_3^2-y_1^2] \\
&-\frac{1}{2\Delta}(y_3-y_1)[r_{a2}^2-r_{a3}^2+x_3^2-x_2^2+y_3^2-y_2^2] \\
&+\frac{1}{2\Delta}(y_3-y_2)[2r_{a1}e_1+e_1^2-2r_{a3}e_3-e_3^2] \\
&-\frac{1}{2\Delta}(y_3-y_1)[2r_{a2}e_2+e_2^2-2r_{a3}e_3-e_3^2] \\
x &= x_a + E_x
\end{aligned}
\tag{4.39}
$$

so that

$$E_x = x - x_a \tag{4.40}$$

where x_a is the x-coordinate of the actual position of the unknown node and is given by

$$\begin{aligned} x_a = & \frac{1}{2\Delta}(y_3 - y_2)[r_{a1}^2 - r_{a3}^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2] \\ & - \frac{1}{2\Delta}(y_3 - y_1)[r_{a2}^2 - r_{a3}^2 + x_3^2 - x_2^2 + y_3^2 - y_2^2] \end{aligned} \tag{4.41}$$

$$x_a = k_1 C_1 + k_2 C_2 \tag{4.42}$$

and E_x is the x component of the localisation error and is given by

$$\begin{aligned} E_x = & \frac{1}{2\Delta}(y_3 - y_2)[2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2] \\ & - \frac{1}{2\Delta}(y_3 - y_1)[2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2] \end{aligned} \tag{4.43}$$

$$E_x = k_1 E_1 + k_2 E_2 \tag{4.44}$$

where

$$k_1 = \frac{1}{2\Delta}(y_3 - y_2) \tag{4.45}$$

$$k_2 = -\frac{1}{2\Delta}(y_3 - y_1) \tag{4.46}$$

$$C_1 = r_{a1}^2 - r_{a3}^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2 \tag{4.47}$$

$$C_2 = r_{a2}^2 - r_{a3}^2 + x_3^2 - x_2^2 + y_3^2 - y_2^2 \tag{4.48}$$

$$E_1 = 2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2 \tag{4.49}$$

$$E_2 = 2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2 \tag{4.50}$$

Similarly, estimated value of the y coordinate is given by

$$y = \frac{\Delta_2}{\Delta} \tag{4.51}$$

$$\begin{aligned}
y = & \frac{1}{2\Delta}(x_3 - x_1)[r_{a2}^2 - r_{a3}^2 + x_3^2 - x_2^2 + y_3^2 - y_2^2] \\
& - \frac{1}{2\Delta}(x_3 - x_2)[r_{a1}^2 - r_{a3}^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2] \\
& + \frac{1}{2\Delta}(x_3 - x_1)[2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2] \\
& - \frac{1}{2\Delta}(x_3 - x_2)[2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2] \\
y = & \, y_a + E_y
\end{aligned} \tag{4.52}$$

so that

$$E_y = y - y_a \tag{4.53}$$

where y_a is the y-coordinate of the actual position of the unknown node and is given by

$$\begin{aligned}
y_a = & \frac{1}{2\Delta}(x_3 - x_1)[r_{a2}^2 - r_{a3}^2 + x_3^2 - x_2^2 + y_3^2 - y_2^2] \\
& - \frac{1}{2\Delta}(x_3 - x_2)[r_{a1}^2 - r_{a3}^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2]
\end{aligned} \tag{4.54}$$

$$y_a = k_3 C_1 + k_4 C_2 \tag{4.55}$$

and E_y is the y component of the localisation error and is given by

$$\begin{aligned}
E_y = & -\frac{1}{2\Delta}(x_3 - x_2)[2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2] \\
& + \frac{1}{2\Delta}(x_3 - x_1)[2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2]
\end{aligned} \tag{4.56}$$

$$E_y = k_3 E_1 + k_4 E_2 \tag{4.57}$$

where

$$k_3 = -\frac{1}{2\Delta}(x_3 - x_2) \tag{4.58}$$

$$k_4 = \frac{1}{2\Delta}(x_3 - x_1) \tag{4.59}$$

In summary, E_x and E_y can be written as

$$\begin{aligned} E_x &= k_1[2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2] \\ &+ k_2[2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2] \end{aligned} \tag{4.60}$$

$$\begin{aligned} E_y &= k_3[2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2] \\ &+ k_4[2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2] \end{aligned} \tag{4.61}$$

where k_1, k_2, k_3 and k_4 are constants whose values depend upon the known fixed position coordinates of beacon nodes. E_1 and E_2 are functions of distance estimation errors e_1, e_2 and e_3. From (4.44) and (4.57), we can infer that x and y components of localisation error differ only due to the constants k_1, k_2, k_3 and k_4. As these constants depend upon the relative positions of beacon nodes, the difference in E_x and E_y components of localisation error depends upon the geometry of placement of the beacon nodes. The distance estimation error, reflected in E_1 and E_2, does not contribute to this difference. From (4.44) and (4.57), we also observe that $E_x = E_y$ if $k_1 = k_3$ and $k_2 = k_4$. However, under these conditions $y_3 - y_2 = x_2 - x_3$ and $y_1 - y_3 = x_3 - x_1$ so that the beacon nodes are collinear and position estimation using either (4.25), (4.28) or (4.30) may not be possible as the resultant matrix $\mathbf{A}$ is singular.

It is to be noted that in the estimated position coordinates x and y, the errors e_1, e_2 and e_3 appear only in the numerators Δ_1 and Δ_2. There are no error terms in the denominator Δ. Further, if the error terms E_x and

E_y are eliminated, then the localisation error will be reduced to zero and the position estimate will be accurate. Alternatively, the smaller are the error terms, the higher is the localisation accuracy.

It is further to be noted that the distance estimation error e_i can be either positive or negative. As $r_i = r_{ai} + e_i$, e_i is positive when $r_i \geq r_{ai}$ i.e. the estimated distance is greater than the actual distance. Similarly, e_i is negative when $r_i \leq r_{ai}$ i.e. the estimated distance is smaller than the actual distance. However, the localisation error e_l is only an absolute value. It has a magnitude but no sign. It is defined as the distance between the actual (x_a, y_a) and estimated (x, y) positions and is given by

$$e_l = \sqrt{(x - x_a)^2 + (y - y_a)^2} \tag{4.62}$$

Substituting (4.40) and (4.53) in (4.62)

$$e_l = \sqrt{E_x^2 + E_y^2} \tag{4.63}$$

In localisation experiments, we can determine the localisation error using (4.62) knowing the actual and estimated positions. With (4.63), we can calculate the localisation error using analytical results in (4.60) and (4.61) knowing actual distances and distance estimation errors and without knowing the actual and estimated positions.

As is evident from (4.40) and (4.53), E_x and E_y can be either positive or negative. The sign gives direction of the x and y components of localisation error. The direction of the whole localisation error e_l, which does not have a sign, is determined as following:

$$\theta_l = \tan^{-1} \frac{E_y}{E_x} \tag{4.64}$$

Hence, the analytical model completely specifies the error vector in terms of both magnitude and direction. The magnitude is given by e_l and the direction is given by θ_l.

The analytical model of multilateration localisation error developed in this section can be applied for localisation error analysis in all applications wherever multilateration is used for position estimation. This includes GNSS such as GPS and miscellaneous applications discussed in Section 4.1. However, in the next chapter, we restrict ourselves to localisation error analysis due to multilateration in wireless networks such as wireless sensor networks (WSN) and internet of things (IoT). In these networks, the distance estimation error is comparable to the actual distance itself.

The proposed analytical model has certain limitations. The model is limited only to two dimensions. Moreover, we consider multilateration using only three beacon nodes for the development of our analytical model.

4.5 Summary

We have shown that the overdetermined system of equations resulting from multilateration can be reduced to a pair of linear equations which can be solved using conventional techniques, such as Cramer's rule. We have also presented an accurate analytical model for multilateration error.

Chapter 5

Trilateration Error Analysis

We investigate localisation error and each of its two coordinate components, E_x and E_y, in four different ways. First we investigate individual error terms within E_1 and E_2 to investigate their impact on the estimated value of the coordinate. Second, we look at each of E_1 and E_2. Third we investigate the coordinate error term, E_x or E_y and evaluate its impact on the estimated value of the coordinate. Fourth, we analyse the localisation error e_l as a whole. Further, each of these four analyses and investigations comprise of two parts. In the first part, we determine conditions for which the error can be eliminated i.e. reduced to zero. In the second part, we investigate and analyse the value of the distance estimation error e_i for which a minimum or maximum in the error, E_x or E_y occurs.

5.1 Error Analysis

5.1.1 Case I: Individual error terms

From (4.49) and (4.50), we observe that there are three error terms responsible for localisation error in the estimated position: $2r_{a1}e_1 + e_1^2$, $2r_{a2}e_2 + e_2^2$ and $-2r_{a3}e_3 - e_3^2$. These error terms are of the form $2r_{ai}e_i + e_i^2$ $\forall i \in \{1, 2, 3\}$. If these error terms are zero, there will be no error in the estimated position. Therefore, if

$$2r_{ai}e_i + e_i^2 = 0 \quad \Rightarrow \quad e_i(2r_{ai} + e_i) = 0 \tag{5.1}$$

then either $e_i = 0$, or

$$2r_{ai} + e_i = 0 \quad \Rightarrow \quad e_i = -2r_{ai} \tag{5.2}$$

If $e_i = 0$, then there is no error in the estimated distance r_i i.e. $r_i = r_{ai}$ $\forall i \in \{1, 2, 3\}$. However, if $e_i = -2r_{ai}$, then the estimated distance is given by

$$r_i = r_{ai} + e_i = r_{ai} - 2r_{ai} = -r_{ai} \tag{5.3}$$

which is not possible as the estimated distance cannot be negative. This implies that a localisation error term $2r_{ai}e_i + e_i^2$ can be eliminated only if the corresponding distance estimation error e_i is eliminated.

Now, to determine the value of e_i at which the error term is minimum, let

$$\zeta_i = 2r_{ai}e_i + e_i^2 \quad \forall i \in \{1, 2, 3\} \tag{5.4}$$

At the minima, $\frac{d\zeta_i}{de_i} = 0$, therefore

$$\frac{d\zeta_i}{de_i} = 2r_{ai} + 2e_i = 0 \tag{5.5}$$

$$e_i = -r_{ai} \tag{5.6}$$

$$r_i = r_{ai} + e_i = 0 \tag{5.7}$$

Hence, the individual error term has a minimum when the estimated distance from the beacon node is taken as zero. Now, substituting $e_i = -r_{ai}$ from (5.6) in (5.4), the minimum possible value of the error term is given by

$$\zeta_{imin} = 2r_{ai}(-r_{ai}) + (-r_{ai})^2 \tag{5.8}$$

$$\zeta_{imin} = -r_{ai}^2 \tag{5.9}$$

As $r_{ai} \geq 0$, $\zeta_{imin} = -r_{ai}^2 \leq 0$ i.e. the minimum has a negative value. The error term cannot have a signed value lesser than this minimum. It is also to be noted that, the valid domain of the function ζ_i is $e_i \geq -r_{ai}$. To further explain this consider the relation between estimated distance r_i and actual distance r_{ai} and distance estimation error e_i i.e. $r_i = r_{ai} + e_i$. If $e_i < -r_{ai}$, then $r_i < 0$ i.e. the estimated distance becomes negative which is not possible. Therefore, valid domain of ζ_i is $e_i \geq -r_{ai}$. This also implies that the distance estimation error has no theoretical upper bound. However, in practical situations, the positive distance estimation error beyond a certain limit renders the localisation system useless. If we assume $-r_{ai} \leq e_i \leq r_{ai}$, then substituting the other extreme $e_i = r_{ai}$ in (5.4), the maximum value of the error term ζ_i is given by

$$\zeta_{imax} = 2r_{ai}(r_{ai}) + r_{ai}^2 \tag{5.10}$$

$$\zeta_{imax} = 3r_{ai}^2 \tag{5.11}$$

Comparing (5.9) and (5.11), it is observed that

$$|\zeta_{imax}| = 3|\zeta_{imin}| \tag{5.12}$$

In other words, considering the domain $-r_{ai} \leq e_i \leq r_{ai}$, the maximum value of the error term occurs at $e_i = r_{ai}$ and is three times the minimum value which occurs at $e_i = -r_{ai}$.

5.1.2 Case II: E_1 and E_2

Now let us consider E_1 and E_2, each of which is a combination of two error terms. If E_1 or E_2 is to be eliminated, then from (4.49) and (4.50)

$$E_1 = 2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2 = 0 \tag{5.13}$$

$$E_2 = 2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2 = 0 \tag{5.14}$$

If $r_{a1} = r_{a3}$ and $e_1 = e_3$, then $E_1 = 0$ so that from (4.44) $E_x = k_2E_2$ and from (4.57) $E_y = k_4E_2$. Similarly, if $r_{a2} = r_{a3}$ and $e_2 = e_3$, then $E_2 = 0$ so that $E_x = k_1E_1$ and $E_y = k_3E_1$. However, if $r_{a1} = r_{a2}$ and $e_1 = e_2$, then $E_1 = E_2$. In other words, if an unknown node has the same error in distance estimates from two equidistant neighbour beacon nodes, then either $E_1 = 0$ or $E_2 = 0$ or $E_1 = E_2$ and the localisation error is solely determined by either E_1 or E_2 or $E_1 = E_2$. Additionally, if the beacon nodes are positioned such that $|x_3 - x_2| = |y_3 - y_2|$ so that from (4.45) and (4.58) $|k_1| = |k_3|$ or such that $|x_3 - x_1| = |y_3 - y_1|$ so that from (4.46) and (4.59) $|k_2| = |k_4|$, then $|E_x| = |E_y|$.

As a result, from (4.63)

$$e_l = \sqrt{2}E_x = \sqrt{2}E_y \tag{5.15}$$

Now let us determine the values of errors e_1, e_2 and e_3 for which the functions E_1 and E_2 are minimum or maximum. From (4.49), we determine the critical points for E_1 as below:

$$\frac{\partial E_1}{\partial e_1} = 2r_{a1} + 2e_1 = 0 \quad \Rightarrow \quad e_1 = -r_{a1} \tag{5.16}$$

$$\frac{\partial E_1}{\partial e_3} = -2r_{a3} - 2e_3 = 0 \quad \Rightarrow \quad e_3 = -r_{a3} \tag{5.17}$$

Hence, the critical point is given by

$$(e_1, e_3) = (-r_{a1}, -r_{a3}) \tag{5.18}$$

which implies that

$$\begin{aligned} r_1 &= r_{a1} + e_1 = 0 \\ r_3 &= r_{a3} + e_3 = 0 \end{aligned} \tag{5.19}$$

As $r_{a1} \geq 0$ and $r_{a3} \geq 0$, therefore, $e_1 = -r_{a1} \leq 0$ and $e_3 = -r_{a3} \leq 0$. Hence, the critical point has negative coordinate values. Now the Hessian matrix of E_1 is given by

$$\mathbf{H}E_1 = \begin{bmatrix} \frac{\partial^2 E_1}{\partial e_1^2} & \frac{\partial^2 E_1}{\partial e_1 \partial e_3} \\ \frac{\partial^2 E_1}{\partial e_3 \partial e_1} & \frac{\partial^2 E_1}{\partial e_3^2} \end{bmatrix} \tag{5.20}$$

We calculate elements of the Hessian from (4.49) as below:

$$\frac{\partial E_1}{\partial e_1} = 2r_{a1} + 2e_1 \tag{5.21}$$

$$\frac{\partial E_1}{\partial e_3} = -2r_{a3} - 2e_3 \tag{5.22}$$

$$\frac{\partial^2 E_1}{\partial e_1^2} = 2 \tag{5.23}$$

$$\frac{\partial^2 E_1}{\partial e_3 \partial e_1} = 0 \tag{5.24}$$

$$\frac{\partial^2 E_1}{\partial e_1 \partial e_3} = 0 \tag{5.25}$$

$$\frac{\partial^2 E_1}{\partial e_3^2} = -2 \tag{5.26}$$

Substituting (5.23)-(5.26) in (5.20), we get

$$\mathbf{H}E_1 = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \tag{5.27}$$

Now

$$D_1 = \frac{\partial^2 E_1}{\partial e_1^2} \tag{5.28}$$

Therefore, from (5.23), at the critical point

$$D_1(-r_{a1}, -r_{a3}) = 2 \tag{5.29}$$

Furthermore, $D_2 = det\ \mathbf{H}E_1$. Therefore

$$D_2 = \begin{vmatrix} 2 & 0 \\ 0 & -2 \end{vmatrix} \tag{5.30}$$

At the critical point

$$D_2(-r_{a1}, -r_{a3}) = -4 \tag{5.31}$$

As $D_2 < 0$, E_1 has a saddle point at $(-r_{a1}, -r_{a3})$. We substitute the critical point from (5.18) in (4.49) to determine the E_1 at the saddle point:

$$E_{1sad} = r_{a3}^2 - r_{a1}^2 \tag{5.32}$$

It is to be noted that $E_{1sad} = 0$ if $r_{a1} = r_{a3}$.

The critical and saddle points for E_2 can be calculated similarly. We give the complete derivation here for the sake of completeness. From (4.50), we calculate the critical point for E_2 as below:

$$\frac{\partial E_2}{\partial e_2} = 2r_{a2} + 2e_2 = 0 \quad \Rightarrow \quad e_2 = -r_{a2} \tag{5.33}$$

$$\frac{\partial E_2}{\partial e_3} = -2r_{a3} - 2e_3 = 0 \quad \Rightarrow \quad e_3 = -r_{a3} \tag{5.34}$$

Hence, the critical point is given by

$$(e_2, e_3) = (-r_{a2}, -r_{a3}) \tag{5.35}$$

which implies that

$$\begin{aligned} r_2 &= r_{a2} + e_2 = 0 \\ r_3 &= r_{a3} + e_3 = 0 \end{aligned} \tag{5.36}$$

The Hessian matrix of E_2 is given by

$$\mathbf{H}E_2 = \begin{bmatrix} \frac{\partial^2 E_2}{\partial e_2^2} & \frac{\partial^2 E_2}{\partial e_2 \partial e_3} \\ \frac{\partial^2 E_2}{\partial e_3 \partial e_2} & \frac{\partial^2 E_2}{\partial e_3^2} \end{bmatrix} \tag{5.37}$$

We calculate elements of the Hessian from (4.50) as below:

$$\frac{\partial E_2}{\partial e_2} = 2r_{a2} + 2e_2 \tag{5.38}$$

$$\frac{\partial E_2}{\partial e_3} = -2r_{a3} - 2e_3 \tag{5.39}$$

$$\frac{\partial^2 E_2}{\partial e_2^2} = 2 \tag{5.40}$$

$$\frac{\partial^2 E_2}{\partial e_3 \partial e_2} = 0 \tag{5.41}$$

$$\frac{\partial^2 E_2}{\partial e_2 \partial e_3} = 0 \tag{5.42}$$

$$\frac{\partial^2 E_2}{\partial e_3^2} = -2 \tag{5.43}$$

Substituting (5.40)-(5.43) in (5.37), we get

$$\mathbf{H}E_2 = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} \tag{5.44}$$

To test for minima and maxima, we have

$$D_1 = \frac{\partial^2 E_2}{\partial e_2^2} \tag{5.45}$$

Using (5.40) in (5.45), we get D_1 at the critical point

$$D_1(-r_{a2}, -r_{a3}) = 2 \tag{5.46}$$

As $D_2 = det\ \mathbf{H}E_2$,

$$D_2 = \begin{vmatrix} 2 & 0 \\ 0 & -2 \end{vmatrix} \tag{5.47}$$

At the critical point for E_2, we have

$$D_2(-r_{a2}, -r_{a3}) = -4 \tag{5.48}$$

It can be seen that $D_2 < 0$. Therefore, E_2 has a saddle point at $(-r_{a2}, -r_{a3})$. Substituting the critical point from (5.35) in (4.50), we determine E_2 at the saddle point:

$$E_{2sad} = r_{a3}^2 - r_{a2}^2 \tag{5.49}$$

Again, it is to be noted that $E_{2sad} = 0$ if $r_{a2} = r_{a3}$.

Results in (5.32) and (5.49) give the values of E_1 and E_2 at the critical points which are one of the possible combinations of extreme values of distance estimation errors. For each of E_1 and E_2, there are four such possible combinations over the domain $-r_{ai} \leq e_i \leq r_{ai}$. We calculate E_1 and E_2 using (4.49) and (4.50) for all the possible combinations of extreme values of distance estimation errors and give the results in Table 5.1 and Table 5.2. As noted earlier, $r_{ai} \geq 0$. Therefore, from Table 5.1, we observe that E_1 has a minimum value at $(e_1, e_3) = (-r_{a1}, r_{a3})$ and is given by

$$E_{1min} = -3r_{a3}^2 - r_{a1}^2 \tag{5.50}$$

Table 5.1: Possible extreme values of E_1.

Estimation errors		Estimated distances		Error component
e_1	e_3	r_1	r_3	E_1
$-r_{a1}$	$-r_{a3}$	0	0	$r_{a3}^2 - r_{a1}^2$
$-r_{a1}$	r_{a3}	0	$2r_{a3}$	$-3r_{a3}^2 - r_{a1}^2$
r_{a1}	$-r_{a3}$	$2r_{a1}$	0	$r_{a3}^2 + 3r_{a1}^2$
r_{a1}	r_{a3}	$2r_{a1}$	$2r_{a3}$	$-3(r_{a3}^2 - r_{a1}^2)$

The maximum value of E_1 occurs at $(e_1, e_3) = (r_{a1}, -r_{a3})$ and is given by

$$E_{1max} = r_{a3}^2 + 3r_{a1}^2 \tag{5.51}$$

Similarly, from Table 5.2, we note that the minimum and maximum values of E_2 occur at $(e_2, e_3) = (-r_{a2}, r_{a3})$ and $(e_2, e_3) = (r_{a2}, -r_{a3})$ respectively and are given by

$$E_{2min} = -3r_{a3}^2 - r_{a2}^2 \tag{5.52}$$

$$E_{2max} = r_{a3}^2 + 3r_{a2}^2 \tag{5.53}$$

From Table 5.1 and Table 5.2, it is also noted that the extreme values of E_1 and E_2 occur when the distance estimation errors have opposite signs. For example, the minimum value of E_1 occurs when e_1 is negative and e_3 is positive, and the maximum value of E_1 occurs when e_1 is positive and e_3 is negative. The value of E_1 or E_2 lies between extreme values when the distance estimation errors have the same sign. Therefore, we can infer that for a given set of magnitudes of distance estimation errors, the corresponding localisation error is not extreme and lies between the extreme values if all the distance estimation errors have the same sign. From Table 5.1, E_1 due to negative distance estimation errors

Table 5.2: Possible extreme values of E_2.

Estimation errors		Estimated distances		Error component
e_2	e_3	r_2	r_3	E_2
$-r_{a2}$	$-r_{a3}$	0	0	$r_{a3}^2 - r_{a2}^2$
$-r_{a2}$	r_{a3}	0	$2r_{a3}$	$-3r_{a3}^2 - r_{a2}^2$
r_{a2}	$-r_{a3}$	$2r_{a2}$	0	$r_{a3}^2 + 3r_{a2}^2$
r_{a2}	r_{a3}	$2r_{a2}$	$2r_{a3}$	$-3(r_{a3}^2 - r_{a2}^2)$

$(-r_{a1}, -r_{a3})$ is given by

$$E_{1^-} = r_{a3}^2 - r_{a1}^2 \tag{5.54}$$

E_1 due to the positive distance estimation errors (r_{a1}, r_{a3}) is given by

$$E_{1^+} = -3(r_{a3}^2 - r_{a1}^2) \tag{5.55}$$

Dividing (5.55) by (5.54)

$$E_{1^+} = -3E_{1^-} \tag{5.56}$$

Similarly, from Table 5.2

$$E_{2^-} = r_{a3}^2 - r_{a2}^2 \tag{5.57}$$

$$E_{2^+} = -3(r_{a3}^2 - r_{a2}^2) \tag{5.58}$$

$$E_{2^+} = -3E_{2^-} \tag{5.59}$$

It can be noted from Table 5.1 that $E_1 = 0$ at both negative $(-r_{a1}, -r_{a3})$ and positive (r_{a1}, r_{a3}) points of distance estimation errors if $r_{a1} = r_{a3}$ i.e. if two of the neighbour beacon nodes are equidistant. Similarly, from Table 5.2, we can infer that $E_2 = 0$ at $(-r_{a2}, -r_{a3})$ and (r_{a2}, r_{a3}) if $r_{a2} = r_{a3}$.

