Sensor Network-Based Indoor Localization and Tracking for Emergency Situations

Hewa Halpage Samiru Gayan

(138030 H)

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Philosophy

Department of Electronic and Telecommunication Engineering

University of Moratuwa
Sri Lanka

March 2015

Declaration

I declare that this is my own work, and this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or institute of higher learning, and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part, in print, electronic, or any other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 02/04/2015

The candidate, whose signature appears above, carried out research for the MPhil dissertation under my supervision.

UOM Verified Signature

2/4/2015

Signature: Date:

Abstract

Wireless Sensor Networks (WSNs) are application-specific systems, each having its own requirements related to the design. Using WSNs for emergency rescue operations is one such special application having localization of sensor nodes in a simple manner, tracking of moving nodes, usually worn by rescue workers and navigation support for rescue workers, as its major requirements. The overall objective of this research is to develop a suit of algorithms for localization, tracking and navigation of wireless sensor nodes in multistory indoor environments in emergency situations.

We base our research on the DV-Hop (Distance Vector) algorithm, which is an attractive option for the localization of nodes in a wireless sensor network due to its simplicity. We carry out a comprehensive study of the DV-Hop algorithm and its variations through literature review and computer simulations. We then evaluate its performance in emergency situations, where nodes may perish, new nodes may be introduced, and communications links may be disrupted and new links set up. We then propose a new algorithm for the improvement of localization accuracy of the DV-Hop algorithm. The new algorithm is based on optimizing the Hop Size estimation in the original algorithm, which is its key source of error.

We next present a new approach for target tracking in WSNs by combining the DV-Hop algorithm with Kalman filtering. The DV-Hop algorithm is used for pre-localization of the target and measurement conversion. Finally, we present a novel navigation support algorithm for rescue personnel in emergency situations by emulating virtually through WSN nodes, the *lifeline* used by the fire fighters.

The key contribution of this work is the development of WSN localization and tracking techniques which are distributed in nature and resilient in emergency situations.

Index terms— DV-Hop, Localization, Wireless Sensor Networks, Target Tracking, Navigation Support

To My Parents

Acknowledgements

First and foremost, I would like to extend my deepest gratitude to my supervisor Prof. Dileeka Dias for her continuous guidance, advice and support throughout my research.

I wish to extend my earnest gratitude to members of my project panel Prof. K.S. Walgama, Dr. K.C.B.Wavegedara and Dr. Dilum Bandara, for their valuable comments on my research work.

I also thank my dear friends and colleagues in department of Electronic and Telecommunication Engineering, University of Moratuwa for their assistance and encouragement.

I would like to thank National research Council, Sri Lanka for funding my research under the grant number 11-113 for the entire duration to make it a success.

Last but not lease I would like to thank Zone 24 $\times$ 7 for providing additional support for my research.

Contents

Declaration i

Abstract ii

Acknowledgements iv

1 INTRODUCTION 1

1.1 Wireless Sensor Networks . . . 1

1.2 Emergency Situations and Application of WSNs . . . 2

1.2.1 Fire Rescue . . . 2

1.2.2 Requirements for Systems to Support Fire Rescue Operations 3

1.3 Basic Functions Required of a WSN to Support Rescue Operations 4

1.3.1 Localization . . . 5

1.3.2 Target Tracking . . . 5

1.3.3 Navigation Support . . . 5

1.4 Challenges for Localization, Tracking and Navigation Support in Emergency Situations . . . 6

1.5 Scope . . . 7

1.6 Objectives . . . 7

1.7 Methodology . . . 8

1.8 Organization of the Thesis . . . 9

2 RELATED WORK 11

2.1 Introduction . . . 11

2.2 Overview of Localization . . . 11

2.2.1 Centralized vs Distributed Algorithms . . . 12

2.2.2 Range Based vs Range Free Algorithms . . . 12

2.2.3 Event-Driven Localization . . . 13

2.3 Node Self Localization . . . 13

2.3.1 Range Based Localization . . . 13

2.3.1.1 Ranging Techniques . . . 13

2.3.1.2 Location Estimation 15
2.3.2 Range Free Localization 18
2.4 The DV-Hop algorithm 20
2.5 Review of Improvements to the DV-Hop Localization Algorithm . 23
2.5.1 Improvements using Modified Hop Size 23
2.5.2 Improvements using Controlled Deployment of Anchor Nodes 24
2.5.3 Improvements combining RSSI with the DV-Hop Algorithm 25
2.5.4 Improvements using Other Techniques 25
2.6 Overview of Target Tracking in WSNs 27
2.6.1 Related Work 28

3 PERFORMANCE EVALUATION OF THE DV-HOP ALGORITHM AND ITS VARIATIONS 32

3.1 Introduction 32
3.2 Simulation Environment 32
3.3 Analysis of Shortcomings of the DV-Hop Algorithm 33
3.4 Evaluation Criteria for Localization Algorithms 37
3.4.1 Localization Accuracy 37
3.4.2 Communication Cost 38
3.4.3 Computational Cost 38
3.5 Behavior Analysis of Localization Accuracy 38
3.5.1 Simulation Scenarios 38
3.5.2 Analysis of Results 40
3.6 Behavior Analysis of the DV-Hop in Different Shaped Environments 41
3.6.1 Simulation Scenarios 41
3.6.2 Comparison of Performance 42
3.7 Influence of Anchor Placement 43
3.8 Grid based Sensor Networks 45
3.9 Effect of Radio Range Irregularity on DV-Hop 45
3.10 DV-Hop in 3D Environments 48
3.11 Performance Analysis of Selected Improved DV-Hop Algorithms . 50
3.11.1 An Improved DV-Hop Localization Algorithm for Wireless Sensor Networks (DV-Hop Average) 50
3.11.2 A New Improved DV-Hop Localization Algorithm (DV-Hop Marker) 52
3.11.3 A Weighted DV-Hop Localization Scheme for Wireless Sensor Networks (DV-Hop Weighted) 53