5.1.3 Case III: E_x and E_y

We now consider the coordinate errors, E_x and E_y, which is error in the x and y coordinates. If $E_x = 0$ or $E_y = 0$, then from (4.60) and (4.61)

$$\begin{aligned} E_x &= k_1[2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2] \\ &+ k_2[2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2] = 0 \end{aligned} \tag{5.60}$$

$$E_y = k_3[2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2] \\ + k_4[2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2] = 0 \tag{5.61}$$

From (5.60) and (5.61), it is evident that both $E_x = 0$ and $E_y = 0$ if

$$e_1 = e_2 = e_3 \quad \text{and} \quad r_{a1} = r_{a2} = r_{a3} \tag{5.62}$$

which implies that $r_1 = r_2 = r_3$. In other words, if the estimated distances have the same amount of error ($e_1 = e_2 = e_3$), then the net localisation error in the estimated position will be zero if the unknown node is at the same distance ($r_{a1} = r_{a2} = r_{a3}$) from all the neighbour beacon nodes. This also means that if an unknown node can determine that it is equidistant from three neighbour beacon nodes, then it can determine its exact position by merely using the position information of beacon nodes and without estimating distances from them. For example, the unknown node can use any arbitrary constant value r_a as distance estimate for all the equidistant beacon nodes ($r_{a1} = r_{a2} = r_{a3} = r_a$) to estimate position. From these results we infer that

(i) a correct position estimation with zero localisation error does not necessarily imply that the estimated distances used in the position estimation are accurate,

(ii) accurate position can be obtained even if the estimated distances are not accurate, and that

(iii) it is possible for an unknown node to determine its exact position without knowing its distances from the neighbour beacon nodes.

Let us again consider (4.60) and let

$$k_1 = -k_2 \tag{5.63}$$

so that from (4.45) and (4.46)

$$\frac{1}{2\Delta}(y_3 - y_2) = \frac{1}{2\Delta}(y_3 - y_1) \tag{5.64}$$

$$y_1 = y_2 \tag{5.65}$$

Now substituting (5.63) in (5.60)

$$\begin{aligned} &+ k_1[2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2] \\ &- k_1[2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2] = 0 \end{aligned} \tag{5.66}$$

$$2r_{a1}e_1 + e_1^2 = 2r_{a2}e_2 + e_2^2 \tag{5.67}$$

Therefore, $E_x = 0$, if

$$e_1 = e_2 \quad \text{and} \quad r_{a1} = r_{a2} \tag{5.68}$$

so that $r_1 = r_2$. We infer that if two of the neighbour beacon nodes are collinear parallel to the x-axis and are at the same distance from the unknown node and distance estimates for them have the same amount of error, then x component of the localisation error is zero irrespective of the position and distance estimation error of the third beacon node.

To derive a similar result for the y component of the localisation error, let us consider (4.61) and let

$$k_3 = -k_4 \tag{5.69}$$

so that from (4.58) and (4.59)

$$\frac{1}{2\Delta}(x_3 - x_2) = \frac{1}{2\Delta}(x_3 - x_1) \tag{5.70}$$

$$x_1 = x_2 \tag{5.71}$$

Substituting (5.69) in (5.61)

$$\begin{aligned} &+ k_3[2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2] \\ &- k_3[2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2] = 0 \end{aligned} \tag{5.72}$$

$$2r_{a1}e_1 + e_1^2 = 2r_{a2}e_2 + e_2^2 \tag{5.73}$$

Therefore, $E_y = 0$, if

$$e_1 = e_2 \quad \text{and} \quad r_{a1} = r_{a2} \tag{5.74}$$

so that $r_1 = r_2$. These are the same results that we derived earlier for E_x in (5.67) and (5.68). Hence, we can generalise that if two of the neighbour beacon nodes are collinear parallel to an axis and are at the same distance from the unknown node and distance estimates for them have the same amount of error, then the component of localisation error for that axis is zero irrespective of the position and distance estimation error of the third beacon node.

5.1.3.1 Extrema for E_x

Let us now determine the minima and maxima for E_x and E_y. At the extrema for E_x

$$\frac{\partial E_x}{\partial e_1} = 0, \quad \frac{\partial E_x}{\partial e_2} = 0 \quad \text{and} \quad \frac{\partial E_x}{\partial e_3} = 0 \tag{5.75}$$

From (4.60)

$$\frac{\partial E_x}{\partial e_1} = k_1[2r_{a1} + 2e_1] = 0 \Rightarrow e_1 = -r_{a1} \tag{5.76}$$

$$\frac{\partial E_x}{\partial e_2} = k_2[2r_{a2} + 2e_2] = 0 \Rightarrow e_2 = -r_{a2} \tag{5.77}$$

$$\frac{\partial E_x}{\partial e_3} = k_1[-2r_{a3} - 2e_3] + k_2[-2r_{a3} - 2e_3] = 0$$

$$\begin{aligned} r_{a3} + e_3 &= -\frac{k_2}{k_1}(r_{a3} + e_3) \\ e_3 + \frac{k_2}{k_1}e_3 &= -r_{a3} - \frac{k_2}{k_1}r_{a3} \\ e_3(1 + \frac{k_2}{k_1}) &= -r_{a3}(1 + \frac{k_2}{k_1}) \end{aligned} \tag{5.78}$$

$$e_3 = -r_{a3} \tag{5.79}$$

Hence, E_x has a critical point at

$$(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, -r_{a3}) \tag{5.80}$$

so that

$$r_i = r_{ai} + e_i = 0 \quad \forall i \in \{1, 2, 3\} \tag{5.81}$$

Note that the error given in (5.80) and (5.81) is the maximum possible negative value of error in the estimated distances. If the negative value of distance estimation error becomes greater than this, then the estimated distances become negative i.e. if $e_i < -r_{ai}$, then $r_i = r_{ai} + e_i < 0$, which is not possible. Now the Hessian matrix of E_x is given by

$$\mathbf{H}E_x = \begin{bmatrix} \frac{\partial^2 E_x}{\partial e_1^2} & \frac{\partial^2 E_x}{\partial e_1 \partial e_2} & \frac{\partial^2 E_x}{\partial e_1 \partial e_3} \\ \frac{\partial^2 E_x}{\partial e_2 \partial e_1} & \frac{\partial^2 E_x}{\partial e_2^2} & \frac{\partial^2 E_x}{\partial e_2 \partial e_3} \\ \frac{\partial^2 E_x}{\partial e_3 \partial e_1} & \frac{\partial^2 E_x}{\partial e_3 \partial e_2} & \frac{\partial^2 E_x}{\partial e_3^2} \end{bmatrix} \tag{5.82}$$

We calculate elements of the Hessian from (4.60) as below.

$$\frac{\partial E_x}{\partial e_1} = k_1[2r_{a1} + 2e_1] \tag{5.83}$$

$$\frac{\partial E_x}{\partial e_2} = k_2[2r_{a2} + 2e_2] \tag{5.84}$$

$$\frac{\partial E_x}{\partial e_3} = k_1[-2r_{a3} - 2e_3] + k_2[-2r_{a3} - 2e_3] \tag{5.85}$$

$$\frac{\partial^2 E_x}{\partial e_1^2} = 2k_1 \tag{5.86}$$

$$\frac{\partial^2 E_x}{\partial e_2 \partial e_1} = 0 \tag{5.87}$$

$$\frac{\partial^2 E_x}{\partial e_3 \partial e_1} = 0 \tag{5.88}$$

$$\frac{\partial^2 E_x}{\partial e_1 \partial e_2} = 0 \tag{5.89}$$

$$\frac{\partial^2 E_x}{\partial e_2^2} = 2k_2 \tag{5.90}$$

$$\frac{\partial^2 E_x}{\partial e_3 \partial e_2} = 0 \tag{5.91}$$

$$\frac{\partial^2 E_x}{\partial e_1 \partial e_3} = 0 \tag{5.92}$$

$$\frac{\partial^2 E_x}{\partial e_2 \partial e_3} = 0 \tag{5.93}$$

$$\frac{\partial^2 E_x}{\partial e_3^2} = -2k_1 - 2k_2 = -2(k_1 + k_2) \tag{5.94}$$

Substituting (5.86)-(5.94) in (5.82), we get

$$\mathbf{H}E_x = \begin{bmatrix} 2k_1 & 0 & 0 \\ 0 & 2k_2 & 0 \\ 0 & 0 & -2(k_1 + k_2) \end{bmatrix} \tag{5.95}$$

Now

$$D_{x1} = \frac{\partial^2 E_x}{\partial e_1^2} \tag{5.96}$$

Therefore, from (5.86) at the critical point

$$D_{x1}(-r_{a1}, -r_{a2}, -r_{a3}) = 2k_1 \tag{5.97}$$

Similarly

$$D_{x2} = \begin{vmatrix} \frac{\partial^2 E_x}{\partial e_1^2} & \frac{\partial^2 E_x}{\partial e_1 \partial e_2} \\ \frac{\partial^2 E_x}{\partial e_2 \partial e_1} & \frac{\partial^2 E_x}{\partial e_2^2} \end{vmatrix} \tag{5.98}$$

Substituting (5.86), (5.87), (5.89) and (5.90) in (5.98)

$$D_{x2} = \begin{vmatrix} 2k_1 & 0 \\ 0 & 2k_2 \end{vmatrix} \tag{5.99}$$

Therefore, at the critical point

$$D_{x2}(-r_{a1}, -r_{a2}, -r_{a3}) = 4k_1k_2 \tag{5.100}$$

Now $D_{x3} = det\ \mathbf{H}E_x$, therefore

$$D_{x3} = \begin{vmatrix} 2k_1 & 0 & 0 \\ 0 & 2k_2 & 0 \\ 0 & 0 & -2(k_1 + k_2) \end{vmatrix} \tag{5.101}$$

At the critical point

$$D_{x3}(-r_{a1}, -r_{a2}, -r_{a3}) = -8k_1k_2(k_1 + k_2) \tag{5.102}$$

The values of D_{x1}, D_{x2} and D_{x3} are determined by k_1 and k_2 as can be seen from (5.97), (5.100) and (5.102). For example, from (5.97) it is obvious that $D_{x1} > 0$ if $k_1 > 0$. From (5.100) we can see that $D_{x2} > 0$ if $k_2 > 0$ also. However $D_{x3} < 0$ if $k_1 > 0$ and $k_2 > 0$ as is evident from (5.102). Table 5.3 lists the outcomes for D_{x1}, D_{x2} and D_{x3} for all the different possible combinations of values of k_1 and k_2. From all the possible output combinations of D_{x1}, D_{x2} and D_{x3}, it can be concluded that (5.80) is a saddle point. Substituting (5.80) in (4.60), we get the value of E_x at the saddle point.

$$\begin{aligned} E_{xsad} =& k_1[2r_{a1}(-r_{a1}) + (-r_{a1})^2 - 2r_{a3}(-r_{a3}) - (-r_{a3})^2] \\ &+ k_2[2r_{a2}(-r_{a2}) + (-r_{a2})^2 - 2r_{a3}(-r_{a3}) - (-r_{a3})^2] \end{aligned} \tag{5.103}$$

so that

$$E_{xsad} = k_1[r_{a3}^2 - r_{a1}^2] + k_2[r_{a3}^2 - r_{a2}^2] \tag{5.104}$$

Note that $E_{xsad} = 0$, if $r_{a1} = r_{a2} = r_{a3}$.

Table 5.3: D_{x1}, D_{x2} and D_{x3} for different values of k_1 and k_2.

k_1	k_2	D_{x1}	D_{x2}	D_{x3}
$k_1 < 0$	$k_2 < 0$	$D_{x1} < 0$	$D_{x2} > 0$	$D_{x3} > 0$
$k_1 < 0$	$k_2 > 0$	$D_{x1} < 0$	$D_{x2} < 0$	$D_{x3} > 0$ if $\lvert k_2\rvert > \lvert k_1\rvert$ $D_{x3} < 0$ if $\lvert k_2\rvert < \lvert k_1\rvert$
$k_1 > 0$	$k_2 < 0$	$D_{x1} > 0$	$D_{x2} < 0$	$D_{x3} > 0$ if $\lvert k_1\rvert > \lvert k_2\rvert$ $D_{x3} < 0$ if $\lvert k_1\rvert < \lvert k_2\rvert$
$k_1 > 0$	$k_2 > 0$	$D_{x1} > 0$	$D_{x2} > 0$	$D_{x3} < 0$

5.1.3.2 Extrema for E_y

Let us now consider the extrema for E_y. The derivations are similar to those for E_x, but we give them here for the sake of completeness. At the extrema

$$\frac{\partial E_y}{\partial e_1} = 0, \quad \frac{\partial E_y}{\partial e_2} = 0 \quad \text{and} \quad \frac{\partial E_y}{\partial e_3} = 0 \tag{5.105}$$

From (4.61)

$$\frac{\partial E_y}{\partial e_1} = k_3[2r_{a1} + 2e_1] = 0 \Rightarrow e_1 = -r_{a1} \tag{5.106}$$

$$\frac{\partial E_y}{\partial e_2} = k_4[2r_{a2} + 2e_2] = 0 \Rightarrow e_2 = -r_{a2} \tag{5.107}$$

$$\frac{\partial E_y}{\partial e_3} = k_3[-2r_{a3} - 2e_3] + k_4[-2r_{a3} - 2e_3] = 0$$

$$\begin{aligned} r_{a3} + e_3 &= -\frac{k_4}{k_3}(r_{a3} + e_3) \\ e_3 + \frac{k_4}{k_3}e_3 &= -r_{a3} - \frac{k_4}{k_3}r_{a3} \\ e_3(1 + \frac{k_4}{k_3}) &= -r_{a3}(1 + \frac{k_4}{k_3}) \end{aligned} \tag{5.108}$$

$$e_3 = -r_{a3} \tag{5.109}$$

Hence, E_y has a critical point at

$$(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, -r_{a3}) \tag{5.110}$$

implying that

$$r_i = r_{ai} + e_i \quad \forall i \in \{1, 2, 3\} \tag{5.111}$$

This is the same result as obtained in (5.80) and (5.81) for E_x. Therefore, the possible extrema for both the components of localisation error, E_x and E_y, occurs simultaneously. Now the Hessian matrix of E_y is given by

$$\mathbf{H}E_y = \begin{bmatrix} \frac{\partial^2 E_y}{\partial e_1^2} & \frac{\partial^2 E_y}{\partial e_1 \partial e_2} & \frac{\partial^2 E_y}{\partial e_1 \partial e_3} \\ \frac{\partial^2 E_y}{\partial e_2 \partial e_1} & \frac{\partial^2 E_y}{\partial e_2^2} & \frac{\partial^2 E_y}{\partial e_2 \partial e_3} \\ \frac{\partial^2 E_y}{\partial e_3 \partial e_1} & \frac{\partial^2 E_y}{\partial e_3 \partial e_2} & \frac{\partial^2 E_y}{\partial e_3^2} \end{bmatrix} \tag{5.112}$$

We calculate elements of the Hessian from (4.61) as below.

$$\frac{\partial E_y}{\partial e_1} = k_3[2r_{a1} + 2e_1] \tag{5.113}$$

$$\frac{\partial E_y}{\partial e_2} = k_4[2r_{a2} + 2e_2] \tag{5.114}$$

$$\frac{\partial E_y}{\partial e_3} = k_3[-2r_{a3} - 2e_3] + k_4[-2r_{a3} - 2e_3] \tag{5.115}$$

$$\frac{\partial^2 E_y}{\partial e_1^2} = 2k_3 \tag{5.116}$$

$$\frac{\partial^2 E_y}{\partial e_2 \partial e_1} = 0 \tag{5.117}$$

$$\frac{\partial^2 E_y}{\partial e_3 \partial e_1} = 0 \tag{5.118}$$

$$\frac{\partial^2 E_y}{\partial e_1 \partial e_2} = 0 \tag{5.119}$$

$$\frac{\partial^2 E_y}{\partial e_2^2} = 2k_4 \tag{5.120}$$

$$\frac{\partial^2 E_y}{\partial e_3 \partial e_2} = 0 \tag{5.121}$$

$$\frac{\partial^2 E_y}{\partial e_1 \partial e_3} = 0 \tag{5.122}$$

$$\frac{\partial^2 E_y}{\partial e_2 \partial e_3} = 0 \tag{5.123}$$

$$\frac{\partial^2 E_y}{\partial e_3^2} = -2k_3 - 2k_4 = -2(k_3 + k_4) \tag{5.124}$$

Substituting (5.116)-(5.124) in (5.112), we get

$$\mathbf{H}E_y = \begin{bmatrix} 2k_3 & 0 & 0 \\ 0 & 2k_4 & 0 \\ 0 & 0 & -2(k_3 + k_4) \end{bmatrix} \tag{5.125}$$

Now

$$D_{y1} = \frac{\partial^2 E_y}{\partial e_1^2} = 2k_3 \tag{5.126}$$

Therefore, from (5.116) at the critical point

$$D_{y1}(-r_{a1}, -r_{a2}, -r_{a3}) = 2k_3 \tag{5.127}$$

Furthermore

$$D_{y2} = \begin{vmatrix} \frac{\partial^2 E_y}{\partial e_1^2} & \frac{\partial^2 E_y}{\partial e_1 \partial e_2} \\ \frac{\partial^2 E_y}{\partial e_2 \partial e_1} & \frac{\partial^2 E_y}{\partial e_2^2} \end{vmatrix} \tag{5.128}$$

Substituting (5.116), (5.117), (5.119) and (5.120) in (5.128)

$$D_{y2} = \begin{vmatrix} 2k_3 & 0 \\ 0 & 2k_4 \end{vmatrix} \tag{5.129}$$

Therefore, at the critical point

$$D_{y2}(-r_{a1}, -r_{a2}, -r_{a3}) = 4k_3 k_4 \tag{5.130}$$

Now $D_{y3} = det\ \mathbf{H}E_y$, therefore

$$D_{y3} = \begin{vmatrix} 2k_3 & 0 & 0 \\ 0 & 2k_4 & 0 \\ 0 & 0 & -2(k_3 + k_4) \end{vmatrix} \tag{5.131}$$

At the critical point

$$D_{y3}(-r_{a1}, -r_{a2}, -r_{a3}) = -8k_3k_4(k_3 + k_4) \tag{5.132}$$

The values of D_{y1}, D_{y2} and D_{y3} are determined by k_3 and k_4 as is evident from (5.127), (5.130) and (5.132). Outcomes for D_{y1}, D_{y2} and D_{y3} for different possible combinations of values of k_3 and k_4 are listed in Table 5.4. From these results it can be concluded that (5.110) is a saddle point. Substituting (5.110) in (4.61) we get the value of E_y at the saddle point.

$$\begin{aligned} E_{ysad} =& k_3[2r_{a1}(-r_{a1}) + (-r_{a1})^2 - 2r_{a3}(-r_{a3}) - (-r_{a3})^2] \\ &+ k_4[2r_{a2}(-r_{a2}) + (-r_{a2})^2 - 2r_{a3}(-r_{a3}) - (-r_{a3})^2] \end{aligned} \tag{5.133}$$

so that

$$E_{ysad} = k_3[r_{a3}^2 - r_{a1}^2] + k_4[r_{a3}^2 - r_{a2}^2] \tag{5.134}$$

Table 5.4: D_{y1}, D_{y2} and D_{y3} for different values of k_3 and k_4.

k_3	k_4	D_{y1}	D_{y2}	D_{y3}
$k_3 < 0$	$k_4 < 0$	$D_{y1} < 0$	$D_{y2} > 0$	$D_{y3} > 0$
$k_3 < 0$	$k_4 > 0$	$D_{y1} < 0$	$D_{y2} < 0$	$D_{y3} > 0$ if $\lvert k_4\rvert > \lvert k_3\rvert$ $D_{y3} < 0$ if $\lvert k_4\rvert < \lvert k_3\rvert$
$k_3 > 0$	$k_4 < 0$	$D_{y1} > 0$	$D_{y2} < 0$	$D_{y3} > 0$ if $\lvert k_3\rvert > \lvert k_4\rvert$ $D_{y3} < 0$ if $\lvert k_3\rvert < \lvert k_4\rvert$
$k_3 > 0$	$k_4 > 0$	$D_{y1} > 0$	$D_{y2} > 0$	$D_{y3} < 0$

Again note that $E_{ysad} = 0$, if $r_{a1} = r_{a2} = r_{a3}$.

The results in (5.104) and (5.134) give E_x and E_y at the critical point which is one of the possible combinations of extreme values of distance estimation errors. For each of E_x and E_y, there are eight such possible combinations over the domain $-r_{ai} \le e_i \le r_{ai}$. We calculate E_x and E_y using (4.60) and (4.61) for all the possible combinations of extreme values and list the results in Table 5.5 and Table 5.6. The entries in column E_x in Table 5.5 and column E_y in Table

Table 5.5: Possible extreme values of E_x.

Estimation errors			Estimated distances			Localisation error x-coordinate
e_1	e_2	e_3	r_1	r_2	r_3	E_x
$-r_{a1}$	$-r_{a2}$	$-r_{a3}$	0	0	0	$k_1[r_{a3}^2 - r_{a1}^2] + k_2[r_{a3}^2 - r_{a2}^2]$
$-r_{a1}$	$-r_{a2}$	r_{a3}	0	0	$2r_{a3}$	$-k_1[3r_{a3}^2 + r_{a1}^2] - k_2[3r_{a3}^2 + r_{a2}^2]$
$-r_{a1}$	r_{a2}	$-r_{a3}$	0	$2r_{a2}$	0	$k_1[r_{a3}^2 - r_{a1}^2] + k_2[r_{a3}^2 + 3r_{a2}^2]$
$-r_{a1}$	r_{a2}	r_{a3}	0	$2r_{a2}$	$2r_{a3}$	$-k_1[3r_{a3}^2 + r_{a1}^2] - 3k_2[r_{a3}^2 - r_{a2}^2]$
r_{a1}	$-r_{a2}$	$-r_{a3}$	$2r_{a1}$	0	0	$k_1[r_{a3}^2 + 3r_{a1}^2] + k_2[r_{a3}^2 - r_{a2}^2]$
r_{a1}	$-r_{a2}$	r_{a3}	$2r_{a1}$	0	$2r_{a3}$	$-3k_1[r_{a3}^2 - r_{a1}^2] - k_2[3r_{a3}^2 + r_{a2}^2]$
r_{a1}	r_{a2}	$-r_{a3}$	$2r_{a1}$	$2r_{a2}$	0	$k_1[r_{a3}^2 + 3r_{a1}^2] + k_2[r_{a3}^2 + 3r_{a2}^2]$
r_{a1}	r_{a2}	r_{a3}	$2r_{a1}$	$2r_{a2}$	$2r_{a3}$	$-3k_1[r_{a3}^2 - r_{a1}^2] - 3k_2[r_{a3}^2 - r_{a2}^2]$

Table 5.6: Possible extreme values of E_y.