3.11.4 Improved DV-Hop Node Localization Algorithm in Wireless Sensor Networks (DV-Hop Improved) 55
3.11.5 Iterative Algorithm for Locating Nodes in WSN based on Modifying Average Hopping Distances (DV-Hop Iterative) 58
3.12 Comparison through Simulations 58
3.12.1 Simulation Environment 58
3.12.2 Discussion 59

4 NOVEL APPROACHES FOR IMPROVED DV-HOP LOCALIZATION 63
4.1 Proposed Algorithm 1 63
4.1.1 Overview 63
4.1.2 Recalculating the Anchor Positions (Step 2 of the Proposed Algorithm) 64
4.1.3 Iterative Hop Size Correction (Step 3 of the Proposed Algorithm) 66
4.2 Proposed Algorithm 2 67
4.2.1 Overview 67
4.2.2 Recalculating the Distance between the Anchors 68
4.3 Simulation Results and Discussion 69
4.3.1 Simulation Environment 69
4.4 Performance Evaluation 70
4.4.1 With variable number of Anchor Nodes 70
4.4.2 With variable Total number of Nodes 71
4.4.3 With variable Radio Range 71
4.4.4 Communication and Computational Cost 72
4.4.5 Conclusion 73
4.5 Proposed Algorithm 3 74
4.5.1 Overview 74
4.5.2 Simulation Results 76
4.5.3 Discussion 77
4.6 Localization in Emergency Environments 77
4.7 Introduction of New Nodes to the Network 78
4.7.1 Proposed Algorithm 79
4.7.2 Simulation Results 80
4.8 Introducing a New Node in an Emergency Situation 82
4.8.1 New Node outside the Fire 82

4.8.2 New Node inside the Fire 82
4.8.3 New Node inside a Spreading Fire 83
4.8.4 Discussion 83

5 TARGET TRACKING 86
5.1 Introduction 86
5.2 Target Tracking by Particle Filtering in Binary Sensor Networks . 86
5.2.1 Network Description 87
5.2.2 Mathematical Models 87
5.2.3 Tracking Algorithm 90
5.2.4 Simulation Results 92
5.3 A Novel Target Tracking Algorithm 96
5.3.1 Measurement Model 97
5.3.2 Target Tracking Algorithm 99
5.3.2.1 Noise Statistics Computation 99
5.3.2.2 Pre-localization using the DV-Hop algorithm . . 100
5.3.2.3 Kalman Filtering 102
5.3.3 Simulation Results 104
5.3.4 Performance Comparison 107
5.3.5 Proposed Tracking Algorithm in an Emergency Environment 110
5.3.6 Conclusion 112

6 NAVIGATION SUPPORT 113
6.1 Importance of Localization and Navigation 113
6.1.1 Current Practices 114
6.2 Related Work 114
6.3 Proposed Method to Emulate Lifeline Technique using WSNs . . . 116
6.3.1 Algorithm 116
6.3.2 Simulation Results 117
6.3.3 Conclusion 119

7 CONCLUSION 120
7.1 Research Summary 120
7.1.1 Introduction to Localization 120
7.1.2 Study of the DV-Hop Algorithm 120
7.1.3 Proposed Improved DV-Hop Algorithms for Improved Localization 121
7.1.4 Proposed Target Tracking Algorithm 121

7.1.5 Extending the DV-Hop Algorithm for Emergency Environments . . . 121
7.1.6 Proposed Navigation Support Technique for Fire Fighters . 122
7.2 Key Contributions . . . 122
7.3 Future Work . . . 124

List of Figures

1.1 Typical Deployment of a Wireless Sensor Network 1
1.2 Overall aspects of the research project 8
1.3 Thesis Organization in relation to Research Methodology 10

2.1 Classification of node localization algorithms 12
2.2 Time of Arrival 14
2.3 Time Difference of Arrival 14
2.4 Trilateration 15
2.5 Maximum Likelihood Estimation 16
2.6 Min-Max Estimation method 18
2.7 Example of DV-Hop Localization algorithm 22

3.1 Distribution of sensor nodes in the monitoring area 33
3.2 Effect of sensor node distribution in DV-Hop localization (Total Number of Nodes = 100, Radio Range = 50m, Simulations = 100 times) 34
3.3 Communication between an anchor and unknown nodes 35
3.4 The effect of link failures in localization accuracy of DV-Hop algorithm 36
3.5 Illustration of Radio Range Irregularity 37
3.6 (a) Error Percentage with different number of anchors (b) Error Variance with different number of anchors 39
3.7 (a) Error Percentage with different number of sensor nodes (b) Error Variance with different number of sensor nodes 39
3.8 (a) Error Percentage with different radio range (b) Error Variance with different radio range 39
3.9 DV-Hop on a square shaped area (Number of Nodes = 250, Anchors = 30) 41
3.10 DV-Hop on a L shaped area (Number of Nodes = 110, Anchors = 13) 41