Estimation errors			Estimated distances			Localisation error y-coordinate
e_1	e_2	e_3	r_1	r_2	r_3	E_y
$-r_{a1}$	$-r_{a2}$	$-r_{a3}$	0	0	0	$k_3[r_{a3}^2 - r_{a1}^2] + k_4[r_{a3}^2 - r_{a2}^2]$
$-r_{a1}$	$-r_{a2}$	r_{a3}	0	0	$2r_{a3}$	$-k_3[3r_{a3}^2 + r_{a1}^2] - k_4[3r_{a3}^2 + r_{a2}^2]$
$-r_{a1}$	r_{a2}	$-r_{a3}$	0	$2r_{a2}$	0	$k_3[r_{a3}^2 - r_{a1}^2] + k_4[r_{a3}^2 + 3r_{a2}^2]$
$-r_{a1}$	r_{a2}	r_{a3}	0	$2r_{a2}$	$2r_{a3}$	$-k_3[3r_{a3}^2 + r_{a1}^2] - 3k_4[r_{a3}^2 - r_{a2}^2]$
r_{a1}	$-r_{a2}$	$-r_{a3}$	$2r_{a1}$	0	0	$k_3[r_{a3}^2 + 3r_{a1}^2] + k_4[r_{a3}^2 - r_{a2}^2]$
r_{a1}	$-r_{a2}$	r_{a3}	$2r_{a1}$	0	$2r_{a3}$	$-3k_3[r_{a3}^2 - r_{a1}^2] - k_4[3r_{a3}^2 + r_{a2}^2]$
r_{a1}	r_{a2}	$-r_{a3}$	$2r_{a1}$	$2r_{a2}$	0	$k_3[r_{a3}^2 + 3r_{a1}^2] + k_4[r_{a3}^2 + 3r_{a2}^2]$
r_{a1}	r_{a2}	r_{a3}	$2r_{a1}$	$2r_{a2}$	$2r_{a3}$	$-3k_3[r_{a3}^2 - r_{a1}^2] - 3k_4[r_{a3}^2 - r_{a2}^2]$

5.6 differ only due to the localisation constants k_1, k_2, k_3 and k_4. Furthermore, the signs of E_x and E_y, and which of the eight entries is an extremum also depends upon the localisation constants. For example, if $k_1 > 0$ and $k_2 > 0$, then the minimum for E_x is located at $(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, r_{a3})$ and is given by

$$E_{xmin} = -k_1[3r_{a3}^2 + r_{a1}^2] - k_2[3r_{a3}^2 + r_{a2}^2] \tag{5.135}$$

while the maximum located at $(r_{a1}, r_{a2}, -r_{a3})$ is given by

$$E_{xmax} = k_1[r_{a3}^2 + 3r_{a1}^2] + k_2[r_{a3}^2 + 3r_{a2}^2] \tag{5.136}$$

Similarly, if $k_3 > 0$ and $k_4 > 0$, the minimum for E_y is located at $(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, r_{a3})$ and is given by

$$E_{ymin} = -k_3[3r_{a3}^2 + r_{a1}^2] - k_4[3r_{a3}^2 + r_{a2}^2] \tag{5.137}$$

while the maximum located at $(r_{a1}, r_{a2}, -r_{a3})$ is given by

$$E_{ymax} = k_3[r_{a3}^2 + 3r_{a1}^2] + k_4[r_{a3}^2 + 3r_{a2}^2] \tag{5.138}$$

We observe that the extrema for E_x and E_y occur simultaneously at the same values of distance estimation errors if all the localisation constants have the same sign and the minimum point $(-r_{a1}, -r_{a2}, r_{a3})$ is exactly opposite to the maximum point $(r_{a1}, r_{a2}, -r_{a3})$. Moreover, the minimum point $(-r_{a1}, -r_{a2}, r_{a3})$ for E_x and E_y corresponds to points $(-r_{a1}, r_{a3})$ and $(-r_{a2}, r_{a3})$ at which E_1 and E_2 are minimum. Similarly, the maximum point $(r_{a1}, r_{a2}, -r_{a3})$ for E_x and E_y corresponds to points $(r_{a1}, -r_{a3})$ and $(r_{a2}, -r_{a3})$ at which E_1 and E_2 are maximum. Further note that if $k_1 < 0$, $k_2 < 0$, $k_3 < 0$

and $k_4 < 0$ then the minimum values of E_x and E_y are located at $(r_{a1}, r_{a2}, -r_{a3})$ while their maximum values occur at $(-r_{a1}, -r_{a2}, r_{a3})$. The minimum value of E_x is as in (5.136) so that

$$E_{xmin} = k_1[r_{a3}^2 + 3r_{a1}^2] + k_2[r_{a3}^2 + 3r_{a2}^2] \tag{5.139}$$

while the maximum value of E_x is as in (5.135)

$$E_{xmax} = -k_1[3r_{a3}^2 + r_{a1}^2] - k_2[3r_{a3}^2 + r_{a2}^2]. \tag{5.140}$$

Similarly, the minimum value of E_y is given by (5.138) so that

$$E_{ymin} = k_3[r_{a3}^2 + 3r_{a1}^2] + k_4[r_{a3}^2 + 3r_{a2}^2] \tag{5.141}$$

while the maximum value is as in (5.137) and is given by

$$E_{ymax} = -k_3[3r_{a3}^2 + r_{a1}^2] - k_4[3r_{a3}^2 + r_{a2}^2]. \tag{5.142}$$

If the localisation constants do not have the same sign, the maximum and minimum values of E_x and E_y are given by appropriate entries in column E_x in Table 5.5 and column E_y in Table 5.6 depending upon the relative magnitudes of r_{ai} $\forall i \in \{1, 2, 3\}$ and the relative signs and magnitudes of localisation constants. It is then not necessary that the minimum and maximum values of E_1, E_2, E_x, E_y and e_l lie at the same point as is the case when all the localisation constants have the same sign. To determine the condition under which all the localisation constants have the same sign, let us consider (4.45), (4.46), (4.58) and (4.59). If $k_1 > 0$, $k_2 > 0$, $k_3 > 0$, and $k_4 > 0$, then

$$\frac{1}{2\Delta}(y_3 - y_2) > 0 \tag{5.143}$$

$$-\frac{1}{2\Delta}(y_3 - y_1) > 0 \tag{5.144}$$

$$-\frac{1}{2\Delta}(x_3 - x_2) > 0 \tag{5.145}$$

$$\frac{1}{2\Delta}(x_3 - x_1) > 0 \tag{5.146}$$

Now, if $\Delta > 0$, then

$$y_3 > y_2 \tag{5.147}$$

$$y_1 > y_3 \tag{5.148}$$

$$x_2 > x_3 \tag{5.149}$$

$$x_3 > x_1 \tag{5.150}$$

which imply that

$$x_2 > x_3 > x_1 \tag{5.151}$$

$$y_1 > y_3 > y_2 \tag{5.152}$$

However, if $\Delta < 0$, then subject to $k_i > 0$, (5.143)-(5.146) give

$$y_3 < y_2 \tag{5.153}$$

$$y_1 < y_3 \tag{5.154}$$

$$x_2 < x_3 \tag{5.155}$$

$$x_3 < x_1 \tag{5.156}$$

which imply that

$$x_2 < x_3 < x_1 \tag{5.157}$$

$$y_1 < y_3 < y_2 \tag{5.158}$$

Similarly, if $k_1 < 0$, $k_2 < 0$, $k_3 < 0$ and $k_4 < 0$, then

$$\frac{1}{2\Delta}(y_3 - y_2) < 0 \tag{5.159}$$

$$-\frac{1}{2\Delta}(y_3 - y_1) < 0 \tag{5.160}$$

$$-\frac{1}{2\Delta}(x_3 - x_2) < 0 \tag{5.161}$$

$$\frac{1}{2\Delta}(x_3 - x_1) < 0 \tag{5.162}$$

If $\Delta > 0$, then we obtain the same result as in (5.153)-(5.158). On the other hand, if $\Delta < 0$, then we obtain the same result as in (5.147)-(5.152) subject to $k_i < 0$. We infer that all the localisation constants, k_1, k_2, k_3 and k_4 have the same sign if the beacon nodes are positioned such that their coordinates satisfy either (5.151) and (5.152) or (5.157) and (5.158). Furthermore, from Table 5.5 and Table 5.6, we observe that the error E_x or E_y at $(e_1, e_2, e_3) = (r_{a1}, r_{a2}, r_{a3})$ due to the positive distance estimation errors is -3 times the error resulting from the same magnitudes of negative distance estimation errors at $(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, -r_{a3})$. From Table 5.5, E_x due to negative distance estimation errors $(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, -r_{a3})$ is given by

$$E_{x^-} = k_1(r_{a3}^2 - r_{a1}^2) + k_2(r_{a3}^2 - r_{a2}^2) \tag{5.163}$$

E_x due to positive distance estimation errors $(e_1, e_2, e_3) = (r_{a1}, r_{a2}, r_{a3})$ is given by

$$E_{x^+} = -3[k_1(r_{a3}^2 - r_{a1}^2) + k_2(r_{a3}^2 - r_{a2}^2)] \tag{5.164}$$

Dividing (5.164) by (5.163)

$$E_{x^+} = -3E_{x^-} \tag{5.165}$$

Similarly, from Table 5.6

$$E_{y^-} = k_3(r_{a3}^2 - r_{a1}^2) + k_4(r_{a3}^2 - r_{a2}^2) \tag{5.166}$$

$$E_{y^+} = -3[k_3(r_{a3}^2 - r_{a1}^2) + k_4(r_{a3}^2 - r_{a2}^2)] \tag{5.167}$$

Dividing (5.167) by (5.166)

$$E_{y^+} = -3E_{y^-} \tag{5.168}$$

It can also be noted from Table 5.5 and Table 5.6 that $E_x = 0$ and $E_y = 0$ at both negative $(-r_{a1}, -r_{a2}, -r_{a3})$ and positive (r_{a1}, r_{a2}, r_{a3}) points of distance estimation errors if $r_{a1} = r_{a2} = r_{a3}$ i.e. if all the three beacon nodes are equidistant and have the same amount of range error.

5.1.4 Case IV: Localisation error (e_l)

As localisation error e_l is the distance between the actual and estimated position coordinates, it cannot have a negative value. Therefore, the most minimum possible value of e_l is

$$e_{lmin} = 0 \tag{5.169}$$

according to (4.63) when both $E_x = 0$ and $E_y = 0$. As concluded in Section 5.1.3, this is possible when $r_{a1} = r_{a2} = r_{a3}$ and $e_1 = e_2 = e_3$. The maximum value of e_l depends upon the localisation constants k_1, k_2, k_3 and k_4, and hence the geometry of positions of beacon nodes. As determined in Section 5.1.3, all localisation constants have the same sign when (5.151) and (5.152) or (5.157) and (5.158) are satisfied. Under these conditions, either the simultaneous minimum or the simultaneous maximum values of E_x and E_y result in the maximum value of e_l. When $k_1 > 0$, $k_2 > 0$, $k_3 > 0$ and $k_4 > 0$, comparing (5.135) with (5.136) and (5.137) with (5.138), we observe that if

$$3r_{a3}^2 + r_{a1}^2 > r_{a3}^2 + 3r_{a1}^2 \tag{5.170}$$

$$r_{a3} > r_{a1} \tag{5.171}$$

and

$$3r_{a3}^2 + r_{a2}^2 > r_{a3}^2 + 3r_{a2}^2 \tag{5.172}$$

$$r_{a3} > r_{a2} \tag{5.173}$$

then

$$e_{lmax} = \sqrt{E_{xmin}^2 + E_{ymin}^2} \tag{5.174}$$

and e_{lmax} is located at $(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, r_{a3})$. However, if

$$r_{a3}^2 + 3r_{a1}^2 > 3r_{a3}^2 + r_{a1}^2 \tag{5.175}$$

$$r_{a1} > r_{a3} \tag{5.176}$$

and

$$r_{a3}^2 + 3r_{a2}^2 > 3r_{a3}^2 + r_{a2}^2 \tag{5.177}$$

$$r_{a2} > r_{a3} \tag{5.178}$$

then

$$e_{lmax} = \sqrt{E_{xmax}^2 + E_{ymax}^2} \tag{5.179}$$

and e_{lmax} is located at $(e_1, e_2, e_3) = (r_{a1}, r_{a2}, -r_{a3})$. When $k_1 < 0$, $k_2 < 0$, $k_3 < 0$ and $k_4 < 0$, and (5.171) and (5.173) are satisfied, then e_{lmax} is located at $(-r_{a1}, -r_{a2}, r_{a3})$ and is given by (5.179), and if (5.176) and (5.178) are satisfied, then it is located at $(r_{a1}, r_{a2}, -r_{a3})$ and is given by (5.174). The conditions specified for e_{lmax} in (5.171) and (5.173) or (5.176) and (5.178) are sufficient but not necessary. When these conditions are not met, e_{lmax} is still given by either (5.174) or (5.179) provided all localisation constants have the same sign.

5.1.5 Further analysis

5.1.5.1 Equal error

In the preceding discussions in Section 5.1.2 and Section 5.1.3, we have analysed the localisation error for cases where the beacon nodes are equidistant from the unknown node and the distance estimation errors are also equal. Let us now consider the situation where the beacon nodes are not necessarily equidistant ($r_{a1} \neq r_{a2} \neq r_{a3}$). However, the distance estimation errors are the same ($e_1 = e_2 = e_3 = e_d$). The distance estimation error, say e_d, can be either positive ($e_d > 0$) in which case

$$r_i = r_{ai} + e_d \geq r_{ai} \; \forall i \in \{1, 2, 3\} \tag{5.180}$$

or negative ($e_d < 0$) in which case

$$r_i = r_{ai} + e_d \leq r_{ai} \ \forall i \in \{1, 2, 3\} \tag{5.181}$$

Let us first consider the case where the distance estimation error is positive so that

$$e_1 = e_2 = e_3 = +e_d \tag{5.182}$$

Substituting this in (4.60)

$$\begin{aligned} E_{xd^+} &= k_1[2r_{a1}e_d + e_d^2 - 2r_{a3}e_d - e_d^2] \\ &+ k_2[2r_{a2}e_d + e_d^2 - 2r_{a3}e_d - e_d^2] \end{aligned} \tag{5.183}$$

$$E_{xd^+} = 2k_1e_d(r_{a1} - r_{a3}) + 2k_2e_d(r_{a2} - r_{a3}) \tag{5.184}$$

Substituting (5.182) in (4.61)

$$\begin{aligned} E_{yd^+} &= k_3[2r_{a1}e_d + e_d^2 - 2r_{a3}e_d - e_d^2] \\ &+ k_4[2r_{a2}e_d + e_d^2 - 2r_{a3}e_d - e_d^2] \end{aligned} \tag{5.185}$$

$$E_{yd^+} = 2k_3e_d(r_{a1} - r_{a3}) + 2k_4e_d(r_{a2} - r_{a3}) \tag{5.186}$$

From the above results in (5.184) and (5.186), we can see that the squared error terms (e_d^2) have been eliminated.

Let us now consider the case where the distance estimation errors are equal and negative.

$$e_1 = e_2 = e_3 = -e_d \tag{5.187}$$

Substituting (5.187) in (4.60)

$$\begin{aligned} E_{xd^-} &= k_1[-2r_{a1}e_d + e_d^2 + 2r_{a3}e_d - e_d^2] \\ &+ k_2[-2r_{a2}e_d + e_d^2 + 2r_{a3}e_d - e_d^2] \end{aligned} \tag{5.188}$$

$$E_{xd^-} = -[2k_1e_d(r_{a1} - r_{a3}) + 2k_2e_d(r_{a2} - r_{a3})] \tag{5.189}$$

We obtain the same expression as in (5.184) but with opposite sign. The error term in (5.184) is additive while that in (5.189) is subtractive. Now substituting (5.187) in (4.61)

$$\begin{aligned} E_{yd^-} &= k_3[-2r_{a1}e_d + e_d^2 + 2r_{a3}e_d - e_d^2] \\ &+ k_4[-2r_{a2}e_d + e_d^2 + 2r_{a3}e_d - e_d^2] \end{aligned} \tag{5.190}$$

$$E_{yd^-} = -[2k_3e_d(r_{a1} - r_{a3}) + 2k_4e_d(r_{a2} - r_{a3})] \tag{5.191}$$

Comparing (5.184) with (5.189) and (5.186) with (5.191), we conclude that

$$E_{xd^+} = -E_{xd^-} \tag{5.192}$$

$$E_{yd^+} = -E_{yd^-} \tag{5.193}$$

The estimated coordinate values, and hence the estimated position is different in both the cases. However, the resultant localisation error e_l is the same.

$$e_l = \sqrt{E_{xd^+}^2 + E_{yd^+}^2} = \sqrt{E_{xd^-}^2 + E_{yd^-}^2} \tag{5.194}$$

5.1.5.2 Unequal error

Let us now assume that the distances of the beacon nodes from the unknown node are unequal i.e.

$$r_{a1} \neq r_{a2} \neq r_{a3} \tag{5.195}$$

Furthermore, the distance estimation errors are also unequal so that

$$e_1 \neq e_2 \neq e_3 \tag{5.196}$$

However, the distance estimation error is proportional to the actual distance of the beacon node from the unknown node. This proportion $\mathbf{p}$ is the same for all the beacon nodes. In other words

$$\begin{aligned} e_1 &= \mathbf{p}r_{a1} \\ e_2 &= \mathbf{p}r_{a2} \\ e_3 &= \mathbf{p}r_{a3} \end{aligned} \tag{5.197}$$

so that

$$r_i = r_{ai} + e_i = r_{ai} + \mathbf{p}r_{ai} = (\mathbf{p}+1)r_{ai} \tag{5.198}$$

$\mathbf{p}$ can be either positive or negative such that $-1 \leq \mathbf{p} \leq 1$ so that $-r_{ai} \leq e_i \leq r_{ai}$ and $0 \leq r_i \leq 2r_{ai}$. However, $\mathbf{p} \nless -1$ because this will result in $e_i < -r_{ai}$ and $r_i < 0$ which is not possible. Further, $\mathbf{p} \ngtr 1$ because equal and opposite negative error for comparison will result in $\mathbf{p} < -1$ which is not possible.

We now calculate and compare the localisation error for two scenarios. In the first instance, we consider $\mathbf{p}$ to be positive, and for the second case we consider

p to be negative but having the same magnitude as in the first instance. For the first case, let $\mathbf{p} = +p$ so that

$$\begin{aligned} e_1 &= pr_{a1} \\ e_2 &= pr_{a2} \\ e_3 &= pr_{a3} \end{aligned} \tag{5.199}$$

For the second case, we let $\mathbf{p} = -p$ so that

$$\begin{aligned} e_1 &= -pr_{a1} \\ e_2 &= -pr_{a2} \\ e_3 &= -pr_{a3} \end{aligned} \tag{5.200}$$

We calculate E_x and E_y for both the cases. Let us first substitute (5.199) in (4.60) to calculate localisation error E_x due to positive distance estimation error.

$$\begin{aligned} E_{xp^+} &= k_1[2pr_{a1}^2 + p^2r_{a1}^2 - 2pr_{a3}^2 - p^2r_{a3}^2] \\ &+ k_2[2pr_{a2}^2 + p^2r_{a2}^2 - 2pr_{a3}^2 - p^2r_{a3}^2] \end{aligned} \tag{5.201}$$

$$\begin{aligned} E_{xp^+} &= k_1[pr_{a1}^2(p+2) - pr_{a3}^2(p+2)] \\ &+ k_2[pr_{a2}^2(p+2) - pr_{a3}^2(p+2)] \end{aligned} \tag{5.202}$$

$$E_{xp^+} = p(p+2)[k_1(r_{a1}^2 - r_{a3}^2) + k_2(r_{a2}^2 - r_{a3}^2)] \tag{5.203}$$

Let us now substitute (5.200) in (4.60) to calculate localisation error E_x due to negative distance estimation error.

$$\begin{aligned} E_{xp^-} &= k_1[-2pr_{a1}^2 + p^2 r_{a1}^2 + 2pr_{a3}^2 - p^2 r_{a3}^2] \\ &\quad + k_2[-2pr_{a2}^2 + p^2 r_{a2}^2 + 2pr_{a3}^2 - p^2 r_{a3}^2] \end{aligned} \tag{5.204}$$

$$\begin{aligned} E_{xp^-} &= k_1[pr_{a1}^2(p-2) - pr_{a3}^2(p-2)] \\ &\quad + k_2[pr_{a2}^2(p-2) - pr_{a3}^2(p-2)] \end{aligned} \tag{5.205}$$

$$E_{xp^-} = p(p-2)[k_1(r_{a1}^2 - r_{a3}^2) + k_2(r_{a2}^2 - r_{a3}^2)] \tag{5.206}$$

Comparing (5.203) and (5.206) we observe that $E_{xp^+} \neq E_{xp^-}$. Therefore, the localisation error is not the same when unequal but proportionate distance estimation errors are additive compared to the case when the distance estimation errors are subtractive. Dividing (5.203) by (5.206), we obtain

$$\frac{E_{xp^+}}{E_{xp^-}} = \frac{p+2}{p-2} \quad 0 \leq p \leq 1 \tag{5.207}$$

As $p = |\mathbf{p}|$ and $p \geq 0$, we can see from (5.207) that $|E_{xp^+}| \geq |E_{xp^-}|$. If an error $+pr_{ai}$ is present in the estimated distances, then the localisation error is higher compared to the case when the distance estimation error is $-pr_{ai}$. In other words, the error E_x is higher if the distance estimation errors proportionate to the actual distances of the beacon nodes are additive to the estimated distances and the resultant estimated distances are longer than the actual distances. The localisation error E_x is smaller if the same amount of distance errors are subtractive and the resultant estimated distances are shorter than the actual distances.

Let us now derive similar results for E_y. First, let us substitute (5.199) in

(4.61)

$$\begin{aligned} E_{yp+} &= k_3[2pr_{a1}^2 + p^2r_{a1}^2 - 2pr_{a3}^2 - p^2r_{a3}^2] \\ &+ k_4[2pr_{a2}^2 + p^2r_{a2}^2 - 2pr_{a3}^2 - p^2r_{a3}^2] \end{aligned} \tag{5.208}$$

$$\begin{aligned} E_{yp+} &= k_3[pr_{a1}^2(p+2) - pr_{a3}^2(p+2)] \\ &+ k_4[pr_{a2}^2(p+2) - pr_{a3}^2(p+2)] \end{aligned} \tag{5.209}$$

$$E_{yp^+} = p(p+2)[k_3(r_{a1}^2 - r_{a3}^2) + k_4(r_{a2}^2 - r_{a3}^2)] \tag{5.210}$$

Now we substitute (5.200) in (4.61) to calculate localisation error E_y due to negative distance estimation error.

$$\begin{aligned} E_{yp^-} &= k_3[-2pr_{a1}^2 + p^2r_{a1}^2 + 2pr_{a3}^2 - p^2r_{a3}^2] \\ &+ k_4[-2pr_{a2}^2 + p^2r_{a2}^2 + 2pr_{a3}^2 - p^2r_{a3}^2] \end{aligned} \tag{5.211}$$

$$\begin{aligned} E_{yp^-} &= k_3[pr_{a1}^2(p-2) - pr_{a3}^2(p-2)] \\ &+ k_4[pr_{a2}^2(p-2) - pr_{a3}^2(p-2)] \end{aligned} \tag{5.212}$$

$$E_{yp^-} = p(p-2)[k_3(r_{a1}^2 - r_{a3}^2) + k_4(r_{a2}^2 - r_{a3}^2)] \tag{5.213}$$

Dividing (5.210) by (5.213), we get

$$\frac{E_{yp^+}}{E_{yp^-}} = \frac{p+2}{p-2} \quad 0 \le p \le 1 \tag{5.214}$$

which is the same result as obtained in (5.207) for the x component of the localisation error E_x.

The resultant localisation error, e_{lp^+} when the distance errors are additive in the estimated distances

$$e_{lp^+} = \sqrt{E_{xp^+}^2 + E_{yp^+}^2} \tag{5.215}$$

From (5.207) and (5.214)

$$E_{xp^+} = \frac{p+2}{p-2} E_{xp^-} \tag{5.216}$$

$$E_{yp^+} = \frac{p+2}{p-2} E_{yp^-} \tag{5.217}$$

Substituting (5.216) and (5.217) in (5.215)

$$e_{lp^+} = \left|\frac{p+2}{p-2}\right| \sqrt{E_{xp^-}^2 + E_{yp^-}^2} \tag{5.218}$$

However, the localisation error, e_{lp^-} when the distance errors are subtractive in the estimated distances is given by

$$e_{lp^-} = \sqrt{E_{xp^-}^2 + E_{yp^-}^2} \tag{5.219}$$

Substituting (5.219) in (5.218)

$$e_{lp^+} = \left|\frac{p+2}{p-2}\right| e_{lp^-} \quad 0 \leq p \leq 1 \tag{5.220}$$

$$e_{lp^+} = \xi e_{lp^-} \tag{5.221}$$

where ξ is error coefficient and is given by

$$\xi = \left|\frac{p+2}{p-2}\right| \quad 0 \leq p \leq 1 \tag{5.222}$$

As $p \geq 0$, $|e_{lp^+}| \geq |e_{lp^-}|$. In other words, the localisation error is higher if the

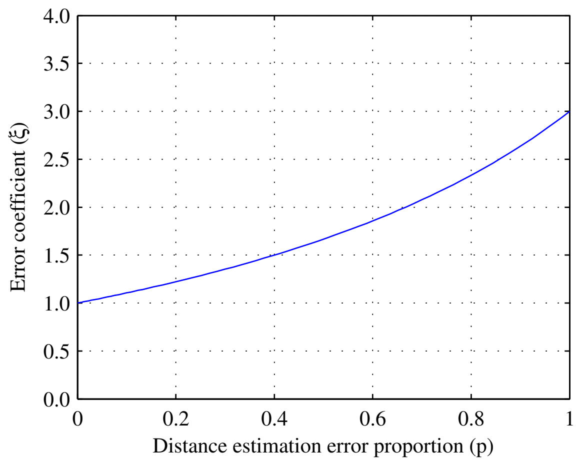


Figure 5.1: Relative change in localisation error with distance estimation error.

distance estimation error is positive and the estimated distances are longer than the actual distances compared to the case where distance estimation errors are negative so that the estimated distances are shorter than the actual distances. As an example, consider $p = 1$ i.e. we compare localisation error e_{lp^+} when $e_i = r_{ai}$ and $r_i = 2r_{ai}$ to the localisation error e_{lp^-} when $e_i = -r_{ai}$ and $r_i = 0$. Substituting $p = 1$ in (5.220), we get

$$e_{lp^+} = 3e_{lp^-} \tag{5.223}$$

i.e. if the errors equal to the actual distances are added to the estimated distances then the localisation error is three times as high as when the same errors are subtracted from the estimated distances. In Fig. 5.1 we plot the error coefficient ξ. It can be observed that the error coefficient increases as the error proportion in the estimated distances increases. This again confirms our result derived earlier. We also observe that the higher the distance estimation error, the higher is the relative increase in e_{lp^+} compared to e_{lp^-}.

It is also interesting to compare E_{xd^+} and E_{yd^+} with E_{xp^+} and E_{yp^+} and similarly E_{xd^-} and E_{yd^-} with E_{xp^-} and E_{yp^-}. From (5.184), (5.186), (5.189), (5.191), (5.203), (5.206), (5.210) and (5.213), we observe that E_{xd^+}, E_{yd^+}, E_{xd^-} and E_{yd^-} involve linear terms whereas E_{xp^+}, E_{yp^+}, E_{xp^-} and E_{yp^-} involve squared error terms. This leads to the conclusion that the localisation error is lower if the distance estimation error is equal for all neighbour beacon nodes than if it is unequal.

5.1.5.3 Geometry of positioning of nodes

If x component of localisation error is eliminated i.e. $E_x = 0$, then from (4.44)

$$\begin{aligned} k_1E_1 + k_2E_2 &= 0 \\ \frac{E_1}{E_2} &= -\frac{k_2}{k_1} \end{aligned} \tag{5.224}$$

Substituting from (4.45), (4.46), (4.49) and (4.50)

$$\frac{2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2}{2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2} = \frac{y_3 - y_1}{y_3 - y_2} \tag{5.225}$$

$$\begin{aligned} &(y_3 - y_2)(2r_{a1}e_1 + e_1^2) + y_2(2r_{a3}e_3 + e_3^2) \\ =&(y_3 - y_1)(2r_{a2}e_2 + e_2^2) + y_1(2r_{a3}e_3 + e_3^2) \end{aligned} \tag{5.226}$$

$$\begin{aligned} &(y_3 - y_2)(2r_{a1}e_1 + e_1^2) + (y_2 - y_1)(2r_{a3}e_3 + e_3^2) \\ =&(y_3 - y_1)(2r_{a2}e_2 + e_2^2) \end{aligned} \tag{5.227}$$

Similarly, if $E_y = 0$, then from (4.57)

$$\begin{aligned} k_3E_1 + k_4E_2 &= 0 \\ \frac{E_1}{E_2} &= -\frac{k_4}{k_3} \end{aligned} \tag{5.228}$$

Substituting from (4.49), (4.50), (4.58) and (4.59)

$$\frac{2r_{a1}e_1 + e_1^2 - 2r_{a3}e_3 - e_3^2}{2r_{a2}e_2 + e_2^2 - 2r_{a3}e_3 - e_3^2} = \frac{x_3 - x_1}{x_3 - x_2} \tag{5.229}$$

$$\begin{aligned} &(x_3 - x_2)(2r_{a1}e_1 + e_1^2) + x_2(2r_{a3}e_3 + e_3^2) \\ =&(x_3 - x_1)(2r_{a2}e_2 + e_2^2) + x_1(2r_{a3}e_3 + e_3^2) \end{aligned} \tag{5.230}$$

$$\begin{aligned} &(x_3 - x_2)(2r_{a1}e_1 + e_1^2) + (x_2 - x_1)(2r_{a3}e_3 + e_3^2) \\ =&(x_3 - x_1)(2r_{a2}e_2 + e_2^2) \end{aligned} \tag{5.231}$$

If relative positions of neighbour beacon nodes and errors in distance estimates are such that (5.225), (5.226) or (5.227) is satisfied, then $E_x = 0$. Similarly, if (5.229), (5.230) or (5.231) is satisfied, then $E_y = 0$. Hence, it is possible that for a certain relative geometry of positions of nodes, there is no error in the estimated position even though there are errors in the estimated distances.

In the above analysis, the error term ζ_i characterises error due to a single beacon node. E_1 and E_2 characterise error due to a pair of beacon nodes. E_x, E_y and e_l characterise error due to the three beacon nodes.

5.1.6 Summary of results

Below is a summary of results that we have derived from our analysis:

(1) Localisation error e_l comprises of x and y components E_x and E_y given by:

$E_x = k_1E_1 + k_2E_2$

$E_y = k_3E_1 + k_4E_2$

(2) The distance estimation error contributes equally to the x and y components of the localisation error. The difference in E_x and E_y components of localisation error depends only upon the geometry of placement of the beacon nodes.

(3) Error terms of the form $\zeta_i = 2r_{ai}e_i + e_i^2$ are responsible for localisation error in the estimated position.

(4) A localisation error term $2r_{ai}e_i + e_i^2$ can be eliminated only if the corresponding distance estimation error e_i is eliminated.

(5) The individual error term $\zeta_i = 2r_{ai}e_i + e_i^2$ has a minimum at $e_i = -r_{ai}$ and the minimum possible value of the error term is $\zeta_{imin} = -r_{ai}^2$. The error term cannot have a signed value lesser than this minimum.

(6) The individual error term $\zeta_i = 2r_{ai}e_i + e_i^2$ has a maximum at $e_i = r_{ai}$ for the domain $-r_{ai} \leq e_i \leq r_{ai}$ and the maximum value of the error term is $3r_{ai}^2$. The maximum value is 3 times the minimum value.

(7) If an unknown node has the same amount of error in the distance estimates from two equidistant neighbour beacon nodes, then either $E_1 = 0$ or $E_2 = 0$ or $E_1 = E_2$, and hence the localisation error is solely determined by either E_1 or E_2.

(8) Each of the x and y components of the localisation error is a function of two types of error components. E_1 has a critical point at $(-r_{a1}, -r_{a3})$ and E_2 has a critical point at $(-r_{a2}, -r_{a3})$. E_1 and E_2 at the critical points are $E_{1sad} = r_{a3}^2 - r_{a1}^2$ and $E_{2sad} = r_{a3}^2 - r_{a2}^2$.

(9) The minimum value of E_1 is $-3r_{a3}^2 - r_{a1}^2$ which occurs at $(e_1, e_3) = (-r_{a1}, r_{a3})$. The maximum value of E_1 is $r_{a3}^2 + 3r_{a1}^2$ over

the domain $-r_{ai} \leq e_i \leq r_{ai}$ which occurs at $(e_1, e_3) = (r_{a1}, -r_{a3})$.

(10) The minimum value of E_2 is $-3r_{a3}^2 - r_{a2}^2$ which occurs at $(e_2, e_3) = (-r_{a2}, r_{a3})$. The maximum value of E_2 is $r_{a3}^2 + 3r_{a2}^2$ over the domain $-r_{ai} \leq e_i \leq r_{ai}$ which occurs at $(e_2, e_3) = (r_{a2}, -r_{a3})$.

(11) The extreme values of E_1 and E_2 occur when the distance estimation errors of the involved pair of beacon nodes have opposite signs. E_1 and E_2 lie between the extreme values when the distance estimation errors have the same sign.

(12) The error component E_1 or E_2 due to the positive distance estimation errors is three times the error component resulting from the same magnitudes of negative distance estimation errors.

(13) If the estimated distances have the same amount of error ($e_1 = e_2 = e_3$), then the net localisation error in the estimated position will be zero if the unknown node is at the same distance ($r_{a1} = r_{a2} = r_{a3}$) from all the neighbour beacon nodes.

(14) If an unknown node can determine that it is equidistant from three neighbour beacon nodes, then it can determine its exact position by merely using the position information of beacon nodes and without estimating distances from them.

(15) A correct position estimation with zero localisation error does not necessarily imply that the estimated distances used in the position estimation are accurate.

(16) Accurate position can be obtained even if the estimated distances are not accurate.

(17) It is possible for an unknown node to determine its exact position without knowing its distances from the neighbour beacon nodes.

(18) If two of the neighbour beacon nodes are collinear parallel to an axis and are at the same distance from the unknown node and distance estimates for them have the same amount of error, then the component of localisation error for that axis is zero irrespective of the position and distance estimation error of the third beacon node.

(19) Each of the x and y components of localisation error, E_x and E_y, has a critical point at $(-r_{a1}, -r_{a2}, -r_{a3})$. At the critical point:
$E_{xsad} = k_1[r_{a3}^2 - r_{a1}^2] + k_2[r_{a3}^2 - r_{a2}^2]$ $E_{ysad} = k_3[r_{a3}^2 - r_{a1}^2] + k_4[r_{a3}^2 - r_{a2}^2]$

(20) The minima and maxima of E_x and E_y depend upon the localisation constants k_1, k_2, k_3 and k_4, and hence on the geometry formed by the positions of beacon nodes.

(21) If the localisation constants are positive, the minima of E_x and E_y are located at $(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, r_{a3})$ and are given by
$E_{xmin} = -k_1[3r_{a3}^2 + r_{a1}^2] - k_2[3r_{a3}^2 + r_{a2}^2]$
$E_{ymin} = -k_3[3r_{a3}^2 + r_{a1}^2] - k_4[3r_{a3}^2 + r_{a2}^2]$

(22) If the localisation constants are positive, the maxima of E_x and E_y are located at $(e_1, e_2, e_3) = (r_{a1}, r_{a2}, -r_{a3})$ and are given by
$E_{xmax} = k_1[r_{a3}^2 + 3r_{a1}^2] + k_2[r_{a3}^2 + 3r_{a2}^2]$
$E_{ymax} = k_3[r_{a3}^2 + 3r_{a1}^2] + k_4[r_{a3}^2 + 3r_{a2}^2]$

(23) The point $(-r_{a1}, -r_{a2}, r_{a3})$ at which the minima occur is exactly opposite to the point $(r_{a1}, r_{a2}, -r_{a3})$ at which the maxima occur.

(24) The error E_x or E_y at $(e_1, e_2, e_3) = (r_{a1}, r_{a2}, r_{a3})$ due to the positive distance estimation errors is -3 times the error resulting from the same magnitudes of negative distance estimation errors at $(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, -r_{a3})$.

(25) If the distance estimation error is constant and equal for all the neighbour beacon nodes, then the localisation error is the same whether the distance estimation error is positive or negative. However, the estimated coordinate values, and hence the estimated position will be different in both the cases.

(26) When the distance estimation error is different for different beacon nodes but proportionate to the actual distances, then the localisation error is not the same when the distance estimation error is additive compared to the case when the distance estimation error is subtractive.

(27) The localisation error is higher if the distance estimation errors proportionate to the actual distances of the beacon nodes are additive to the estimated distances and the resultant estimated distances are longer than the actual distances. The localisation error is smaller if the same amount of distance errors are subtractive and the resultant estimated distances are shorter than the actual distances.

(28) If the distance estimation error is a constant proportion of actual distances for all neighbour beacon nodes, then the localisation error is $\left|\frac{p+2}{p-2}\right|$ times higher when the distance estimation error is positive compared to when it is negative.

(29) If the errors equal to the actual distances are added to the estimated distances then the localisation error is three times as high as when the same errors are subtracted from the estimated distances.

(30) The distance estimation error in the region $-r_{ai} \leq e_i \leq 0$ gives lower position error. Hence, it is the preferred region of operation for location estimation.

(31) The signed value of the ζ_i component of the localisation error has a lower bound at $e_i = -r_{ai}$ where $r_i = 0$ and $\zeta_i = -r_{ai}^2$. If we consider signed value, this is also the minimum for ζ_i. There is no upper bound or maximum for ζ_i. However, if we consider the upper bound $e_i \leq r_{ai}$, then the maximum of ζ_i is $3r_{ai}^2$.

(32) It is possible that for a certain relative geometry of positions of nodes, there is no error in the estimated position even though there are errors in the estimated distances.

(33) The most minimum possible value of e_l is $e_{lmin} = 0$.

(34) When all the localisation constants have the same sign, the simultaneous minimum or simultaneous maximum values of E_x and E_y result in the maximum value of e_l.

(35) When $k_1 > 0$, $k_2 > 0$, $k_3 > 0$ and $k_4 > 0$, then e_{lmax} due to simultaneous minimum values of E_x and E_y is located at $(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, r_{a3})$ given by $e_{lmax} = \sqrt{E_{xmin}^2 + E_{ymin}^2}$. If e_{lmax} is due to simultaneous maximum values of E_x and E_y, it is located at $(e_1, e_2, e_3) = (r_{a1}, r_{a2}, -r_{a3})$ given by $e_{lmax} = \sqrt{E_{xmax}^2 + E_{ymax}^2}$.

(36) If $k_1 < 0$, $k_2 < 0$, $k_3 < 0$ and $k_4 < 0$, then e_{lmax} due to simultaneous minimum values of E_x and E_y is located at $(e_1, e_2, e_3) = (r_{a1}, r_{a2}, -r_{a3})$ and is given by $e_{lmax} = \sqrt{E_{xmin}^2 + E_{ymin}^2}$. If e_{lmax} is due to simultaneous

maximum values of E_x and E_y, then in this case it is located at $(e_1, e_2, e_3) = (-r_{a1}, -r_{a2}, r_{a3})$ and is given by $e_{lmax} = \sqrt{E_{xmax}^2 + E_{ymax}^2}$.

5.2 Numerical Results

We conduct a number of simulation experiments to verify the analytical results derived in Sections 4.3, 4.4 and 5.1. In the simulation experiments an unknown node estimates its position using three neighbour beacon nodes analogous to the results derived analytically. We use a two dimensional square sensor field of size 50×50 for our simulation experiments. We assume that the unknown node can estimate its distance r_i from a beacon node B_i with distance estimation error e_i. The actual position (x_a, y_a) of the unknown node is $(30, 20)$ using a Cartesian coordinate system for the square sensor field.

Table 5.7: Beacon nodes and their distances.

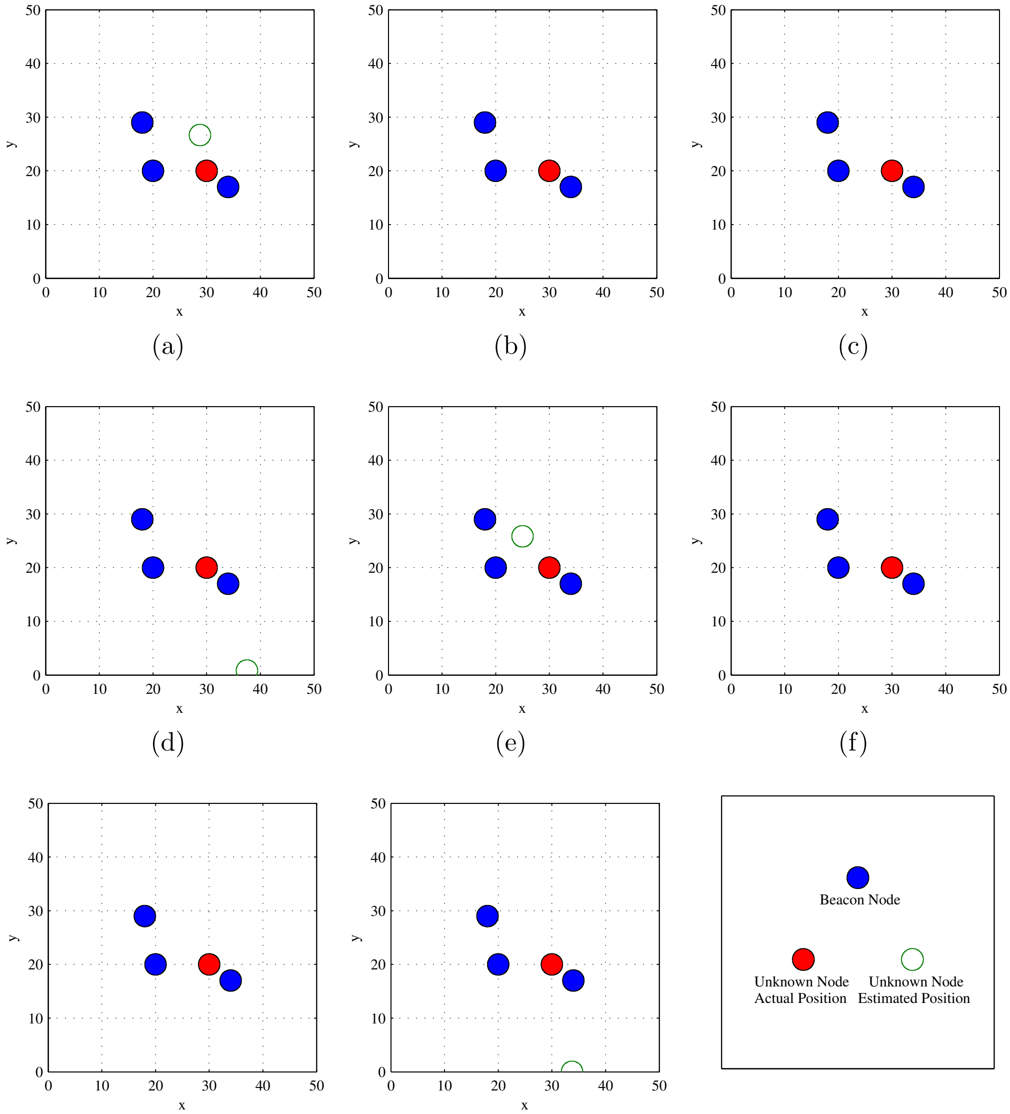
Beacon Node	Position	Actual Distance (r_{ai})
B_1	$(34, 17)$	$r_{a1} = 5$
B_2	$(18, 29)$	$r_{a2} = 15$
B_3	$(20, 20)$	$r_{a3} = 10$

Table 5.8: Extreme values of errors.

Experiment	Distance Estimation Error			Estimated Distances			Experimental Results				Analytical Results				
	e_1	e_2	e_3	r_1	r_2	r_3	(x, y)	E_x	E_y	e_l	E_1	E_2	E_x	E_y	e_l
1	-5	-15	-10	0	0	0	$(28.75, 26.67)$	-1.25	6.67	6.78	75	-125	-1.25	6.67	6.78
2	-5	-15	10	0	0	20	$(48.75, 53.33)$	18.75	33.33	38.24	-325	-525	18.75	33.33	38.24
3	-5	15	-10	0	30	0	$(17.50, -25.83)$	-12.50	-45.83	47.51	75	775	-12.50	-45.83	47.51
4	-5	15	10	0	30	20	$(37.50, 0.83)$	7.50	-19.17	20.58	-325	375	7.50	-19.17	20.58
5	5	-15	-10	10	0	0	$(25.00, 25.83)$	-5.00	5.83	7.68	175	-125	-5.00	5.83	7.68
6	5	-15	10	10	0	20	$(45.00, 52.50)$	15.00	32.50	35.79	-225	-525	15.00	32.50	35.79
7	5	15	-10	10	30	0	$(13.75, -26.67)$	-16.25	-46.67	49.41	175	775	-16.25	-46.67	49.41
8	5	15	10	10	30	20	$(33.75, 0.00)$	3.75	-20.00	20.35	-225	375	3.75	-20.00	20.35

5.2.1 Extreme values of error

To verify the minimum and maximum values of different components of localisation error i.e. ζ_i, E_1, E_2, E_x, E_y and e_l, we perform a simulation experiment using an unknown node U_1 positioned at $(30, 20)$ with three beacon nodes B_1, B_2 and B_3 deployed in a square sensor field of size 50×50. The position coordinates of the beacon nodes and their actual distances from the unknown node U_1 are given in Table 5.7. For simplicity of comparison, the positions of beacon nodes satisfy $x_2 < x_3 < x_1$ and $y_1 < y_3 < y_2$. As a result, all the localisation constants have the same sign and $k_1 < 0$, $k_2 < 0$, $k_3 < 0$ and $k_4 < 0$. The experiment is repeated for all the possible combinations of extreme values of distance estimation errors. While the positions of beacon nodes and their distances from the unknown node remain fixed as shown in Fig. 5.2, the distance estimation errors change with each repetition of the experiment. As positions of beacon nodes do not change, the geometry does not change. Hence, the localisation constants remain unchanged for all repetitions of the experiment. Using (4.45), (4.46), (4.58) and (4.59), these are calculated as $k_1 = -0.0375$, $k_2 = -0.0125$, $k_3 = -0.0083$ and $k_4 = -0.0583$. We measure the estimated position using multilateration. We also measure the experimental values of E_x, E_y and e_l for each version of the experiment using (4.40), (4.53) and (4.62). At the same time, we also calculate these values using our analytical model from (4.60), (4.61) and (4.63) for the purpose of comparison. Both the sets of results are given in Table 5.8. We observe that for a given set of distance estimation errors, the localisation error calculated using the analytical results is the same as measured using the simulation experiment. Therefore, we conclude that our given analytical model and its various components are validated. We



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 5.2: Multilateration analysis for extreme values of distance estimation errors. (a)-(h) Node positions for 8 repetitions of the experiment. (f) Legend.

further analyse the results and comment on different errors in the following paragraphs.