3.11 DV-Hop on a U shaped area (Number of Nodes = 145, Anchors = 17) 41
3.12 DV-Hop on a rectangular shaped area (Number of Nodes = 75, Anchors = 9) 42
3.13 DV-Hop on a O shaped area (Number of Nodes = 196, Anchors = 24) 42
3.14 DV-Hop Localization Error in different shaped environments . . . 42
3.15 Influence of anchor placement on localization accuracy of DV-Hop algorithm (Number of Nodes = 200, Anchors = 20, Radio Range = 22m) 43
3.16 Influence of regular anchor placement (a)Anchors distributed randomly (b)Anchors placed on a grid (c)Anchors placed on diagonal (Number of Nodes = 200, Anchors = 20, Radio Range = 22m) . . 44
3.17 (a)Anchors are placed randomly inside the monitoring area, no anchors on the border (b)Anchors are placed only along the border (c)Potion of anchors are placed along the border and rest are randomly deployed inside the monitoring area (Number of Nodes = 200, Anchors = 20, Radio Range = 22m) 45
3.18 (a)25 nodes on a grid (Anchors = 5, Radio Range = 30m) (b)64 nodes on a grid (Anchors = 10, Radio Range = 20m) (c)100 nodes on a grid (Anchors = 20, Radio Range = 15m) 46
3.19 Radio Range Irregularity Model 47
3.20 (a)Localization error with different number of anchors (b)Localization error with different total number of nodes (c)Localization error with different radio ranges 47
3.21 Missing connections due to radio range irregularity 48
3.22 Sensor node distribution in a 3D environment 50
3.23 (a)Localization error with different number of anchors (b)Localization error variance with different number of anchors (Nodes = 200, Radio Range = 50m, Area = 100m×100m×100m) 50
3.24 (a)Error percentage with different number of anchors (b)Error variance with different number of anchors (N=200,R=22m) 59
3.25 (a)Error percentage with different total number of nodes (b)Error variance with different total number of nodes (AR=10%,R=22m) 59
3.26 (a)Error percentage with different radio range (b)Error variance with different radio range (N=200,A=20) 60
3.27 Time Ratio with different number of anchors 60

4.1 Algorithm 1 64
4.2 Algorithm 2 68
4.3 Actual and Estimated locations of anchors 69
4.4 Optimum Hop Size Correction for AHS computed in initial step . 69
4.5 (a)Error percentage with different number of anchors (b)Error variance with different number of anchors (N=200,R=22m) 71
4.6 (a)Error percentage with different total number of nodes (b)Error variance with different total number of nodes (AR=10%,R=22m) 72
4.7 (a)Error percentage with different radio range (b)Error variance with different radio range (N=200,A=20) 72
4.8 Time Ratio with different number of anchors 73
4.9 Sensors deployed on a grid in a random manner 75
4.10 Grid based DV-Hop algorithm 75
4.11 (a)25 nodes on a grid (b)64 nodes on a grid (c)100 nodes on a grid 77
4.12 (a)64 nodes on a grid (b)DV-Hop algorithm (c)Nearest grid point DV-Hop algorithm (d)Proposed DV-Hop based algorithm (e)Localization error of each node when Nearest grid point DV-Hop algorithm used (f)Localization error of each node when proposed DV-Hop based algorithm used 78
4.13 Performance of the proposed grid based DV-Hop algorithm in different node distributions 79
4.14 Localizing a newly added node 80
4.15 Localizing a newly added node using DV-Hop algorithm (Nodes = 200, Anchors = 20, Radio Range = 22m) 81
4.16 (a)Localization error with different number of anchors (b)Localization error with different total number of nodes (c)Localization error with different radio range 81
4.17 Localizing a new node in a node destroying environment (Nodes = 200, Radio Range = 22m) 82
4.18 Localizing a new node in a node destroying environment (Nodes = 200, Radio Range = 22m) 83
4.19 Localizing a new node inside a spreading fire (Nodes = 200, Anchors = 20, Radio Range = 22m, New node = (30,20)) 84

5.1 A binary sensor with a moving target 87

5.2 A realization of a target trajectory (Nodes = 121, Radio Range = 22m) (a)Deterministically deployed sensor network (b)Randomly deployed sensor network 93
5.3 RMSEs of the location estimates of the target (Nodes = 121, Radio Range = 22m) 94
5.4 RMSEs of location coordinates and velocity components (Nodes = 121, Radio Range = 22m) 94
5.5 CDF of RMSEs (Nodes = 121, Radio Range = 22m) 95
5.6 (a)RMSE with different total number of nodes (b)RMSE with different number of particles 95
5.7 Localizing a newly added node 97
5.8 Novel tracking algorithm 97
5.9 A realization of a target trajectory (Nodes = 200, Radio Range = 22m) 105
5.10 CDF of RMSEs (Nodes = 200, Radio Range = 22m) 105
5.11 (a)RMSE with different number of anchors (b)RMSE with different total number of nodes (c)RMSE with different radio ranges 106
5.12 Comparison of target tracking (Nodes = 200, Radio Range = 22m) 107
5.13 Average positioning error with different number of nodes (Nodes = 200, Radio Range = 22m) 108
5.14 Computational cost comparison (Anchor Ratio = 10%, Radio Range = 22m) 109
5.15 Average positioning error with different number of nodes (Nodes = 200, Radio Range = 22m) 110
5.16 Target tracking in a fire situation (Initial configuration: Anchors = 20, Nodes = 200, Radio Range = 22m) 111
5.17 RMSE in position with different radius of fire (Initial configuration: Anchors = 20, Nodes = 200, Radio Range = 22m) 111