5.2.1.1 Error term ζ_i

As analysed in Section 5.1.1, three types of error terms are responsible for localisation error in the estimated position. These error terms are $\zeta_i = 2r_{ai}e_i + e_i^2 \ \ \forall i \in \{1, 2, 3\}$. In Fig. 5.3, we plot the error term $\zeta_1 = 2r_{a1}e_1 + e_1^2$ as a function of distance estimation error e_1 over the interval $-r_{a1} \leq e_1 \leq r_{a1}$ where $r_{a1} = 5$ as used in our simulation experiment. As we can see, the plot is a parabola. The plot for any error term ζ_i is similar. We can rewrite the equation for an error term ζ_i as below

$$\zeta_i = (e_i - (-r_{ai}))^2 - r_{ai}^2 \tag{5.232}$$

Let us compare it with the vertex form of the equation of a parabola

$$y = a(x - h)^2 + k \tag{5.233}$$

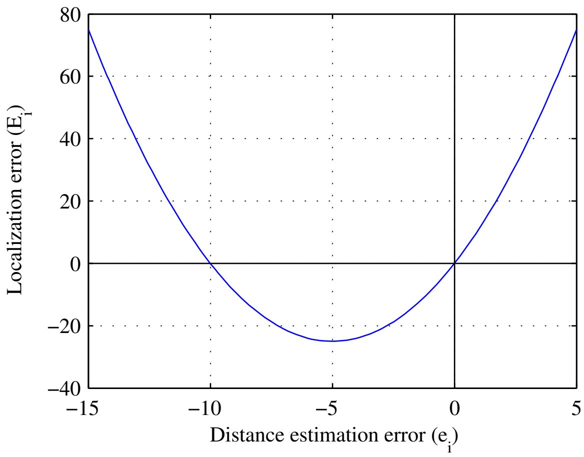


Figure 5.3: Localisation error term ζ_i as a function of distance estimation error.

Hence, the vertex (h, k) of the parabola lies at $(-r_{ai}, -r_{ai}^2)$. Now general equation of a parabola with focus (a, b) and directrix $y = k$ is given by

$$y = \frac{(x-a)^2}{2(b-k)} + \frac{b+k}{2} \tag{5.234}$$

Comparing it with (5.232), we get

$$a = -r_{ai} \tag{5.235}$$

$$\frac{1}{2(b-k)} = 1 \tag{5.236}$$

$$\frac{b+k}{2} = -r_{ai}^2 \tag{5.237}$$

Solving (5.236) and (5.237), we get

$$b = \frac{1}{4} - r_{ai}^2 \tag{5.238}$$

$$k = -\frac{1}{4} - r_{ai}^2 \tag{5.239}$$

Hence, the parabola has axis of symmetry at $e_i = -r_{ai}$, focus at $(-r_{ai}, \frac{1}{4} - r_{ai}^2)$ and directrix at $\zeta_i = -\frac{1}{4} - r_{ai}^2$. Further, the plot can be divided into three regions:

1. $e_1 < -5$ i.e. $e_1 < -r_{a1}$
2. $-5 \leq e_1 < 0$ i.e. $-r_{a1} \leq e_1 < 0$
3. $e_1 \geq 0$

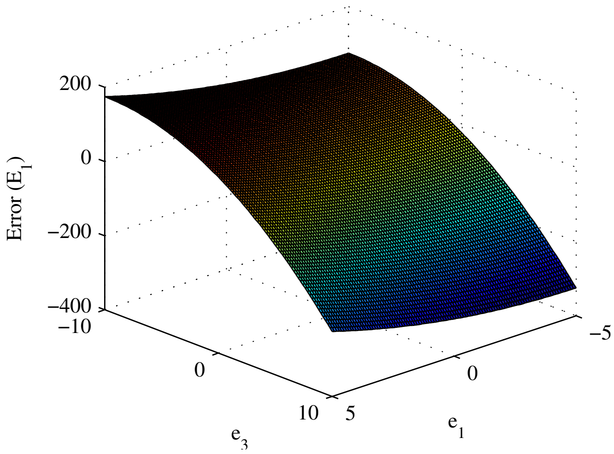
In the first region, $e_i < -r_{ai}$. This implies that $r_i = r_{ai} + e_i < 0$ i.e. the estimated distance is negative. As this is not possible, the part of the graph in this region is not significant. In the second region ($-r_{ai} \leq e_i < 0$), the resulting error term ζ_i is negative i.e. the error contributed to the estimated position coordinate is negative. Rate of change of error in this region is smaller compared to the other two regions. In other words, error e_i in distance estimation does not cause as much error in position estimation in this region as in the other two regions. The localisation error in this region is bounded on both sides. It has a lower bound at $e_i = -r_{ai}$ where $\zeta_i = 2r_{ai}e_i + e_i^2 = -r_{ai}^2$. The localisation error in this region has upper bound at $e_i = 0$ where $\zeta_i = 0$. In the third region ($0 \leq e_i \leq r_{ai}$), the distance estimation error as well as the resulting error term ζ_i is positive. The localisation error in this region is bounded only on one side. It has lower bound at $e_i = 0$ where $\zeta_i = 0$. It has no upper bound. As the first region is not significant, we consider and compare only later two regions. We observe that for an equal change in distance estimation error, the resultant change in the localisation error is higher in the third region compared to the second region. As the distance estimation error varies from $e_1 = 0$ to $e_1 = -r_{a1} = -5$ in the second region, the error ζ_1 changes from 0 to its minimum value $\zeta_{1min} = -25$. However, the same amount of change in distance estimation error from $e_1 = 0$ to $e_1 = +r_{a1} = 5$ in the third region results in change in the error ζ_1 from 0 to its maximum value $\zeta_{1max} = 75$ for the interval $-5 \leq e_1 \leq 5$. Hence, this change in the localisation error is three times as high as when the distance estimation error varies from 0 to $-r_{a1}$. This is in conformance with our earlier result derived in (5.12). The localisation error is higher when the distance estimation error is positive compared to when the distance estimation error is negative. In other words, localisation accuracy is higher if the estimated distances are shorter than the actual distances than if the estimated distances are longer. As distance estimation error in the region

$-r_{ai} \leq e_i \leq 0$ gives lower position error, it is the preferred region of operation for location estimation.

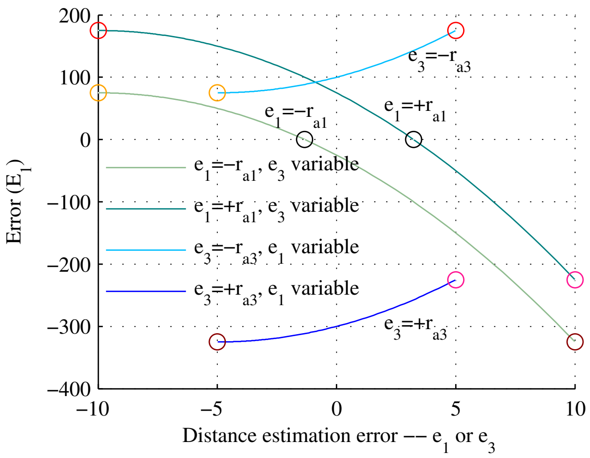
5.2.1.2 E_1 and E_2

From the results in Table 5.8, we observe that $E_{1min} = -325$ occurs when $(e_1, e_3) = (-5, 10)$ and $E_{1max} = 175$ is located at $(e_1, e_3) = (5, -10)$. Similarly, $E_{2min} = -525$ occurs whenever $(e_2, e_3) = (-15, 10)$ and $E_{2max} = 775$ is located at $(e_2, e_3) = (15, -10)$ in accordance with (5.50)-(5.53). We further observe that $E_{1^-} = 75$ at $(-5, -10)$ and $E_{1^+} = -225$ at $(5, 10)$ satisfy (5.56). Similarly, $E_{2^-} = -125$ at $(-15, -10)$ and $E_{2^+} = 375$ at $(15, 10)$ in accordance with (5.59). These values of E_1 and E_2, when the range estimation errors of the two beacon nodes have the same sign, lie between the extreme values which occur when the distance estimation errors of the two involved beacon nodes have opposite signs. Note that E_1 is a function of e_1 and e_3, and E_2 is a function of e_2 and e_3. There are only four possible combinations of extreme values of $(e_1, e_3) = (\pm r_{a1}, \pm r_{a3})$ as listed in Table 5.1 and $(e_2, e_3) = (\pm r_{a2}, \pm r_{a3})$ as listed in Table 5.2. Each of these combinations appears twice in Table 5.8 as part of $(e_1, e_2, e_3) = (\pm r_{a1}, \pm r_{a2}, \pm r_{a3})$. Hence, each value of E_1 and E_2, including the minimum and the maximum values, appears twice in Table 5.8.

We plot error E_1 given by (4.49) in Fig. 5.4 and error E_2 given by (4.50) in Fig. 5.5. To give an idea of their relative dimensions, we draw a combined plot of E_1 and E_2 in Fig. 5.6. For the two dimensional plot in Fig. 5.4b, we keep either $e_1 = \pm r_{a1}$ or $e_3 = \pm r_{a3}$ and plot E_1 using (4.49) as a function of the remaining variable. The graph in Fig. 5.5b is plotted in a similar manner. The plots in both Fig. 5.4 and Fig. 5.5 also verify the results



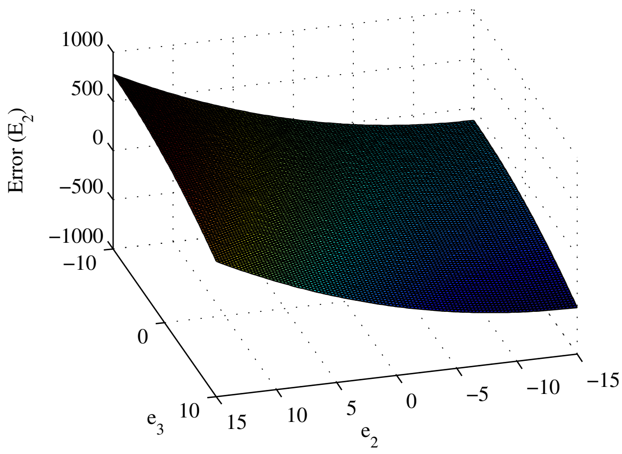
(a)



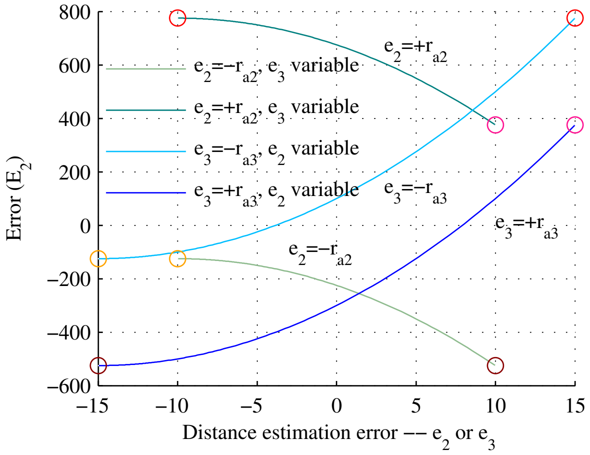
(b)

Figure 5.4: Error (E_1) as a function of distance estimation error for two non-equidistant beacon nodes.

derived in Section 5.1.2. The minimum value of E_1 is -325 and is located at $(e_1, e_3) = (-5, 10)$ conforming to (5.50) and the results in Table 5.8. The maximum value of E_1 is 175 and is located at $(e_1, e_3) = (5, -10)$ conforming to (5.51) and the results in Table 5.8. E_{1^-} at $(e_1, e_3) = (-5, -10)$ is 75 and E_{1^+} at $(e_1, e_3) = (5, 10)$ is -225 conforming to (5.56). The plots of E_2 in Fig. 5.5 also conform to the data in Table 5.8. Furthermore, from the plots in Fig. 5.4 and Fig. 5.5 for non-equidistant beacon nodes, it can further be observed that $E_1 = 0$ at a set of values of (e_1, e_3) and $E_2 = 0$ at a set of values of (e_2, e_3). For



(a)



(b)

Figure 5.5: Error (E_2) as a function of distance estimation error for two non-equidistant beacon nodes.

example, substituting $e_1 = \pm 5$, $r_{a1} = 5$, $r_{a3} = 10$ in (4.49), we calculate that $E_1 = 0$ at $(e_1, e_3) = (5, 3.23)$ and $(e_1, e_3) = (-5, -1.34)$. Further, substituting $r_{a2} = 15$, $r_{a3} = 10$, $e_3 = 3.23, -1.34$ in (4.50), we determine that $E_2 = 0$ at $(e_2, e_3) = (2.32, 3.23)$ and $(e_2, e_3) = (-0.86, -1.34)$. Hence, even if the beacon nodes B_1, B_2 and B_3 have range estimation errors $(e_1, e_2, e_3) = (5, 2.32, 3.23)$ or $(e_1, e_2, e_3) = (-5, -0.86, -1.34)$, the estimated position is still accurate without any localisation error i.e. $e_l = 0$. This is in accordance with results in Section 5.1.5.3. If the beacon nodes were equidistant so that $r_{a1} = r_{a3} = 5$, the

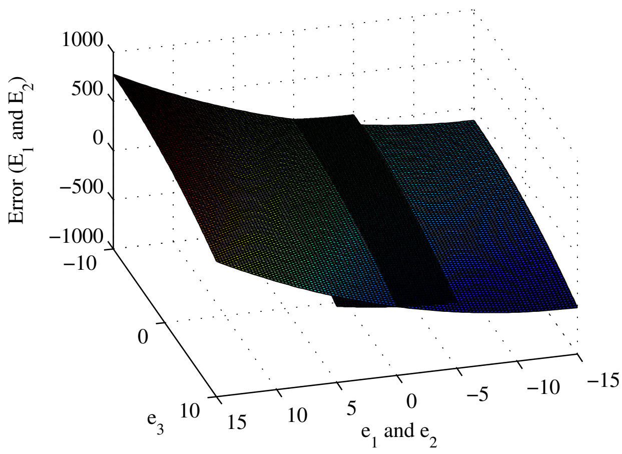


Figure 5.6: Error E_1 and E_2.

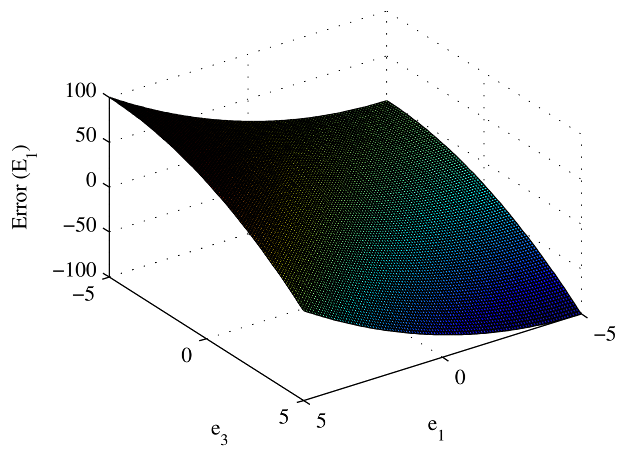
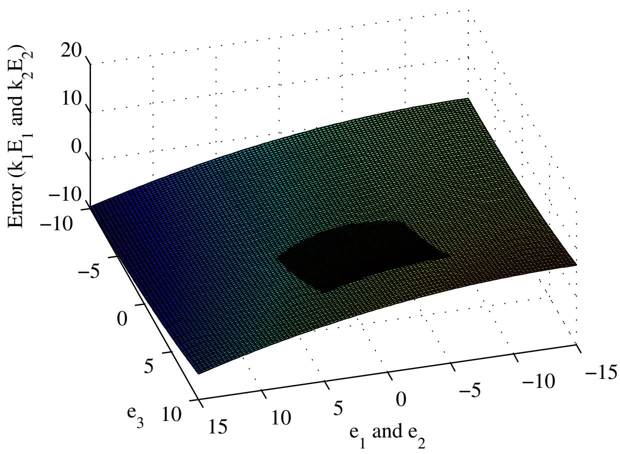


Figure 5.7: Error (E_1) as a function of distance estimation error for two equidistant beacon nodes.

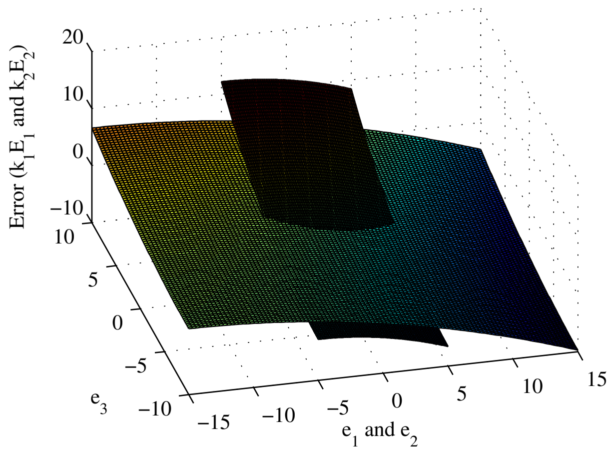
plot for E_1 would look as in Fig. 5.7. It can be observed that $E_{1^+} = E_{1^-} = 0$ for this case as noted in Section 5.1.2.

5.2.1.3 E_x and E_y

As noted in Section 5.2.1.2, the minimum and maximum values of E_1 and E_2 are repeated twice in Table 5.8. However, the minimum and maximum of E_1 and E_2 occur simultaneously only once. The simultaneous minima of



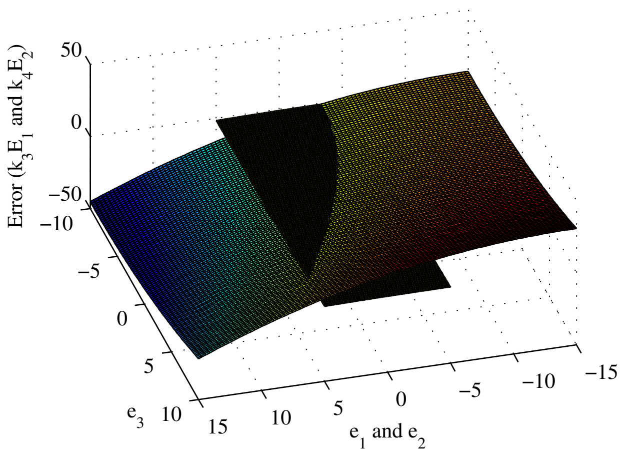
(a)



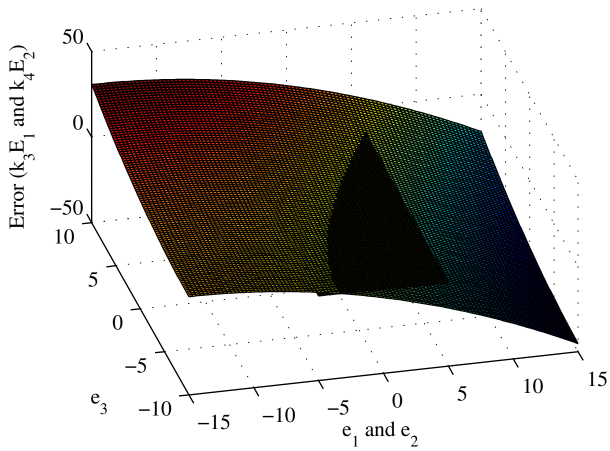
(b)

Figure 5.8: Components k_1E_1 and k_2E_2 of E_x from two angles (a) front azimuth and (b) back azimuth.

E_1 and E_2 occur at $(-5, -15, 10)$. As $k_1 < 0$, $k_2 < 0$, $k_3 < 0$ and $k_4 < 0$, E_x and E_y have maximum values $E_{xmax} = 18.75$ and $E_{ymax} = 33.33$ at this point as given by (5.140) and (5.142) respectively. The simultaneous maxima of E_1 and E_2 occur at $(5, 15, -10)$, and E_x and E_y have minimum values $E_{xmin} = -16.25$ and $E_{ymin} = -46.67$ at this point as given by (5.139) and (5.141). Similarly, we observe from the results in Table 5.8 that $E_{x^-} = -1.25$, $E_{x^+} = 3.75$, $E_{y^-} = 6.67$ and $E_{y^+} = -20.00$ in accordance with (5.165) and (5.168) in Section 5.1.3 and (5.207) and (5.214) with $p = 1$ in Section 5.1.5.2.



(a)



(b)

Figure 5.9: Components k_3E_1 and k_4E_2 of E_y from two angles (a) front azimuth and (b) back azimuth.

We plot the two components k_1E_1 and k_2E_2 of E_x in Fig. 5.8 using two different angles. Similarly, the components k_3E_1 and k_4E_2 of E_y are plotted in Fig. 5.9. We plot E_x using (4.60) in Fig. 5.10 as a function of e_1 and e_2 for three different constant values of e_3 i.e. $e_3 = -r_{a3} = -10$, $e_3 = 0$ and $e_3 = r_{a3} = 10$. Similarly, E_y is plotted using (4.61) in Fig. 5.11. To give a relative comparison, we give combined plots of E_x and E_y as a function of e_1 and e_2 for $e_3 = -r_{a3}, 0, +r_{a3}$ in Fig. 5.12. The plots in Fig. 5.10, Fig. 5.11 and Fig. 5.12 again verify the points of occurrence and magnitudes of extreme

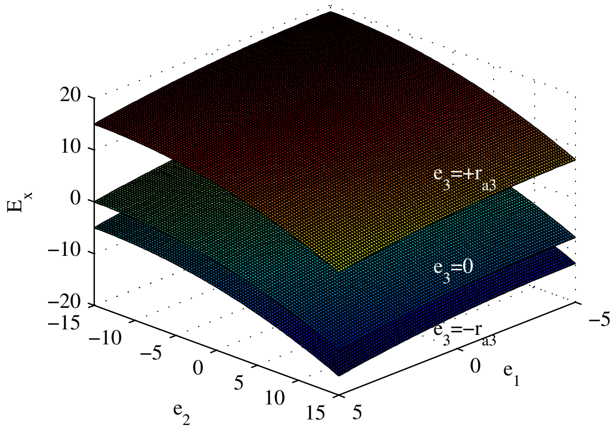


Figure 5.10: Error E_x as a function of e_1 and e_2 for three different constant values of e_3 i.e. $e_3 = -r_{a3}$, $e_3 = 0$ and $e_3 = +r_{a3}$.

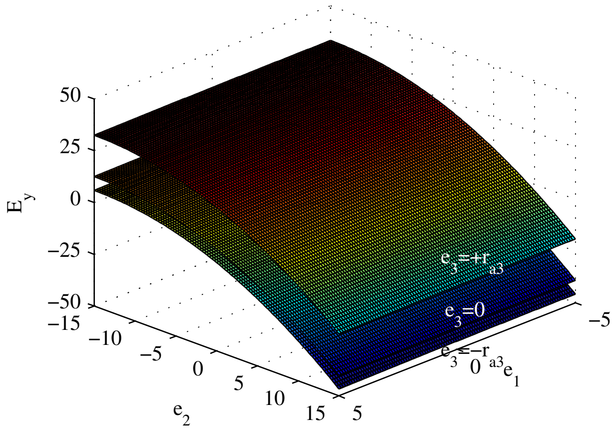


Figure 5.11: Error E_y as a function of e_1 and e_2 for three different constant values of e_3 i.e. $e_3 = -r_{a3}$, $e_3 = 0$ and $e_3 = +r_{a3}$.

values of E_x and E_y. It can also be observed that the distance between graphs of E_x at $e_3 = +r_{a3}$ and $e_3 = 0$ is three times the distance between graphs of E_x at $e_3 = 0$ and $e_3 = -r_{a3}$. A similar observation can be made about the plots of E_y. This again shows that the localisation error due to positive range estimation error equal to the actual distance is three times higher compared to error due to the same magnitude of negative range estimation error.

5.2.1.4 Localisation error (e_l)

From the results in Table 5.8, we observe that $e_{lmax} = 49.41$ which occurs at $(5, 15, -10)$ in accordance with (5.174) and observations in Section 5.1.4. At the corresponding points $(5, -10)$ and $(15, -10)$, E_1 and E_2 have their simultaneous maximum values $E_{1max} = 175$ and $E_{2max} = 775$ respectively. E_x and E_y have their simultaneous minimum values $E_{xmin} = -16.25$ and $E_{ymin} = -46.67$ at $(5, 15, -10)$. Furthermore, $e_{lp^-} = 6.78$ and $e_{lp^+} = 20.35$ for $p = 1$ and $\xi = 3$ in accordance with results in (5.220)-(5.223) derived in Section 5.1.5.2. We plot localisation error e_l as a function of e_1 and e_2 for three different values of e_3 in Fig. 5.13. In accordance with the results in Table 5.8, the maximum value of e_l occurs at $(5, 15, -10)$. It is also evident that the localisation error is lower at points where the range estimation errors are similar for the three beacon nodes.

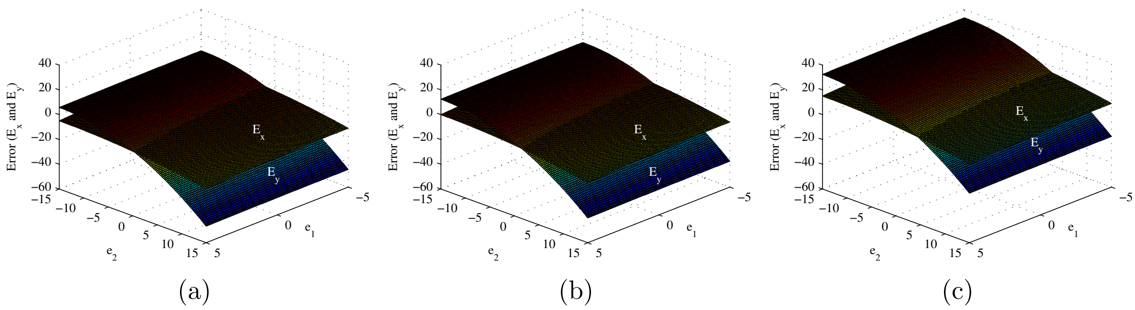
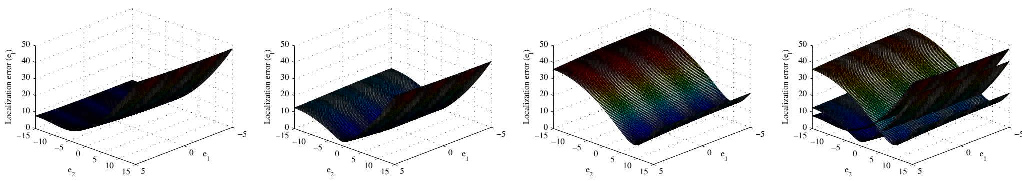


Figure 5.12: E_x and E_y as a function of e_1 and e_2 for different constant values of e_3 (a) $e_3 = -r_{a3}$ (b) $e_3 = 0$ (c) $e_3 = +r_{a3}$.

5.2.2 Localisation error due to equal distance estimation errors

The unknown node at $(30, 20)$ estimates its position using three beacon nodes whose positions are given in Table 5.7 when an equal amount of error is present in all the estimated distances i.e. $e_1 = e_2 = e_3$. Resulting node positions are shown in Fig. 5.14 and the numerical results are given in Table 5.9. As the positions of the beacon nodes remain unchanged, the localisation constants also remain the same for all repetitions of the experiment at $k_1 = -0.0375$, $k_2 = -0.0125$, $k_3 = -0.0083$ and $k_4 = -0.0583$. We perform three pairs of experiments. In the first version of each pair of experiment, we add an equal



(a) (b) (c) (d)

Figure 5.13: Localisation error e_l as a function of e_1 and e_2 for different constant values of e_3 (a) $e_3 = -r_{a3}$ (b) $e_3 = 0$ (c) $e_3 = +r_{a3}$ (d) combined plot of (a), (b) and (c).

Table 5.9: Localisation Error Due to Equal Distance Estimation Errors.

Experiment	Distance estimation error	Estimated distances			Experimental results	Analytical results		Common results		
	e_i	r_1	r_2	r_3	(x, y)	E_1	E_2	E_x	E_y	e_l
1	3	8	18	13	$(30.75, 18.50)$	-30	30	0.75	-1.50	1.68
	-3	2	12	7	$(29.25, 21.50)$	30	-30	-0.75	1.50	1.68
2	4	9	19	14	$(31.00, 18.00)$	-40	40	1.00	-2.00	2.24
	-4	1	11	6	$(29.00, 22.00)$	40	-40	-1.00	2.00	2.24
3	5	10	20	15	$(31.25, 17.50)$	-50	50	1.25	-2.50	2.80
	-5	0	10	5	$(28.75, 22.50)$	50	-50	-1.25	2.50	2.80

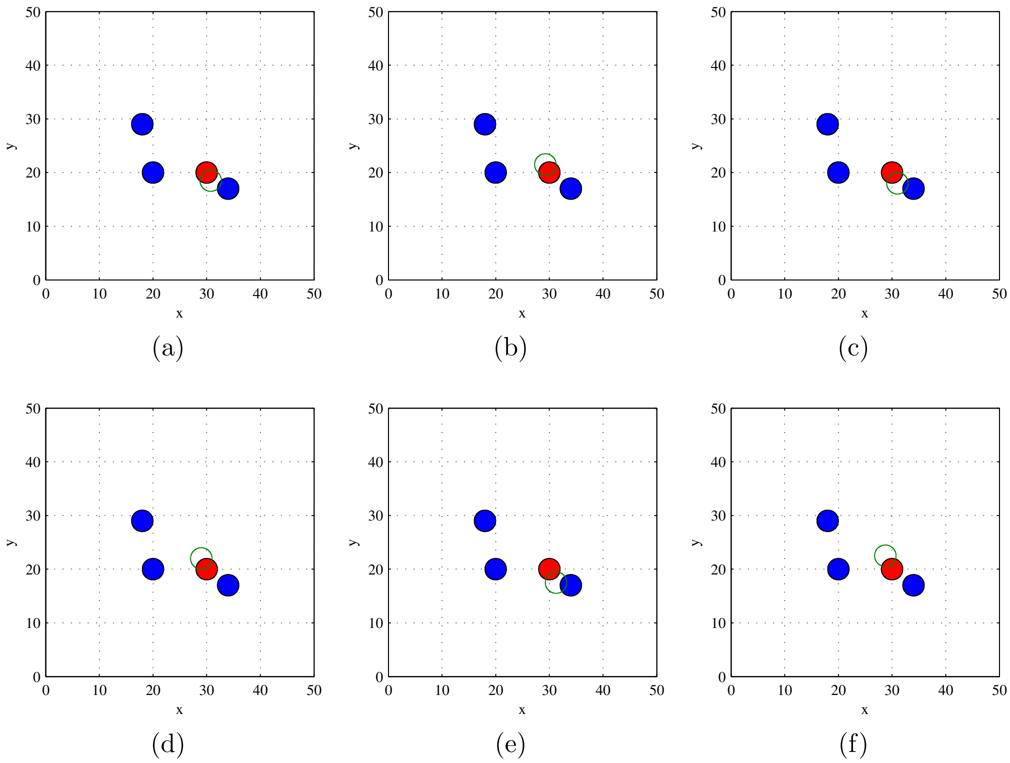


Figure 5.14: Trilateration error due to equal values of distance estimation errors.

amount of positive distance estimation error to all the distances. In the second version of the experiment, an equal but negative distance estimation error with the same magnitude as in the first version is added to all the distances. From the results in Table 5.9, it is evident that the estimated position in both the cases is different but the magnitude of the localisation error remains the same. Furthermore, magnitudes of E_1, E_2, E_x and E_y also remain the same. However, their signs are opposite in the two versions of an experiment. As a result, the estimated positions of the unknown node are 180° opposite but at an equal distance from the actual position. The results are in accordance with our derivations in Section 5.1.5.1.

5.2.3 Localisation error due to unequal distance estimation errors

Using unequal distance estimation errors proportionate to the actual distances, we run three pairs of experiments with unknown node at $(30, 20)$ and beacon nodes having positions and distances from the unknown node given in Table 5.7. In the first part of a pair of experiments, the distance estimation errors are negative according to (5.200). Positive distance estimation errors

Table 5.10: Localisation Error Due to Unequal Distance Estimation Errors.

Experiment	Error proportion		Distance estimation errors			Estimated distances			Experimental results	Analytical results		Common results		
	$\mathbf{p}$	ξ	e_1	e_2	e_3	r_1	r_2	r_3	(x, y)	E_1	E_2	E_x	E_y	e_l
1	-0.1	-1.11	-0.50	-1.50	-1.00	4.50	13.50	9.00	$(29.76, 21.27)$	14.25	-23.75	-0.24	1.27	1.29
	0.1		0.50	1.50	1.00	5.50	16.50	11.00	$(30.26, 18.60)$	-15.75	26.25	0.26	-1.40	1.42
2	-0.5	-1.67	-2.50	-7.50	-5.00	2.50	7.50	5.00	$(29.06, 25.00)$	56.25	-93.75	-0.94	5.00	5.09
	0.5		2.50	7.50	5.00	7.50	22.50	15.00	$(31.56, 11.67)$	-93.75	156.25	1.56	-8.33	8.48
3	-1.0	-3.00	-5.00	-15.00	-10.00	0.00	0.00	0.00	$(28.75, 26.67)$	75.00	-125.00	-1.25	6.67	6.78
	1.0		5.00	15.00	10.00	10.00	30.00	20.00	$(33.75, -0.00)$	-225.00	-375.00	3.75	-20.00	20.35

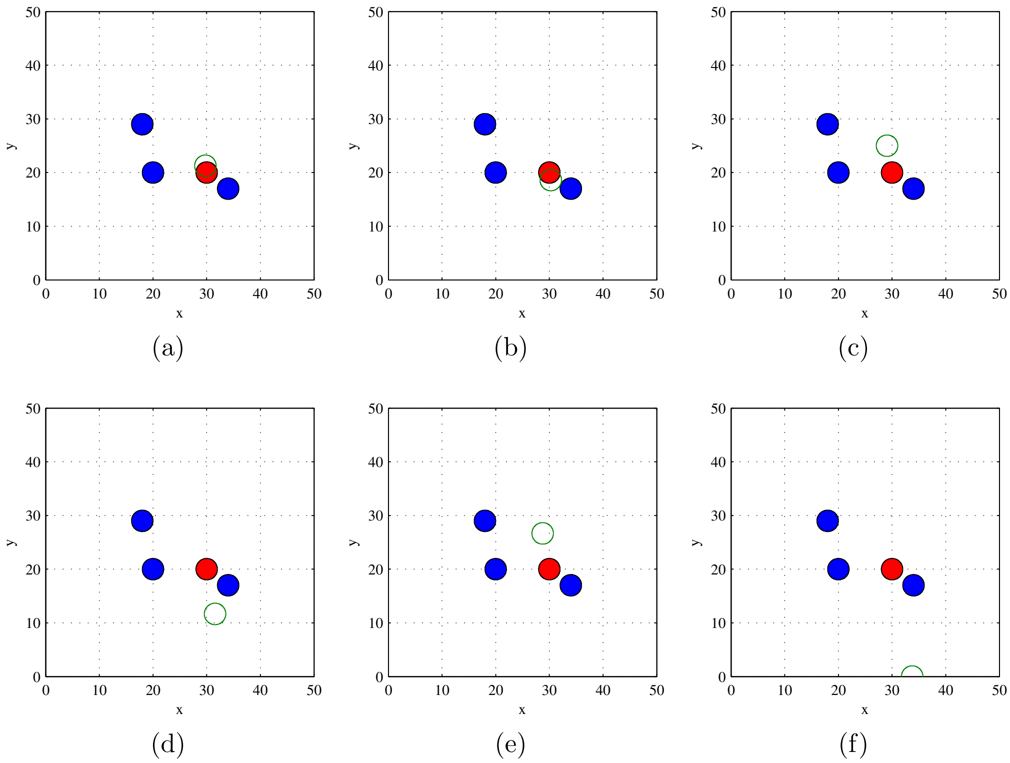


Figure 5.15: Trilateration error due to unequal values of distance estimation errors.

having the same magnitude as in the first part are used in the second part of the experiment. The node positions are shown in Fig. 5.15 and the results are given in Table 5.10. Both the estimated positions and the localisation error are different in either version of a pair of experiments. If E_x, E_y and e_l are designated as E_{xp^-}, E_{yp^-} and e_{lp^-} respectively for the first experiment with negative distance estimation errors and as E_{xp^+}, E_{yp^+} and e_{lp^+} in the second version of a pair of experiments with positive distance estimation errors in Table 5.10, then $|E_{xp^+}| = \xi|E_{xp^-}|$, $|E_{yp^+}| = \xi|E_{yp^-}|$ and $e_{lp^+} = \xi e_{lp^-}$ in accordance with our derivations in Section 5.1.5.2.

5.2.4 Geometry of nodes and distance estimation errors

To determine the effect of geometry of positions of beacon nodes on localisation error, we run a simulation experiment using an unknown node U_1 positioned at $(30, 20)$ with three beacon nodes B_1, B_2 and B_3. The experiment comprises of two parts with five repetitions in each part. In the first part of the experiment, the positions of the three beacon nodes and hence their actual distances from the unknown node remain fixed and are recorded in Table 5.11. In this way, the geometry formed by the positions of the nodes remains unchanged with each repetition of the experiment as is shown in Fig. 5.16. However, we introduce a variable range estimation error so that the estimated distances are different for each repetition. The unknown node estimates position (x, y) using multilateration and we measure experimental values of E_x, E_y and e_l using (4.40), (4.53) and (4.62). We also calculate k_1, k_2, k_3, k_4, E_1, E_2, E_x, E_y and e_l using analytically derived results in (4.44)-(4.46), (4.49), (4.50), (4.57), (4.58), (4.59) and (4.63). The resulting data are recorded in Table 5.12. E_x,

E_y and e_l are the same and are common in both experimental and analytical results as was also confirmed in Section 5.2.1. Therefore, these are recorded only once in Table 5.12. In the second part of the experiment, we change the positions of nodes and hence the resulting geometry but use the same set of distance estimation errors and estimated distances for each repetition of the experiment. For this purpose, we vary the positions of the beacon nodes such that their distances from the unknown node remain unchanged. In each next repetition of the second part of the experiment, we pick new position of a beacon node along the circumference of the circle with unknown node as the centre and its fixed distance from the beacon node as radius. In this way, the geometry formed by positions of nodes changes with each repetition of the experiment as shown in Fig. 5.17. However, the distance estimation errors and the estimated distances remain unchanged and are given in Table 5.13. Both experimental and analytical results are recorded in Table 5.14. In both Table 5.12 and Table 5.14, the localisation error and its x and y components E_x and E_y calculated using the analytical results are the same as the localisation error measured using the simulation experiment. This validates our analysis and the derived results. In Table 5.12, we observe that k_1, k_2, k_3 and k_4 remain unchanged as the geometry of the nodes does not change during first part of the experiment. However, E_1 and E_2 change with each next iteration of the experiment. This change in E_1, E_2 and localisation error e_l is due only to the change in the distance estimation errors. In Table 5.14. k_1, k_2, k_3 and k_4 have different values for different iterations as the geometry of the nodes also changes with each iteration of the experiment. However, E_1 and E_2 remain unchanged as the distance estimation errors and estimated distances do not change in the second part of the experiment. The change in localisation error, in this case, from one iteration to the next is due only to the change in geometry formed by node positions. We also observe from Table 5.12 that the localisation error is comparatively larger when all the range estimation errors do not have the same sign.

Table 5.11: Positions and Distances of Beacon Nodes in a Fixed Geometry.

Beacon Node	Position	Actual Distance from Unknown Node
B_1	$(30, 34)$	14.00
B_2	$(40, 15)$	11.18
B_3	$(16, 8)$	18.44

Table 5.12: Positions of Beacon Nodes are Fixed but Distance Estimation Errors are Variable.

Experiment	Distance Estimation Error			Estimated Distances			Experimental Results	Analytical Results						Common Results		
	e_1	e_2	e_3	r_1	r_2	r_3	(x, y)	k_1	k_2	k_3	k_4	E_1	E_2	E_x	E_y	e_l
1	0.13	−0.40	−1.50	14.13	10.79	16.94	$(29.28, 19.30)$	0.0067	−0.0247	−0.0228	0.0133	56.56	44.25	−0.72	−0.70	1.00
2	−4.05	−0.66	2.09	9.95	10.52	20.53	$(31.18, 22.80)$	0.0067	−0.0247	−0.0228	0.0133	−178.53	−95.93	1.18	2.80	3.04
3	−3.84	−4.22	−1.31	10.16	6.96	17.13	$(30.43, 20.66)$	0.0067	−0.0247	−0.0228	0.0133	−46.27	−30.03	0.43	0.66	0.79
4	−4.66	−3.08	−0.29	9.34	8.10	18.15	$(30.55, 21.59)$	0.0067	−0.0247	−0.0228	0.0133	−98.35	−48.88	0.55	1.59	1.69
5	−3.55	2.18	1.62	10.45	13.36	20.06	$(29.23, 23.28)$	0.0067	−0.0247	−0.0228	0.0133	−149.07	−8.80	−0.77	3.28	3.37

Table 5.13: Fixed Distances of Beacon Nodes in a Variable Geometry.

Beacon Node	Actual Distance from Unknown Node	Distance Estimation Error	Estimated Distance
B_1	14.00	0.13	14.13
B_2	11.18	−0.40	10.79
B_3	18.44	−1.50	16.94

Table 5.14: Positions of Beacon Nodes are Variable but Distance Estimation Errors are Fixed.

Experiment	Beacon Node Position			Experimental Results	Analytical Results						Common Results		
	B_1	B_2	B_3	(x, y)	k_1	k_2	k_3	k_4	E_1	E_2	E_x	E_y	e_l
1	$(30.00, 34.00)$	$(40.00, 15.00)$	$(16.00, 8.00)$	$(29.28, 19.30)$	0.0067	−0.0247	−0.0228	0.0133	56.56	44.25	−0.72	−0.70	1.00
2	$(43.82, 22.26)$	$(39.83, 14.67)$	$(24.39, 2.44)$	$(28.65, 19.90)$	0.0895	−0.1450	−0.1129	0.1421	56.56	44.25	−1.35	−0.10	1.36
3	$(16.41, 16.63)$	$(21.78, 27.58)$	$(47.49, 25.83)$	$(30.87, 20.14)$	0.0030	0.0158	0.0442	−0.0534	56.56	44.25	0.87	0.14	0.88
4	$(43.27, 24.46)$	$(36.11, 10.63)$	$(14.98, 30.69)$	$(29.01, 20.07)$	−0.0230	0.0071	−0.0242	0.0325	56.56	44.25	−0.99	0.07	0.99
5	$(17.14, 14.48)$	$(26.38, 30.58)$	$(30.23, 1.56)$	$(31.62, 19.45)$	0.0439	−0.0196	0.0058	−0.0198	56.56	44.25	1.62	−0.55	1.71

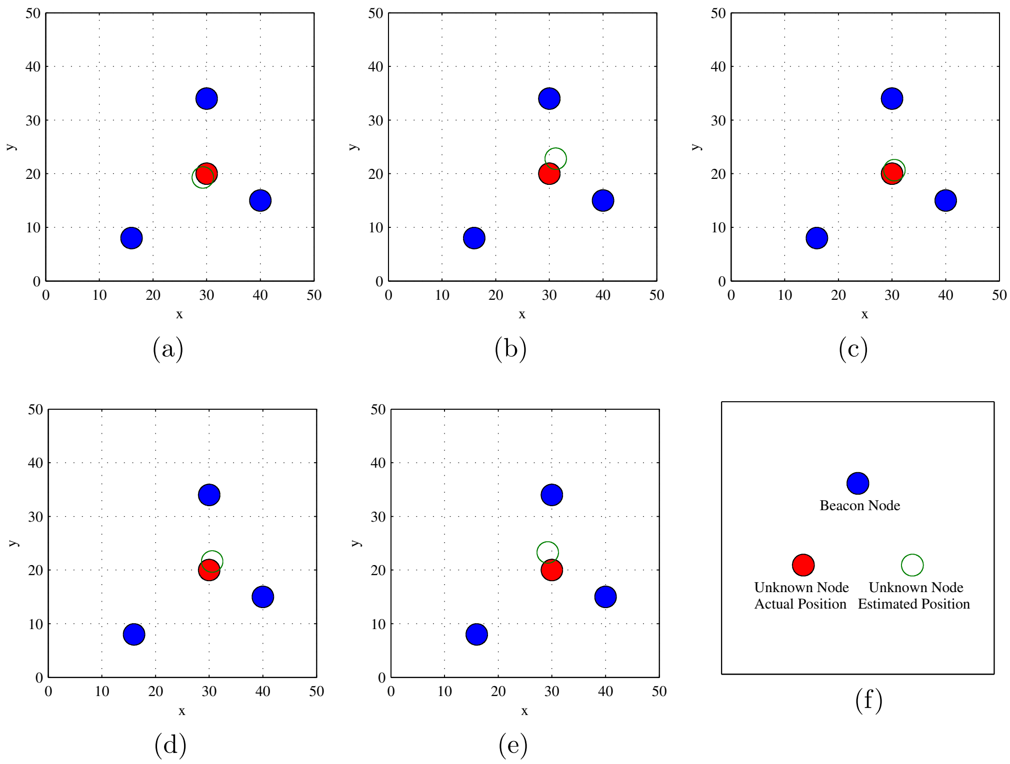


Figure 5.16: Multilateration analysis when geometry of nodes is unchanged but distance estimation errors are variable (a)-(e) Node positions for iteration 1 to 5. (f) Legend.

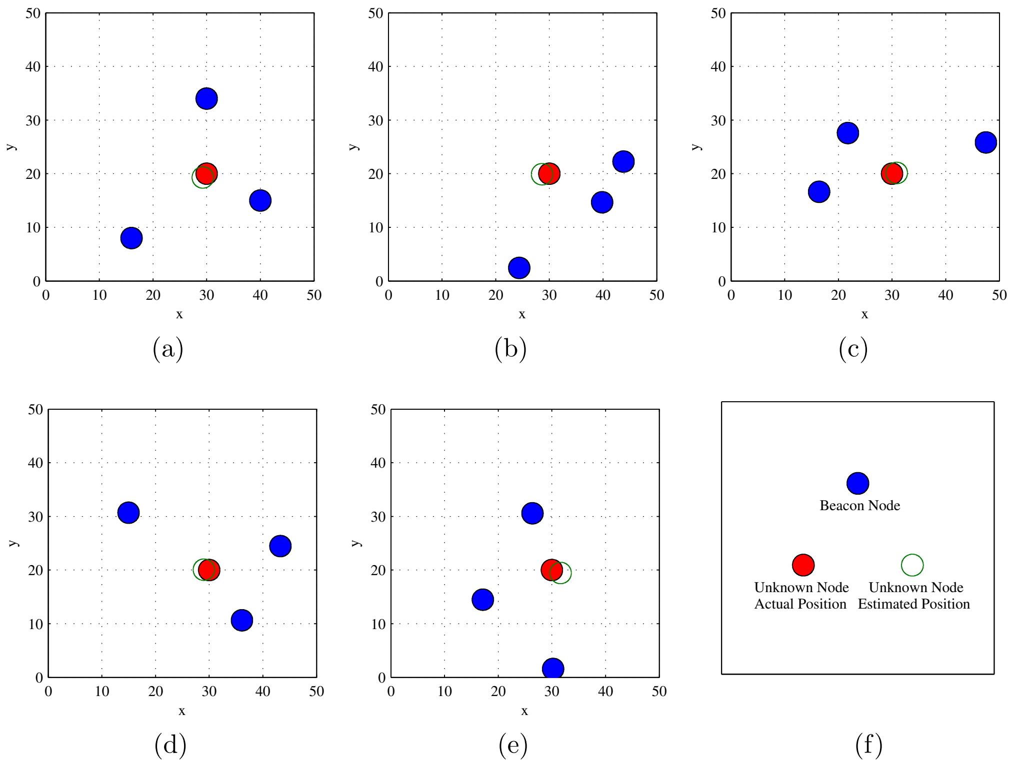


Figure 5.17: Multilateration analysis when geometry of nodes changes but distances and distance estimation errors remain the same (a)-(e) Node positions for iteration 1 to 5. (f) Legend.