6.1 Lifeline for fire fighters (taken form Shropshire Fire and Rescue Service - Brigade Order 7) 115
6.2 Incoming team follows the lifeline (taken form Shropshire Fire and Rescue Service - Brigade Order 7) 115
6.3 Localizing a newly added node 117
6.4 Spreading fire in an indoor environment 118
6.5 Temperature Distribution 118
6.6 Laying new sensors to emulate the lifeline 119

List of Tables

3.1 Simulation Instances 33
3.2 Simulation Instances 38
3.3 Simulation Instances 59
3.4 Performance Comparison of selected algorithms relative to the DV-Hop algorithm 61

4.1 Simulation Instances 70
4.2 Performance Comparison of Algorithms 74
4.3 Hop Count Table 79
4.4 Hop Count Table 80
4.5 Localization accuracy of a new node in a node destroying environment 83
4.6 Localization accuracy of a new node in a spreading fire (New node at (30,20)) 84

5.1 Simulation Instances 106

7.1 Performance Comparison of novel algorithms relative to the DV-Hop algorithm 123
7.2 Estimation error comparison of novel tracking algorithm (Radio Range = $22m$) 123
7.3 Estimation time comparison of novel tracking algorithm (Radio Range = $22m$) 123
7.4 Estimation error comparison of novel tracking algorithm in an emergency environment (Radio Range = $22m$) 124

List of Abbreviations

Abbreviation	Description
WSN	Wireless Sensor Network
BS	Base Station
GPS	Global Positioning System
MDS	Multi-Dimensional Scaling
DV-Hop	Distance Vector Hop
TOA	Time of Arrival
TDoA	Time Difference of Arrival
RSS	Received Signal Strength
AOA	Angle of Arrival
APIT	Approximate Point in Triangulation
MSP	Multi-Sequence Positioning
RSSI	Received Signal Strength Indicator
CoG	Center of Gravity
EKF	Extended Kalman Filter
PDF	Probability Density Function
CDF	Cumulative Distribution Function
PSO	Particle Swarm Optimization
RMSE	Root Mean Square Error

Chapter 1

INTRODUCTION

1.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) is a large scale ad hoc network with a large number of sensor nodes distributed in a monitoring field [1]. Often these sensor networks are deployed in remote and inaccessible areas. To cope with this, most of the sensor nodes require not only a sensing component but also on-board processing, communication and storage capabilities [2]. On the other hand these sensor nodes are often light, small and cheap with low power transceivers and limited data processing capabilities. Most of the time these sensor node perceive the environment, process or fuse the collected data and communicate the same with the neighbors and a base station (BS). Figure 1.1 [2] shows such a WSN deployment.

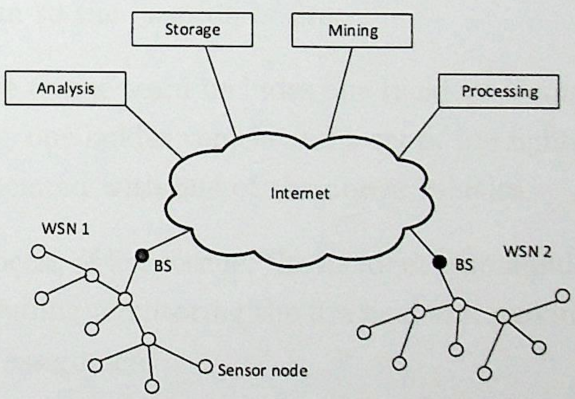


Figure 1.1: Typical Deployment of a Wireless Sensor Network

With the increase of the number of sensors in a network, sensors communicate the gathered data wirelessly to a central processing station. This is the case for most of the sensor networks deployed in remote and inaccessible areas. When many sensors cooperatively monitor large physical environments, sensor nodes communicate not only with each other but also with a base station. This allows sensor nodes to communicate the gathered data to remote processing, visualization, analysis and storage systems [2].

WSNs have been used in many practically important applications such as battle field surveillance, environmental monitoring, indoor user tracking, emergency rescue operations, pipeline (water, oil, gas) monitoring, structural health monitoring, precision agriculture, supply chain monitoring, active volcano monitoring and underground mining [2].

1.2 Emergency Situations and Application of WSNs

Natural and man-made disasters, such as fire situations, gas leak, earthquakes, floods and terrorist attacks, have reinforced the need for better emergency response solutions. WSNs can be applied in rescue operations in most of these emergency situations. In this thesis we use a typical scenario of a fire emergency for the illustration of our work.

1.2.1 Fire Rescue

Fire rescue operations are important and responsible activities that take place for public safety. A typical scenario of a fire rescue operation can be summarized as follows [3].

- When the fire department receives a fire alarm call, authorities will send a fire rescue team to the location of fire.
- Normally, a fire rescue team includes one Incident Commander Vehicle, two engine vehicles, one ladder vehicle and a set of fire fighters who are grouped as squads associated with one of the above vehicles.
- During the process of fire rescue, the incident commander is in charge of the operation, including monitoring the fire field and making real-time schedule on fire fighter assignment.
- The two engine vehicles carry water which will be used in case water is in short supply at the fire location.
- Ladder vehicles hold the utilities such as ladders that are needed for the fire fighters.
- The fire fighters are organized into different squads based on their specialty and fight cooperatively to extinguish the fire in the field.