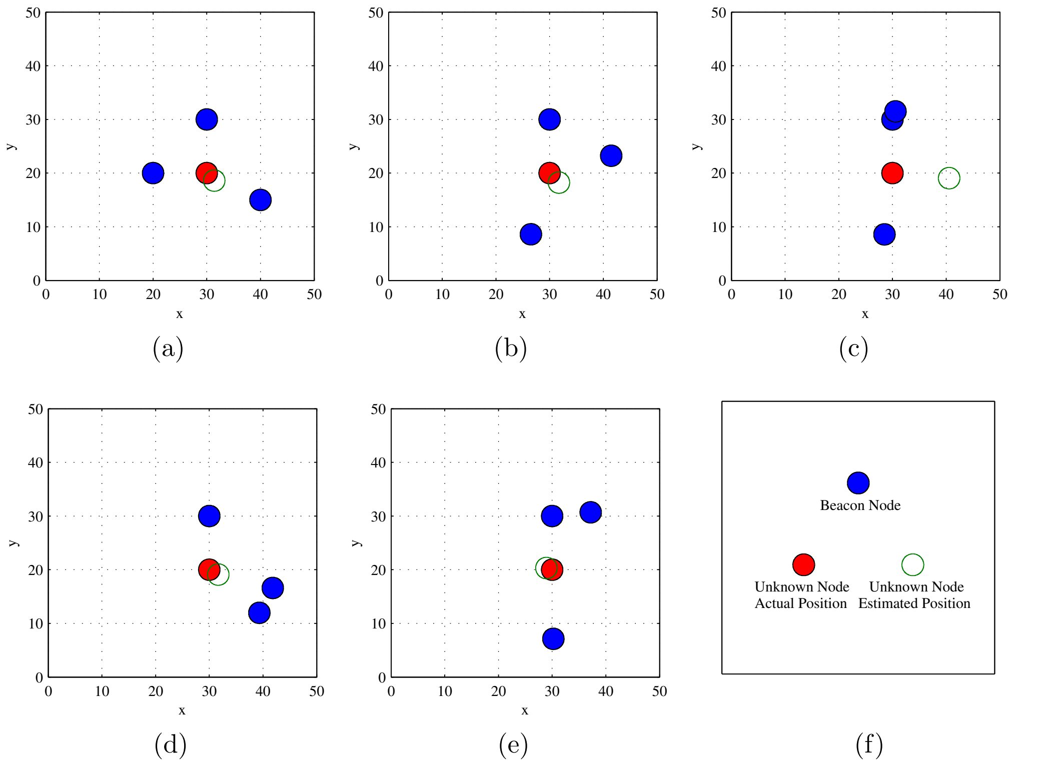


Figure 5.18: Multilateration analysis with two equidistant beacon nodes (a)-(e) Node positions for iteration 1 to 5. (f) Legend.

5.2.5 Two equidistant beacon nodes

To investigate localisation error when two of the three neighbour beacon nodes are equidistant from the unknown node, we deploy three beacon nodes along with an unknown node in a simulation experiment as shown in Fig. 5.18. The results are given in Table 5.15. The positions of beacon nodes are chosen such that $r_{a1} = r_{a3}$ for odd numbered experiments 1 and 3, and $r_{a2} = r_{a3}$ for even numbered experiments 2 and 4. For the last experiment 5, $r_{a1} = r_{a2}$. It can be observed from the results that $E_1 = 0$ when $r_{a1} = r_{a3}$ and $e_1 = e_3$. Similarly, $E_2 = 0$ when $r_{a2} = r_{a3}$ and $e_2 = e_3$. For experiment 5, $E_1 = E_2$ as $r_{a1} = r_{a2}$ and $e_1 = e_2$. The first version of the experiment with results in the first row of Table 5.15 is a special case. The beacon nodes with $r_{a1} = r_{a3}$ and $e_1 = e_3$

Table 5.15: Localisation with Two Equidistant Beacon Nodes.

Experiment	Beacon Node Position			Actual Distances			Estimated Distances			Localisation Errors				
	B_1	B_2	B_3	r_{a1}	r_{a2}	r_{a3}	r_1	r_2	r_3	E_1	E_2	E_x	E_y	e_l
1	$(20.00, 20.00)$	$(40.00, 15.00)$	$(30.00, 30.00)$	10.00	11.18	10.00	14.03	12.34	14.03	0.00	-69.52	1.39	-1.39	1.97
2	$(30.00, 30.00)$	$(41.47, 23.23)$	$(26.52, 8.61)$	10.00	11.92	11.92	11.16	10.15	10.15	63.58	0.00	1.73	-1.77	2.47
3	$(28.51, 8.58)$	$(30.00, 30.00)$	$(30.56, 31.50)$	11.51	10.00	11.51	9.56	8.23	9.56	0.00	9.01	10.55	-0.94	10.59
4	$(30.00, 30.00)$	$(41.82, 16.60)$	$(39.32, 11.97)$	10.00	12.30	12.30	8.23	7.45	7.45	63.69	0.00	1.67	-0.90	1.90
5	$(30.23, 7.13)$	$(37.18, 30.69)$	$(30.00, 30.00)$	12.87	12.87	10.00	10.35	10.35	5.14	14.88	14.88	-1.07	0.31	1.11

are positioned such that $|x_3 - x_1| = |y_3 - y_1|$ so that $|k_2| = |k_4|$. As a result, $E_1 = 0$ and $|E_x| = |E_y|$. The experiment verifies results derived in Section 5.1.2.

5.2.6 Two equidistant collinear beacon nodes parallel to an axis

We deploy three beacon nodes with an unknown node such that the two of the beacon nodes are collinear and are equidistant to the unknown node. In addition, the two nodes are parallel to either x-axis or y-axis. The deployed

Table 5.16: Beacon nodes when localising with two equidistant and collinear beacon nodes.

Experiment	Beacon Node Position			Actual Distances			Distance Estimation Error			Estimated Distance		
	B_1	B_2	B_3	r_{a1}	r_{a2}	r_{a3}	e_1	e_2	e_3	r_1	r_2	r_3
1	$(20, 30)$	$(40, 30)$	$(30, 10)$	14.14	14.14	10.00	4	4	1	18.14	18.14	11.00
2	$(10, 25)$	$(36, 8)$	$(24, 8)$	20.62	13.42	13.42	-1	-2	-2	19.62	11.42	11.42
3	$(37, 11)$	$(38, 41)$	$(23, 11)$	11.40	22.47	11.40	5	0	5	16.40	22.47	16.40
4	$(20, 10)$	$(20, 30)$	$(6, 27)$	14.14	14.14	25.00	-3	-3	1	11.14	11.14	26.00
5	$(9, 21)$	$(39, 11)$	$(39, 29)$	21.02	12.73	12.73	-4	3	3	17.02	15.73	15.73
6	$(38, 12)$	$(23, 9)$	$(38, 28)$	11.31	13.04	11.31	-1	-2	-1	10.31	11.04	10.31

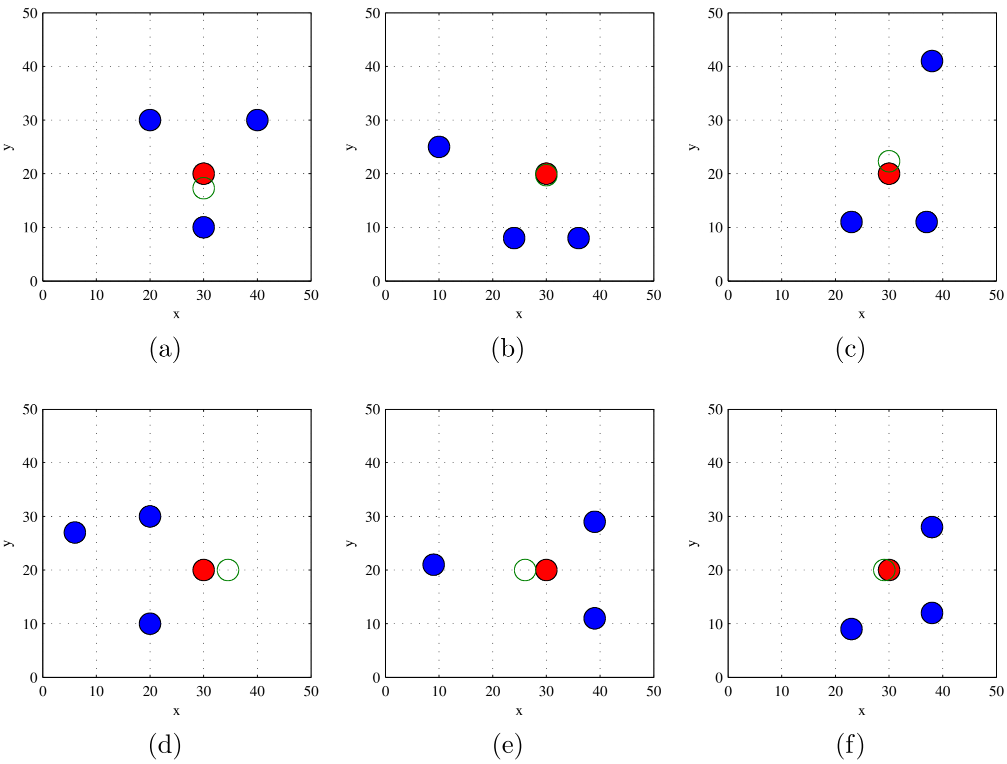


Figure 5.19: Multilateration analysis with two equidistant beacon nodes collinear to an axis (a)-(c) two equidistant collinear nodes parallel to x-axis (d)-(f) two equidistant collinear nodes parallel to y-axis.

nodes are shown in Fig. 5.19. The details of the beacon nodes are given in Table 5.16. Experimental and analytical results are given in Table 5.17 and are in conformance with each other. We obtain the common results in terms of E_x, E_y and e_l from both the experimental and analytical results. Furthermore, $E_x = 0$ or $E_y = 0$ when two collinear and equidistant beacon nodes are parallel to the x- or y-axis respectively in accordance with our analysis in Section 5.1.3.

5.2.7 All equidistant beacon nodes

To test localisation when all the neighbour beacon nodes are equidistant, we deploy an unknown node with three beacon nodes in a simulation experiment. In each iteration, the beacon nodes are equidistant from the unknown node such that $r_{a1} = r_{a2} = r_{a3}$ as is shown in Fig. 5.20. The distance estimation errors are also equal so that $e_1 = e_2 = e_3$. From the results in Table 5.18, it can be seen that the unknown node is able to determine its position accurately with zero localisation error from mere information of position coordinates of beacon nodes irrespective of the estimated distances. For example, the unknown sensor node is able to localise itself even if the distances are assumed to be zero. As

Table 5.17: Results when two equidistant beacon nodes are collinear and parallel to an axis.

Experiment	Experimental Results (Estimated Position)	Analytical Results						Common Results		
	(x, y)	k_1	k_2	k_3	k_4	E_1	E_2	E_x	E_y	e_l
1	$(30, 17.30)$	0.0250	-0.0250	-0.0125	-0.0125	108.14	108.14	0	-2.70	2.70
2	$(30, 19.72)$	0	-0.0417	-0.0294	-0.0343	9.43	0	0	-0.28	0.28
3	$(30, 22.32)$	-0.0357	0	0.0179	-0.0167	0	-139.02	0	2.32	2.32
4	$(34.53, 20)$	-0.0054	-0.0304	0.0250	-0.0250	-126.85	-126.85	4.53	0	4.53
5	$(26.04, 20)$	0.0167	-0.0074	0	0.0278	-237.56	0	-3.96	0	3.96
6	$(29.12, 20)$	-0.0396	0.0333	0.0313	0	0	-26.53	-0.88	0	0.88

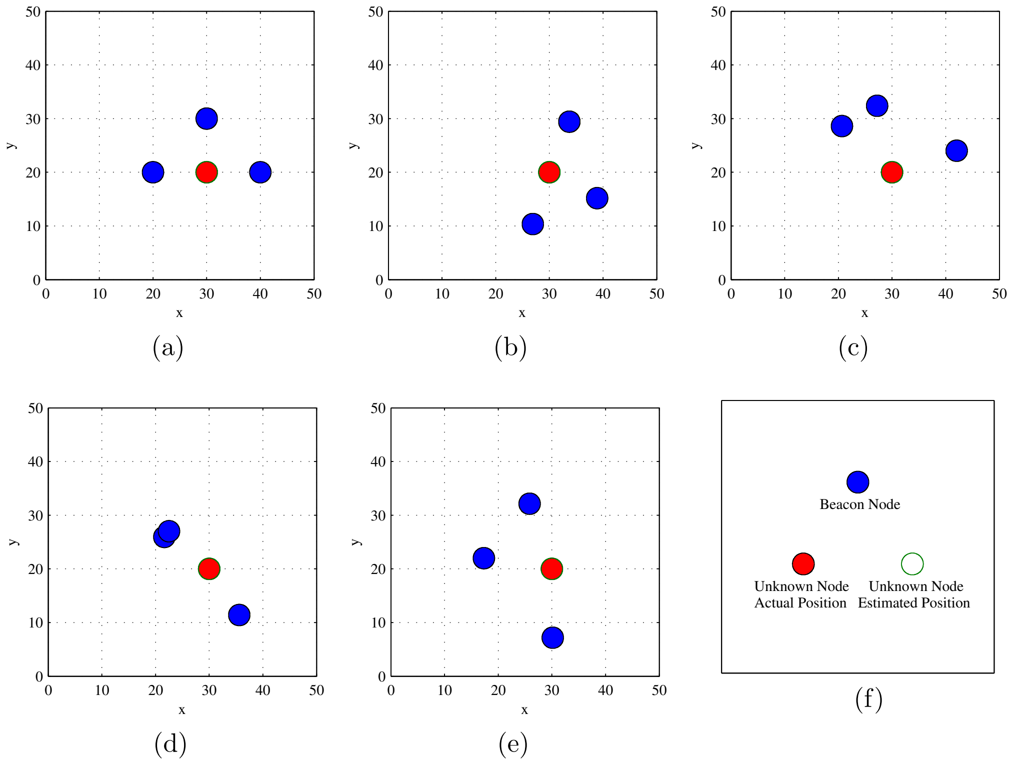


Figure 5.20: Multilateration analysis with all equidistant beacon nodes (a)-(e) Node positions for iteration 1 to 5. (f) Legend.

Table 5.18: Localisation With All Equidistant Beacon Nodes ($E_1 = E_2 = 0$, $E_x = E_y = 0$, $e_l = 0$).

Experiment	Beacon Node Position			Actual Distance $r_{a1} = r_{a2} = r_{a3}$	Distance Estimation Error $e_1 = e_2 = e_3$	Estimated Distance $r_1 = r_2 = r_3$	Estimated Position	Localisation Constants			
	B_1	B_2	B_3	r_{ai}	e_i	$r_i = r_{ai} + e_i$	(x, y)	k_1	k_2	k_3	k_4
1	(20.00, 20.00)	(40.00, 20.00)	(30.00, 30.00)	10.00	−10.00	0.00	(30, 20)	0.0250	−0.0250	0.0250	0.0250
2	(38.91, 15.17)	(26.92, 10.35)	(33.72, 29.42)	10.13	7.35	17.48	(30, 20)	−0.0487	0.0364	0.0174	0.0132
3	(20.67, 28.60)	(42.04, 24.01)	(27.22, 32.39)	12.69	−3.98	8.71	(30, 20)	0.0377	−0.0170	0.0668	0.0295
4	(35.60, 11.41)	(21.64, 25.95)	(22.51, 27.01)	10.26	4.69	14.95	(30, 20)	−0.0193	0.2829	0.0158	0.2373
5	(25.85, 32.13)	(30.16, 7.18)	(17.33, 21.99)	12.82	−6.63	6.19	(30, 20)	−0.0289	−0.0198	−0.0250	0.0166

distances as well as the estimation errors are equal, the positive and negative terms in (4.49) and (4.50) are equal and cancel out resulting in $E_1 = E_2 = 0$ and hence $E_x = E_y = 0$. As a result, the localisation error $e_l = 0$ and the node is able to determine its accurate position.

The preceding analysis gives a good insight into the composition of localisation errors and their relationship with the range estimation errors. However, the analysis has limitations. It is limited only to errors resulting from three neighbour beacon nodes and a planar sensor field. Furthermore, the analysis can be used in improving the localisation accuracy only if the errors in the distances can be approximated.

5.3 Summary

Using the analytical model developed in the preceding chapter, we have analysed localisation error in applications where the distance estimation errors are comparable to the actual distances. In particular, we have analysed localisation error and its various components under the condition of extreme distance estimation errors and determined the minimum and maximum values of localisation error. We have shown that positive distance estimation errors have more severe effect on localisation error compared to negative distance estimation errors. We have derived a number of additional novel and unique results which will help in the design and analysis of better and robust localisation algorithms. The analytical model and the derived results have been verified using simulation.

Chapter 6

Conclusion

This thesis has presented a number of contributions in the preceding chapters. In this concluding chapter, we review and reflect upon the work presented in the thesis. We also discuss possible research directions which can be based upon the work in this thesis.

6.1 Thesis Summary

In the beginning of this thesis, we have presented the system development life cycle of localisation algorithms for wireless sensor networks and critically analysed the performance evaluation techniques. When the positions of sensor nodes are estimated using a localisation algorithm, these may not be accurate. In other words, the estimated positions may have errors. Therefore, the error characteristics of a localisation algorithm play an important role in the decision of its deployment for a particular application. Hence performance evaluation of a localisation algorithm, which investigates and analyses error characteristics of a localisation algorithm is an important step in the system development life cycle of a localisation algorithm. In this thesis, we have presented the context and criteria for the performance evaluation of a localisation algorithm. We have also described and discussed a diverse range of performance metrics which are used in the literature for the measurement and testing of different types of errors and other aspects of localisation algorithms. The comprehensive review of performance evaluation and metrics intends to serve as a reference and guideline for testing of localisation algorithms. No review of such comprehensive nature is available in the previously published literature. We have also presented our own novel metrics which can be used for the evaluation of performance of localisation algorithms. These metrics are used for the performance evaluation of our proposed ripple localisation algorithm. The proposed new metrics evaluate aspects of localisation algorithms which are not covered by the previously available metrics.

We have also presented an new, novel, intelligent and energy efficient localisation algorithm which we have named ripple localisation algorithm. The algorithm is distributed so that each sensor node privately estimates its own position. The algorithm achieves very good levels of localisation accuracy, scalability, coverage and robustness while maintaining efficient utilisation of energy. This is a challenge which has not been addressed by the majority of previously available localisation algorithms. The sensor nodes do not transmit anything and only listen passively to beacon nodes for the purpose of localisation. Hence communication overhead is also avoided and the ripple localisation algorithm is energy efficient. Another factor, which contributes to the energy efficiency of the algorithm is the transmission of beacon signals with successive increment in the transmission power in the form of a ripple. An analysis of the energy utilisation of the algorithm shows that it saves 66.67% energy in the transmission of beacon signals compared to those algorithms that transmit beacons at fixed maximum power level. Simulation results show that the algorithm has a high localisation efficiency. It is able to localise all the sensor nodes with the help of only a few beacon nodes. Response time of the algorithm is also short and it is able to quickly localise a sensor node. The algorithm has a high degree of accuracy and yields better results than the two compared and previously published algorithms. It also provides control over localisation granularity and is therefore suitable for a wide range of applications requiring either coarse-grained or fine-grained estimation of sensor position. It is a completely new feature which is not present in the previously available localisation algorithms.

In this thesis, we have developed a new technique for solving multilateration equations. The new technique greatly simplifies the solution to multilateration problem. For example, one of the widely used previous techniques involves computation of an inverse. However, this is not required in our technique. We have shown that the overdetermined system of equations resulting from multilateration in two dimensions can be reduced to a set of two equations. These equations can then be solved simultaneously to estimate the two position coordinates (x, y) in a plane using conventional techniques, such as Cramer's rule. Based upon this new solution technique, we have developed and presented an accurate analytical model of trilateration error. There is no analytical model of localisation error available in the previously published literature. The model is a function of range estimation errors. Given the actual and estimated distances, it accurately predicts error in the estimated position. The analytical model specifies the complete localisation error vector in terms of magnitude and direction. The analytical model can be employed for the analysis of localisation error in any application where multilateration is used for position estimation. This includes short range wireless networks, such as wireless sensor networks (WSN), internet of things (IoT), wireless personal area networks (WPAN) and global networks, such as global navigation satellite system (GNSS) including global positioning system (GPS). However, in this thesis, we have restricted ourselves to localisation error analysis in short range wireless networks only where the range estimation errors are comparable to the actual distances between the nodes.

As another major contribution in this thesis, we have performed a comprehensive analysis of trilateration localisation error in wireless sensor networks using the analytical model that we have developed. Our investigation has focused on determining the conditions under which the localisation error is zero, minimum and maximum. For the purpose of this analysis, we have investigated the individual components of localisation error in addition to the whole localisation error. Consequently, we have derived a number of novel and useful results. We have determined the minimum and maximum values of localisation error and its various components for a given bound on range estimation errors. We have also determined conditions under which the localisation error can be zero. We have shown that the difference in the x and y components of localisation error depends only upon the geometry of placement or topology of beacon nodes. We have shown that, it is possible to determine accurate position even if the estimated distances are not accurate. We have further shown that an unknown node can determine its exact position by merely using the position coordinates of three neighbour beacon nodes and without determining distances from them if it can ascertain that it is equidistant from the beacon nodes. We have also shown that positive range estimation errors have more severe effect on the localisation error compared to negative range estimation errors. In particular, we have shown that the localisation error is three times higher when the range estimation errors equal to the actual distances are additive compared to when these are subtractive. All the presented results are novel and unique and no previous investigation in the existing literature has arrived at similar results. Therefore, these results are significant and provide a new insight into localisation. The analytical model and the results of error analysis are verified using a comprehensive set of simulation experiments.

6.2 Future Work

In this section, we give future research directions for each of the contributions made in this thesis.

The performance evaluation metrics proposed in this thesis have already been used and tested in the simulation experiments conducted for the performance evaluation of ripple localisation algorithm. As a future work, performance of various localisation algorithms can be tested and compared using the metrics presented in this thesis.