There are a few shortcomings of the above mentioned procedure [3].

WSNs have been used in many practically important applications such as battle field surveillance, environmental monitoring, indoor user tracking, emergency rescue operations, pipeline (water, oil, gas) monitoring, structural health monitoring, precision agriculture, supply chain monitoring, active volcano monitoring and underground mining [2].

1.2 Emergency Situations and Application of WSNs

Natural and man-made disasters, such as fire situations, gas leak, earthquakes, floods and terrorist attacks, have reinforced the need for better emergency response solutions. WSNs can be applied in rescue operations in most of these emergency situations. In this thesis we use a typical scenario of a fire emergency for the illustration of our work.

1.2.1 Fire Rescue

Fire rescue operations are important and responsible activities that take place for public safety. A typical scenario of a fire rescue operation can be summarized as follows [3].

- When the fire department receives a fire alarm call, authorities will send a fire rescue team to the location of fire.
- Normally, a fire rescue team includes one Incident Commander Vehicle, two engine vehicles, one ladder vehicle and a set of fire fighters who are grouped as squads associated with one of the above vehicles.
- During the process of fire rescue, the incident commander is in charge of the operation, including monitoring the fire field and making real-time schedule on fire fighter assignment.
- The two engine vehicles carry water which will be used in case water is in short supply at the fire location.
- Ladder vehicles hold the utilities such as ladders that are needed for the fire fighters.
- The fire fighters are organized into different squads based on their specialty and fight cooperatively to extinguish the fire in the field.

There are a few shortcomings of the above mentioned procedure [3].

- The Incident Commander may not be able to gather a clear view of either the status of fire fighters after the rescue work begins or the accurate situation of the fire field, so that it is difficult for him to make an optimized schedule.

- Fire fighters in the field do not know the danger of the situation around them in advance, which increases the risk associated with the fire fighter.

- It is inflexible for the fire department headquarters located far away from the fire field to get fresh and timely fire rescue information, which is particularly important for big cities which might have multiple fires at the same time.

To address these shortcomings scholars have proposed some systems to support fire rescue operations [3]. WSNs have been widely studied in this context [3], [4].

1.2.2 Requirements for Systems to Support Fire Rescue Operations

Since WSNs are application specific systems, different applications have different specific requirements of the design of the underlying WSN. Using WSNs for fire rescue operations is one such special application with its own set of requirements as listed below [3].

Accountability of the authorities for the well-being of the fire fighters: The Incident Commander and fire department need the information of the fire fighters who are in the fire field to ensure their safety. Therefore the WSN based system should be able to collect the necessary information and report to the incident commander and other responsible authorities.

Real time monitoring: Real time information about the fire field is one of the key aspects of a supporting system for fire rescue operations. The Incident Commander needs the real time locations of the moving fire fighters so that he could make a better schedule. In addition, fire field information is also very useful for the incident commander to judge the real situation of the fire rescue and make correct real time decisions and schedule. Therefore WSN should be able to collect the environmental details of the fire field such as temperature, humidity, smoke and wind speed. Further, information about some vital events such as death of the fire fighters and dramatic changing of the environment parameters, should be captured by the WSN. Based on the above information, the incident commander and fire department will be able to have a clear view of the fire field and make better decisions.

Intelligent scheduling and resource allocation: Based on the gathered data by the WSN, the system should be able to provide support to the incident commander to do better resource allocation and scheduling.

Web-enabled service and integration Not only does the incident commander sitting near the fire field need the information collected by the WSN, but also the officers sitting in the fire department, which is located for away from the fire field. Web-based service is one of the most convenient ways to do this. Further, the collected data can be stored and analyzed later to find some good rescue models to support the future fire fights.

1.3 Basic Functions Required of a WSN to Support Rescue Operations

Before we apply a WSN in emergency rescue operations, we need to analyze the functions that the WSN should perform.

There is no significance of the collected data by the WSN, if the location of the sensor nodes are unknown since position is needed to locate events which occur inside the WSN. The same position information can be extended for target tracking, predicting the tracks of moving objects, assisting routing, managing the topology of the network and for location aware services [2]. Tracking moving objects is another very important functionality that a WSN should perform. Tracking functionality can be used to track and monitor the movement of fire fighters during an fire incident. Further, providing navigation support for fire fighters is another capability that a WSN should be capable of performing in emergency related applications. In short, WSN should be able to perform three basic functions as listed below.

- Localization of sensor nodes collecting information
- Tracking of moving nodes, usually worn by rescue workers
- Navigation Support for rescue workers

Many rescue support operations can be implemented upon these basic functions. Therefore in this thesis we examine the above three aspects in relation to emergency situations.

1.3.1 Localization

Localization is the task of determining the physical coordinates of sensor nodes or spatial relationships among them. Even though the Global Positioning System (GPS) is the most widely used localization technique, it is not accessible in all environments (indoor) and incur resource costs unacceptable compared with cheap wireless sensor nodes. On this basis, finding efficient and effective localization algorithms and techniques has become a hot and rising research area for many scholars.

The basis of most of the WSN localization techniques is to estimate the position of location unaware nodes (Unknown Nodes) relative to few location aware nodes (Anchors). Locations of anchors can be obtained by using GPS or placing the anchors at known points within the monitoring area [5].