The design and analysis of a distributed, intelligent and energy efficient localisation algorithm has been presented in this thesis. The performance of the algorithm has been evaluated using simulation experiments. The work on the ripple localisation algorithm can be extended in many ways. First, the ripple of beacon signals can be used in combination with other techniques to develop hybrid localisation algorithms. Second, the algorithm considers a simple channel model of unobstructed and unconstrained sensor field. The work can be extended to investigate position estimation in more complex and hindered environments. This may include development of channel models of the complex hindered environments which are being investigated for deployment and position estimation of sensor nodes. Third, the concept of ripple can be explored for localisation in cellular networks and cognitive radio networks.

The analytical model of localisation error developed and presented in this thesis considers only three neighbour beacon nodes. It can be generalised for an arbitrary $k \geq 3$ number of neighbour beacon nodes. Furthermore, the model can be extended to three dimensional wireless sensor networks. The generalised and extended model can then be used for analysis of multilateration localisation error in three dimensional networks deploying three or more neighbour beacon nodes.

The analytical model and error analysis can also be applied to GNSS. In the GNSS, the magnitudes of distance estimation errors are much smaller compared to the distances between the satellites and the device being localised. Considering this fact, the error analysis can be applied to the GNSS and the error in the estimated position as a function of distance estimation errors can be investigated.

References

[1] S. Tomic, M. Beko, and R. Dinis, "3-D target localization in wireless sensor networks using RSS and AoA measurements," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 3197–3210, April 2017.

[2] M. Abdelhakim, Y. Liang, and T. Li, "Mobile coordinated wireless sensor network: An energy efficient scheme for real-time transmissions," *IEEE Journal on Selected Areas in Communications*, vol. 34, pp. 1663–1675, May 2016.

[3] R. A. Lara-Cueva, R. Gordillo, L. Valencia, and D. S. Bentez, "Determining the main CSMA parameters for adequate performance of WSN for real-time volcano monitoring system applications," *IEEE Sensors Journal*, vol. 17, pp. 1493–1502, March 2017.

[4] M. Balato, L. Costanzo, and M. Vitelli, "MPPT in wireless sensor nodes supply systems based on electromagnetic vibration harvesters for freight wagons applications," *IEEE Transactions on Industrial Electronics*, vol. 64, pp. 3576–3586, May 2017.

[5] U. I. Minhas, I. H. Naqvi, S. Qaisar, K. Ali, S. Shahid, and M. A. Aslam, "A WSN for monitoring and event reporting in underground

mine environments," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2017.

[6] M. Hammoudeh, F. Al-Fayez, H. Lloyd, R. Newman, B. Adebisi, A. Bounceur, and A. Abuarqoub, "A wireless sensor network border monitoring system: Deployment issues and routing protocols," *IEEE Sensors Journal*, vol. 17, pp. 2572–2582, April 2017.

[7] A. Mehmood, Z. Lv, J. Lloret, and M. M. Umar, "ELDC: An artificial neural network based energy-efficient and robust routing scheme for pollution monitoring in WSNs," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[8] E. Aguirre, P. Lopez-Iturri, L. Azpilicueta, A. Redondo, J. J. Astrain, J. Villadangos, A. Bahillo, A. Perallos, and F. Falcone, "Design and implementation of context aware applications with wireless sensor network support in urban train transportation environments," *IEEE Sensors Journal*, vol. 17, pp. 169–178, Jan 2017.

[9] K. Colins, L. Li, and Y. Liu, "Analysis of a statistical relationship between dose and error tallies in semiconductor digital integrated circuits for application to radiation monitoring over a wireless sensor network," *IEEE Transactions on Nuclear Science*, vol. PP, no. 99, pp. 1–1, 2017.

[10] H. Kim and J. Ben-Othman, "HeteRBar: Construction of heterogeneous reinforced barrier in wireless sensor networks," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–1, 2017.

[11] M. R. Ataei, T. Kunz, and A. H. Banihashemi, "Localization and location verification in non-homogeneous one-dimensional wireless

ad-hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1304–1316, July 2015.

[12] S. Zaidi, A. E. Assaf, S. Affes, and N. Kandil, "Accurate range-free localization in multi-hop wireless sensor networks," *IEEE Transactions on Communications*, vol. 64, pp. 3886–3900, Sept 2016.

[13] S. Tomic, M. Beko, R. Dinis, and P. Montezuma, "A closed-form solution for RSS/AoA target localization by spherical coordinates conversion," *IEEE Wireless Communications Letters*, vol. 5, pp. 680–683, Dec 2016.

[14] J. Chen, W. Dai, Y. Shen, V. K. N. Lau, and M. Z. Win, "Resource management games for distributed network localization," *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 317–329, Feb 2017.

[15] A. D. Angelis and C. Fischione, "Mobile node localization via pareto optimization: Algorithm and fundamental performance limitations," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1288–1303, July 2015.

[16] U. N. Okorafor and D. Kundur, "Security-aware routing and localization for a directional mission critical network," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 664–676, June 2010.

[17] H. Zhou, S. Xia, M. Jin, and H. Wu, "Localized and precise boundary detection in 3-D wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 23, pp. 1742–1754, Dec 2015.

[18] B. Sun, Y. Guo, N. Li, and D. Fang, "Multiple target counting and localization using variational bayesian em algorithm in wireless sensor

networks," *IEEE Transactions on Communications*, vol. PP, no. 99, pp. 1–1, 2017.

[19] C. F. Cheng and C. W. Wang, "The barrier-breach problem of barrier coverage in wireless sensor networks," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–1, 2017.

[20] C. Y. Wei, P. N. Chen, Y. S. Han, and P. K. Varshney, "Local threshold design for target localization using error correcting codes in wireless sensor networks in the presence of byzantine attacks," *IEEE Transactions on Information Forensics and Security*, vol. 12, pp. 1571–1584, July 2017.

[21] B. Zhou, Q. Chen, and P. Xiao, "The error propagation analysis of the received signal strength-based simultaneous localization and tracking in wireless sensor networks," *IEEE Transactions on Information Theory*, vol. PP, no. 99, pp. 1–1, 2017.

[22] K. Derr and M. Manic, "Wireless sensor networks—Node localization for various industry problems," *Industrial Informatics, IEEE Transactions on*, vol. 11, pp. 752–762, June 2015.

[23] D. Wu, D. Chatzigeorgiou, K. Youcef-Toumi, and R. Ben-Mansour, "Node localization in robotic sensor networks for pipeline inspection," *Industrial Informatics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[24] F. Chraim, Y. Erol, and K. Pister, "Wireless gas leak detection and localization," *Industrial Informatics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[25] X. Li, J. Yang, A. Nayak, and I. Stojmenovic, "Localized geographic routing to a mobile sink with guaranteed delivery in sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 30, pp. 1719–1729, October 2012.

[26] L. Shu, Y. Zhang, L. Yang, Y. Wang, M. Hauswirth, and N. Xiong, "TPGF: geographic routing in wireless multimedia sensor networks," *Telecommunication Systems*, vol. 44, no. 1-2, pp. 79–95, 2010.

[27] H. Mahboubi, K. Moezzi, A. Aghdam, K. Sayrafian-Pour, and V. Marbukh, "Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors," *Industrial Informatics, IEEE Transactions on*, vol. 10, pp. 163–174, Feb 2014.

[28] C. Zhu, C. Zheng, L. Shu, and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619 – 632, 2012. Simulation and Testbeds.

[29] L. LoBello and E. Toscano, "An adaptive approach to topology management in large and dense real-time wireless sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 5, pp. 314–324, Aug 2009.

[30] R. Xie and X. Jia, "Transmission-efficient clustering method for wireless sensor networks using compressive sensing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, pp. 806–815, March 2014.

[31] P. Oguz-Ekim, J. Gomes, J. Xavier, M. Stosic, and P. Oliveira, "An angular approach for range-based approximate maximum

likelihood source localization through convex relaxation," *Wireless Communications, IEEE Transactions on*, vol. 13, pp. 3951–3964, July 2014.

[32] N. Bulusu, *Self-Configuring Localization Systems*. PhD thesis, University of California, Los Angeles, 2002.

[33] C. Liu, K. Wu, and T. He, "Sensor localization with ring overlapping based on comparison of received signal strength indicator," in *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 516 – 518, October 2004.

[34] V. Vivekanandan and V. W. S. Wong, "Concentric anchor beacon localization algorithm for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 2733 – 2744, September 2007.

[35] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pp. 81 – 95, ACM, 2003.

[36] O.-H. Kwon, H.-J. Song, and S. Park, "Anchor-free localization through flip-error-resistant map stitching in wireless sensor network," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, pp. 1644–1657, Nov 2010.

[37] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 5, pp. 2926–2931 vol.5, 2001.

[38] M. L. Sichitiu, V. Ramadurai, and P. Peddabachagari, "Simple algorithm for outdoor localization of wireless sensor networks with inaccurate range measurements," in *Proceedings of International Conference on Wireless Networks*, (Las Vegas), pp. 300 – 305, 2003.

[39] J. M. Pak, C. K. Ahn, Y. Shmaliy, and M. T. Lim, "Improving reliability of particle filter-based localization in wireless sensor networks via hybrid particle/FIR filtering," *Industrial Informatics, IEEE Transactions on*, vol. 11, pp. 1089–1098, Oct 2015.

[40] L. Doherty, K. pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1655–1663 vol.3, 2001.

[41] H. Aksu, D. Aksoy, and I. Korpeoglu, "A study of localization metrics: Evaluation of position errors in wireless sensor networks," *Computer Networks*, vol. 55, no. 15, pp. 3562 – 3577, 2011.

[42] Y. X. Chu, J. B. Gou, H. Wu, and Z. X. Li, "Localization algorithms: performance evaluation and reliability analysis," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3652–3657, May 1998.

[43] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," Technical Report TR-892, Massachusetts Institute of Technology, Apr. 2003.

[44] L. M. P. L. de Brito and L. M. R. Peralta, “An Analysis of Localization Problems and Solutions in Wireless Sensor Networks,” *Tékhne - Revista de Estudos Politécnicos*, pp. 146 – 172, June 2008.

[45] M. Allen, S. Baydere, E. Gaura, and G. Kucuk, “Evaluation of localization algorithms,” in *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking* (G. Mao and B. Fidan, eds.), pp. 348–379, Hershey: IGI Global, 2009.

[46] A. Efrat, D. Forrester, A. Iyer, S. G. Kobourov, C. Erten, and O. Kilic, “Force-directed approaches to sensor localization,” *ACM Trans. Sen. Netw.*, vol. 7, pp. 27:1–27:25, Oct. 2010.

[47] J. Eckert, F. Villanueva, R. German, and F. Dressler, “Considerations on quality metrics for self-localization algorithms,” in *Self-Organizing Systems: 5th International Workshop, IWSOS 2011, Karlsruhe, Germany, February 23-24. 2011. Proceedings* (C. Bettstetter and C. Gershenson, eds.), pp. 104–115, Berlin, Heidelberg: Springer, 2011.

[48] M. L. Rajaram, E. Kougianos, S. P. Mohanty, and U. Choppali, “Wireless sensor network simulation frameworks: A tutorial review: MATLAB/Simulink bests the rest,” *IEEE Consumer Electronics Magazine*, vol. 5, pp. 63–69, April 2016.

[49] E. Egea-Lopez, J. Vales-Alonso, A. S. Martinez-Sala, P. Pavon-Marino, and J. García-Haro, “Simulation tools for wireless sensor networks,” in *Summer Simulation Multiconference, SPECTS*, pp. 2–9, 2005.

[50] M. Imran, A. M. Said, and H. Hasbullah, "A survey of simulators, emulators and testbeds for wireless sensor networks," in *2010 IEEE International Symposium on Information Technology*, vol. 2, pp. 897–902, June 2010.

[51] M. O. Farooq and T. Kunz, "Wireless sensor networks testbeds and state-of-the-art multimedia sensor nodes," *Applied Mathematics & Information Sciences*, vol. 8, no. 3, p. 935, 2014.

[52] M. Korkalainen, M. Sallinen, N. Krkkinen, and P. Tukeva, "Survey of wireless sensor networks simulation tools for demanding applications," in *2009 Fifth IEEE International Conference on Networking and Services*, pp. 102–106, April 2009.

[53] B. Musznicki and P. Zwierzykowski, "Survey of simulators for wireless sensor networks," *International Journal of Grid & Distributed Computing*, vol. 5, pp. 23 – 50, September 2012.

[54] Q. I. Ali, "Simulation framework of wireless sensor network (WSN) using MATLAB/SIMULINK software," in *MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 2* (V. N. Katsikis, ed.), ch. 12, Rijeka: InTech, 2012.

[55] J. Horneber and A. Hergenrder, "A survey on testbeds and experimentation environments for wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 1820–1838, Fourthquarter 2014.

[56] Z. Zhao, G. H. Yang, Q. Liu, V. O. K. Li, and L. Cui, "Implementation and application of a multi-radio wireless sensor networks testbed," *IET Wireless Sensor Systems*, vol. 1, pp. 191–199, December 2011.

[57] P. Checkland and S. Holwell, "Action research: Its nature and validity," *Systemic Practice and Action Research*, vol. 11, no. 1, pp. 9–21, 1998.

[58] L. Vencel and S. Cook, "7.4.1 methodology selection for the engineering of defence systems," in *INCOSE International Symposium*, vol. 15, pp. 1087–1100, Wiley Online Library, 2005.

[59] C. Gotsman and Y. Koren, "Distributed graph layout for sensor networks," *Lecture Notes in Computer Science*, pp. 273–284, 2005.

[60] A. A. Ahmed, H. Shi, and Y. Shang, "SHARP: a new approach to relative localization in wireless sensor networks," in *25th IEEE International Conference on Distributed Computing Systems Workshops*, pp. 892–898, June 2005.

[61] A. A. Kannan, B. Fidan, and G. Mao, "Analysis of flip ambiguities for robust sensor network localization," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 2057–2070, May 2010.

[62] S. Slijepcevic, S. Megerian, and M. Potkonjak, "Location errors in wireless embedded sensor networks: sources, models, and effects on applications," *Mobile Computing and Communications Review*, vol. 6, pp. 67–78, 2002.

[63] L. D. Girod, *A Self-calibrating System of Distributed Acoustic Arrays.* PhD thesis, University of California at Los Angeles, Los Angeles, CA, USA, 2005.

[64] J. Niu, B. Wang, L. Shu, T. Q. Duong, and Y. Chen, "Zil: An energy-efficient indoor localization system using zigbee radio to detect wifi fingerprints," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1431–1442, July 2015.

[65] M. S. Elgamel and A. Dandoush, "A modified manhattan distance with application for localization algorithms in ad-hoc WSNs," *Ad Hoc Networks*, vol. 33, pp. 168 – 189, 2015.

[66] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *Personal Communications, IEEE*, vol. 7, pp. 28–34, Oct 2002.

[67] J. Jiang, G. Han, H. Xu, L. Shu, and M. Guizani, "LMAT: Localization with a mobile anchor node based on trilateration in wireless sensor networks," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pp. 1–6, Dec 2011.

[68] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, (New York, NY, USA), pp. 166–179, ACM, 2001.

[69] W. Hong and K. Sarabandi, "Design of low-profile omnidirectional antenna for ground sensor networks," in *IEEE Antennas and Propagation Society International Symposium*, pp. 6007 – 6010, June 2007.

[70] A. Moragrega, P. Closas, and C. Ibars, "Potential game for energy-efficient rss-based positioning in wireless sensor networks," *IEEE*

Journal on Selected Areas in Communications, vol. 33, pp. 1394–1406, July 2015.

[71] P. Abouzar, D. G. Michelson, and M. Hamdi, "RSSI-based distributed self-localization for wireless sensor networks used in precision agriculture," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 6638–6650, Oct 2016.

[72] A. E. Assaf, S. Zaidi, S. Affes, and N. Kandil, "Low-cost localization for multihop heterogeneous wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 472–484, Jan 2016.

[73] T. E. Abrudan, Z. Xiao, A. Markham, and N. Trigoni, "Distortion rejecting magneto-inductive three-dimensional localization (MagLoc)," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 2404–2417, Nov 2015.

[74] S. Li, M. Hedley, and I. B. Collings, "New efficient indoor cooperative localization algorithm with empirical ranging error model," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1407–1417, July 2015.

[75] J. Niu, B. Wang, L. Shu, T. Q. Duong, and Y. Chen, "ZIL: An energy-efficient indoor localization system using zigbee radio to detect wifi fingerprints," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1431–1442, July 2015.

[76] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao, "Magicol: Indoor localization using pervasive magnetic field and opportunistic wifi

sensing," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1443–1457, July 2015.

[77] R. Diamant and L. Lampe, "Underwater localization with time-synchronization and propagation speed uncertainties," *IEEE Transactions on Mobile Computing*, vol. 12, pp. 1257–1269, July 2013.

[78] B. Li, S. Li, A. Nallanathan, and C. Zhao, "Deep sensing for future spectrum and location awareness 5G communications," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1331–1344, July 2015.

[79] F. Chiti, R. Fantacci, F. Archetti, E. Messina, and D. Toscani, "An integrated communications framework for context aware continuous monitoring with body sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 379–386, May 2009.

[80] K. Abrougui, A. Boukerche, and R. W. N. Pazzi, "Design and evaluation of context-aware and location-based service discovery protocols for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 717–735, Sept 2011.

[81] C. Luo, L. Cheng, M. C. Chan, Y. Gu, J. Li, and M. Zhong, "Pallas: Self-bootstrapping fine-grained passive indoor localization using WiFi monitors," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[82] E. Vattapparamban, B. S. Çiftler, İ. Güvenç, K. Akkaya, and A. Kadri, "Indoor occupancy tracking in smart buildings using passive sniffing of probe requests," in *2016 IEEE International Conference on Communications Workshops (ICC)*, pp. 38–44, May 2016.

[83] A. Giannitrapani, N. Ceccarelli, F. Scortecci, and A. Garulli, "Comparison of EKF and UKF for spacecraft localization via angle measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, pp. 75–84, January 2011.

[84] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more.* Springer Vienna, 2007.

[85] M. Ibrahim and M. Youssef, "CellSense: An accurate energy-efficient GSM positioning system," *IEEE Transactions on Vehicular Technology*, vol. 61, pp. 286–296, Jan 2012.

[86] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, pp. 41–53, July 2005.

[87] S. T. Pfister, *Algorithms for mobile robot localization and mapping, incorporating detailed noise modeling and multi-scale feature extraction.* PhD thesis, California Institute of Technology, 2006.

[88] W. Dai, Y. Shen, and M. Z. Win, "Energy-efficient network navigation algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1418–1430, July 2015.

[89] R. H. Wu, "Alternative terminal navigation based on modified airport multilateration system," in *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, pp. 5B4–1–5B4–13, Oct 2012.

[90] J. Van Etten, "Navigation systems. fundamentals of low-and very-low-frequency hyperbolic techniques," *Electrical Communication*, vol. 45, no. 3, 1970.

[91] H. Abdullah, I. Khan, and M. S. Zainal, "Proposed wireless sensor network in vehicles for road surveillance," in *Systems, Process Control (ICSPC), 2013 IEEE Conference on*, pp. 112–116, Dec 2013.

[92] A. Oka and L. Lampe, "Distributed target tracking using signal strength measurements by a wireless sensor network," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 1006–1015, September 2010.

[93] N. Xu, R. Cassell, C. Evers, S. Hauswald, and W. Langhans, "Performance assessment of multilateration systems - a solution to nextgen surveillance," in *Integrated Communications Navigation and Surveillance Conference (ICNS), 2010*, pp. D2–1–D2–8, May 2010.

[94] R. Young and S. Brenton, "Establishing baseline requirements for a UAS ground-based sense and avoid system," in *2016 Integrated Communications Navigation and Surveillance (ICNS)*, pp. 8D4–1–8D4–10, April 2016.

[95] T. P. Waldron, "Detecting airport surface movement events using ground surveillance," in *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*, pp. 4.C.1–1–4.C.1–8, Oct 2009.

[96] A. Szullo and R. Seller, "Maneuvering target tracking in wide area multilateration radar system," in *2016 17th International Radar Symposium (IRS)*, pp. 1–4, May 2016.

[97] B. G. Ferguson, K. W. Lo, and R. J. Wyber, "Acoustic sensing of direct and indirect weapon fire," in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pp. 167–172, Dec 2007.

[98] C. Thurber, *Advances in seismic event location.* Dordrecht Boston: Kluwer Academic Publishers, 2000.

[99] P. R. Escobal, H. F. Fliegel, R. M. Jaffe, P. M. Muller, K. M. Ong, O. H. von Roos, and M. S. Shumate, "3-D multilateration: A precision geodetic measurement system," *JPL Quarterly Technical Review*, vol. 2, October 1972.

[100] P. R. Escobal, K. M. Ong, O. H. von Roos, M. S. Shumate, R. M. Jaffe, H. F. Fliegel, and P. M. Muller, "3-D multilateration: A precision geodetic measurement system," *Technical Memorandum 33-605*, March 1973.

[101] B. Wellenhof, *Navigation : Principles of Positioning and Guidance.* Vienna: Springer Vienna, 2003.

[102] Q. Fengzhong, W. Shiyuan, W. Zhihui, and L. Zubin, "A survey of ranging algorithms and localization schemes in underwater acoustic sensor network," *China Communications*, vol. 13, pp. 66–81, March 2016.

[103] A. Zanella, "Best practice in RSS measurements and ranging," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 2662–2686, Fourthquarter 2016.

[104] R. K. Mahapatra and N. S. V. Shet, "Experimental analysis of RSSI-based distance estimation for wireless sensor networks," in *2016*

IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), pp. 211–215, Aug 2016.

[105] Y. Trofimova, "Multilateration error investigation and classification. error estimation," *Transport and Telecommunication*, vol. 8, no. 2, pp. 28 – 37, 2007.

[106] F. Reichenbach, A. Born, D. Timmermann, and R. Bill, *Splitting the Linear Least Squares Problem for Precise Localization in Geosensor Networks*, pp. 321–337. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[107] R. G. Stansfield, "Statistical theory of d.f. fixing," *Electrical Engineers - Part IIIA: Radiocommunication, Journal of the Institution of*, vol. 94, pp. 762–770, March 1947.

[108] J. F. Cline, "Multilateration error ellipsoids," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-14, pp. 665–667, July 1978.

[109] J. Fan and G. Ma, "Characteristics of GPS positioning error with non-uniform pseudorange error," *GPS Solutions*, vol. 18, no. 4, pp. 615–623, 2014.

[110] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2137–2148, Aug 2003.

[111] E. Elnahrawy, X. Li, and R. P. Martin, "The limits of localization using signal strength: a comparative study," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc*

Communications and Networks, 2004. IEEE SECON 2004., pp. 406–414, Oct 2004.

[112] J. CHOI, J.-S. KANG, Y.-H. KIM, and S.-C. KIM, "Modeling of localization error in wireless sensor network," *IEICE Transactions on Communications*, vol. E92.B, no. 2, pp. 628–631, 2009.

[113] M. Di, E. M. Joo, L. H. Beng, and W. Bang, "A lower bound on expected localization error in wireless sensor network," in *2010 11th International Conference on Control Automation Robotics Vision*, pp. 96–103, Dec 2010.

[114] Y. W. E. Chan and B. H. Soong, "A new lower bound on range-free localization algorithms in wireless sensor networks," *IEEE Communications Letters*, vol. 15, pp. 16–18, January 2011.

[115] M. R. Gholami, E. G. Ström, H. Wymeersch, and S. Gezici, "Upper bounds on position error of a single location estimate in wireless sensor networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 4, 2014.

[116] M. R. Gholami, E. G. Strm, H. Wymeersch, and M. Rydstrm, "On geometric upper bounds for positioning algorithms in wireless sensor networks," *Signal Processing*, vol. 111, pp. 179 – 193, 2015.

Index

F

G

H

I

L

N

S