A review of localization techniques relevant to this study is presented in section 2.2

1.3.2 Target Tracking

Target tracking is the functionality of tracing a moving object in a WSN. To be able to track the moving object at least three or more sensors in the network are required to sense the object simultaneously. A tracking system can be divided mainly into three subsystems, namely sensing subsystem, tracking subsystem and communication subsystem [6]. The sensing subsystem is used to sense the object. Tracking subsystem runs the prediction based algorithm which is used to trace the path of the moving object. Communication subsystem shares the collected data among nodes and fusion centers. Due to limited resources of sensors, tracking of a moving object has become a challenging problem in WSNs. Because of this challenging nature, such object tracking sensor networks are a spring of research opportunities.

The current state of the art in this area is reviewed in Section 2.6.

1.3.3 Navigation Support

Localization and navigation support is very useful in many day to day applications, but essential in emergency rescue operations. Teams must be able to reach safely and quickly if conditions become too dangerous and incident commanders must be able to keep track of their locations [7]. The simple task of getting out of a building becomes a challenge with little or no visibility due to smoke and power failure. High levels of mental and physical stress add to the difficulty: getting lost in a burning or collapsing building can have fatal consequences for both the rescue personnel and the building's occupants as oxygen supplies run out and medical attention is delayed [7]. It has been identified that 'lost inside' as a major cause of injuries to fire fighters. It is also reported that disorientation and failure to locate victims are contributing factors to fire fighter deaths [7]. Therefore providing navigation support for fire fighters in the fire field in a very important functionality.

A review of navigation support techniques relevant to this study is presented in Section 6.2.

1.4 Challenges for Localization, Tracking and Navigation Support in Emergency Situations

As described earlier, localization, tracking and navigation support are key tasks in WSN based rescue applications. Though, localization and tracking have been studied heavily in literature, few of practical localization and tracking algorithms are deployed in emergency environments such as fire situations due to multiple challenges. The conditions are significantly more demanding than non-emergency environments. Darkness, smoke, fire, power outages, water and noise can all prevent a system from working and heavy protective clothing, gloves and face masks make using a standard mobile computing impossible. Therefore localization, tracking and navigation algorithms for fire rescue operations should be able to deal with following challenges.

Accuracy: Incident Commander and the fire department do the scheduling based on the accurate locations of the sensors. Accurate localization in a highly dynamic environment is a challenging one.

Mobility and Tracking: Moving fire fighters should be able to locate and track in a efficient manner.

Node failures: Nodes in the monitoring field can be destroyed due to the fire and this can affect the performances of the localization algorithm. Failure of some sensors should not affect the location estimation of other sensors.

Link failures: More and more signal interferences can be taken place inside the sensor network so that temporary or permanent link failures can be taken place. The localization algorithm should be able to overcome these challenges.

Newly introduced nodes: Nodes can be introduced to the network randomly by the fire fighters to gain more information of a specific area. In this case, the localization algorithm should be able to locate the newly introduced nodes efficiently.

Radio range irregularity: Due to the harsh conditions of the environment, properties of the air medium can be changed dramatically. Therefore the communication rage of a sensor may not be an ideal sphere in 3D, but an irregular surface. The localization algorithm should be able to deal with the radio range irregularity.

Power efficiency: Each sensor node has a limited computing power, memory and communication range. Therefore, the localization algorithm should be able to provide an provide an accurate localization overcoming these limitations.

Feasibility: The localization algorithm has to be practical enough so that the its computing cost be affordable by the limited hardware and software supporting.

1.5 Scope

This research is a component of work carried out under NRC (National Research Council) Grant No. 11-113 entitled "Multipurpose Self-Configurable Indoor Wireless Sensor Network for Green Buildings" where the focus is to develop a complete suite of concepts and solutions for a specific application for WSNs in multi-story indoor environments. To achieve these main goals, the contribution of this work is on localization, target tracking and navigation support for emergency rescue operations. Figure 1.2 shows the overall aspects of the research project and the highlighted boxes show the contributions of this work.

1.6 Objectives

The overall objective of this research is to develop a suit of algorithms for localization, tracking and navigation of wireless sensor nodes in multi-story indoor

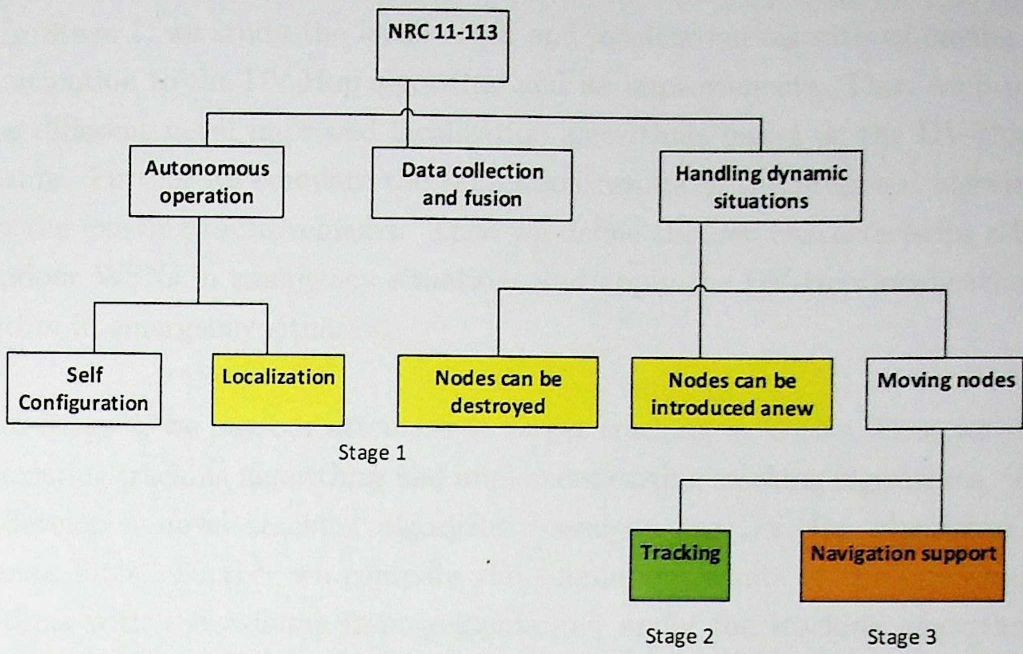


Figure 1.2: Overall aspects of the research project

environments in emergency situations.

Specific Objectives

- Characterizing an emergency environment in terms of WSN usage
- Developing a localization algorithm which is applicable to indoor environments where nodes can be destroyed, added anew and mobile nodes can be present
- Developing a tracking algorithm to track the mobile nodes in a sensor network
- Developing a navigation algorithm capable of supporting rescue operations in emergency situations

1.7 Methodology

Our main focus during this research is on how to develop a localization algorithm for WSNs, which is suited for emergency situations as characterized in Sections 1.2.2, 1.3 and 1.4. Further we develop a tracking algorithm and navigation support architecture to be used in emergency rescue applications based on WSNs. These tasks are carried out in three stages.

In Stage 1, we study the localization and localization algorithms paying special attention to the DV-Hop algorithm and its improvements. Then we propose three different novel improved localization algorithms based on the DV-Hop algorithm. Further we compare the simulation results of the proposed algorithms with the existing improvements. Then we define the key characteristics related to indoor WSNs in emergency situations and apply the DV-Hop localization algorithm in emergency situations.

In Stage 2, we pay our attention to target tracking in WSNs. First we study the existing tracking algorithms and implement exiting tracking algorithms. Then we develop a novel tracking algorithm based on the DV-Hop algorithm and Kalman filter. Further we compare the simulation results of the proposed algorithms with the existing improvements and apply the tracking algorithm in emergency situations.

In Stage 3, we pay the attention to navigation support for rescue works. First we study the existing navigation algorithms for emergency rescue operations. Then we develop a novel navigation support system for fire fighters. Further we simulate the algorithm in an emergency environment.

1.8 Organization of the Thesis

The thesis is organized of seven chapters. The relationship between the methodology and the thesis organization is illustrated in Figure 1.3.

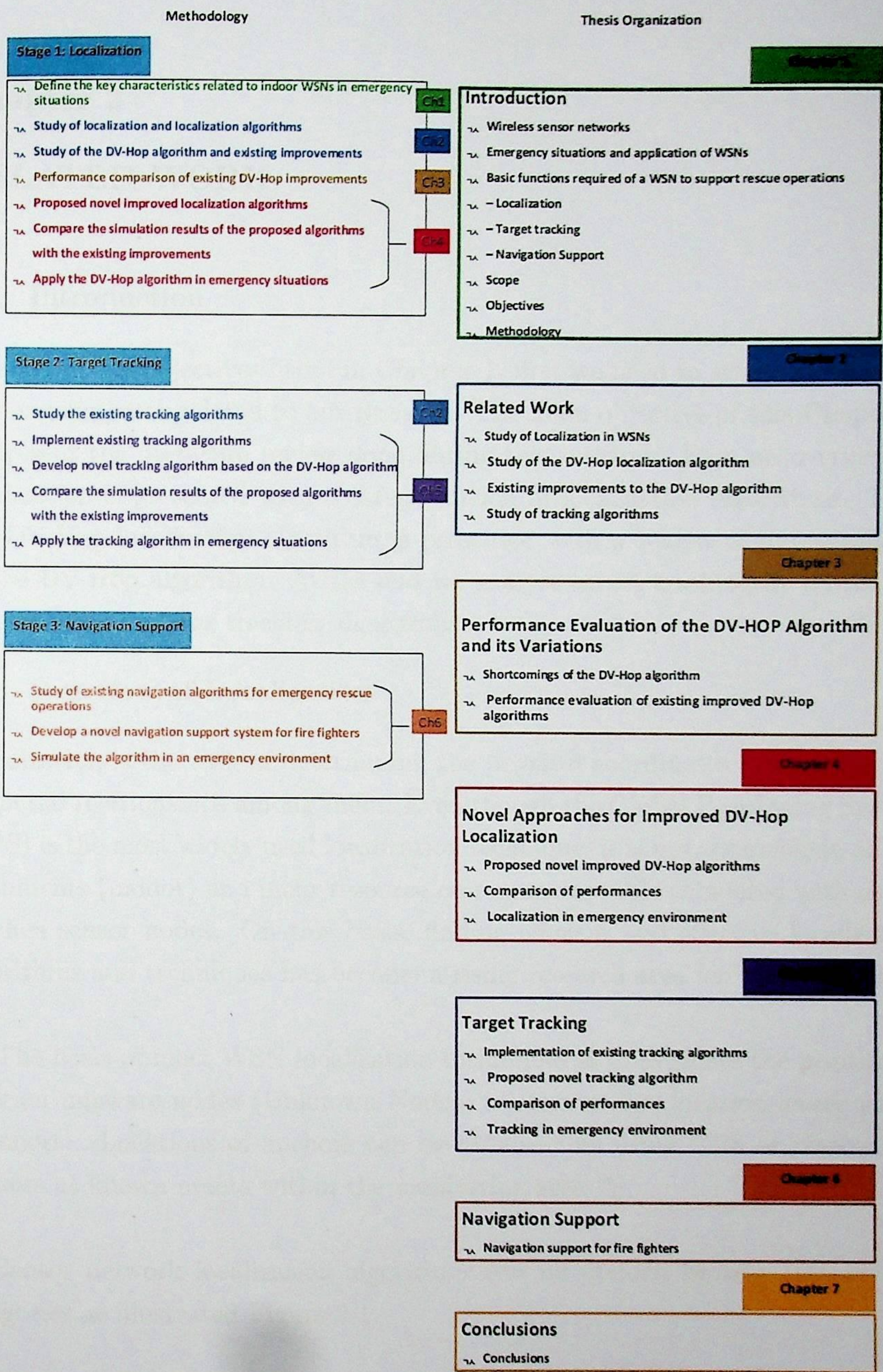


Figure 1.3: Thesis Organization in relation to Research Methodology

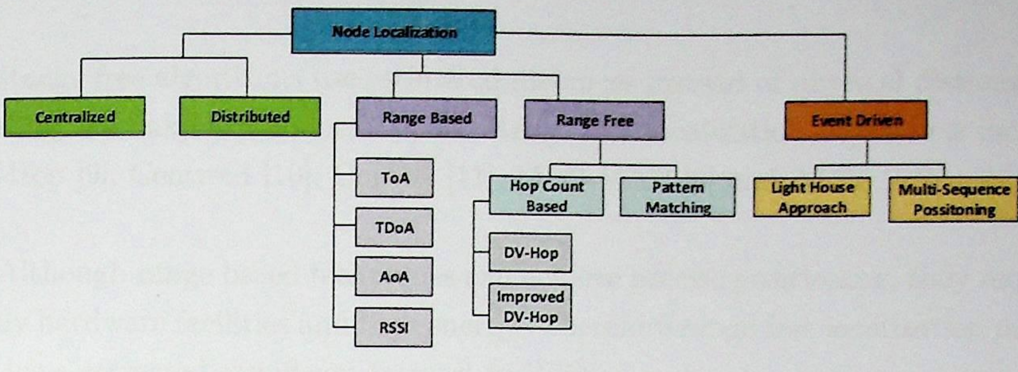


Figure 2.1: Classification of node localization algorithms

2.2.1 Centralized vs Distributed Algorithms

(Based on the approach of processing the individual inter-sensor measurement data)

In centralized algorithms each sensor node sends its inter-sensor measurement information to a single central processing location where positions of unknown nodes are calculated. Multi-Dimensional Scaling (MDS) localization algorithm [8] is a centralized one. In distributed algorithms, each unknown node estimates its own location inside itself by using inter-sensor information and information collected from neighbors. The DV-Hop [9] algorithm is a distributed one.

Distributed localization algorithms are generally considered to be more computationally efficient and easier to deploy in large scale networks. On the other hand, centralized algorithms generally provide more accurate localization estimates. Major drawbacks of centralized algorithms are the scalability and the requirement of higher computational complexity. Most of the distributed localization algorithms can be applied in centralized manner and distributed versions of centralized algorithms also can be designed for distributed applications [5].

2.2.2 Range Based vs Range Free Algorithms

(Depending on the mechanism used to measure inter-node distances)

Range based localization techniques exploit certain characteristics of signals exchanged among sensor nodes such as signal propagation times, signal strengths and angle of arrival. Thereafter, either trilateration or multilateration can be used to locate the unknown sensor nodes. Well known techniques are Time of Arrival (TOA), Time Difference of Arrival (TDOA), Received Signal Strength (RSS) and Angle of Arrival (AOA) [2].

Range free algorithms use estimated distances instead of physical distances in locating the unknown nodes. Typical Range free localization algorithms include DV-Hop [9], Centroid [10], Convex [11], MDS-MAP [8] and APIT [12].

Although range based techniques can achieve precise positioning, they require costly hardware facilities and high energy. Therefore range free localization methods have attracted significant interest in WSNs.

2.2.3 Event-Driven Localization

Event-Driven localization algorithms are based on events that occur in the monitoring area, which can be utilized to determine distances, angles, and positions. Such events can be the arrival of radio waves, beams of light, or acoustic signals at a sensor node [2]. Lighthouse location system of Romer [13], Multi-Sequence Positioning (MSP) approach of Zhong *et al.* [14], spotlight localization system of Stoleru *et al.* [15] and improved Asymmetric Event-Driven Localization Algorithm of Wang *et al.* [16] are some examples of even driven localization algorithms.

2.3 Node Self Localization

Accurate knowledge of node positions in a WSN is a key requirement in most of the applications. In these applications nodes need to estimate their own position by processing noisy range measurements relative to anchors. This is called node self localization.

2.3.1 Range Based Localization

Range based localization techniques exploit certain characteristics exchanged between sensor nodes signal propagation times, angle of arrival and signal strengths to measure the distance between sensor nodes [2]. Ranging techniques are used to take these measurements. Subsequently location estimation is carried out using ranging measurements obtained.

2.3.1.1 Ranging Techniques

- *Time of Arrival (ToA)*

Basis of ToA [2] is to measure the signal propagation time with a known velocity to determine the distance between two sensor nodes (transmitter and the receiver).

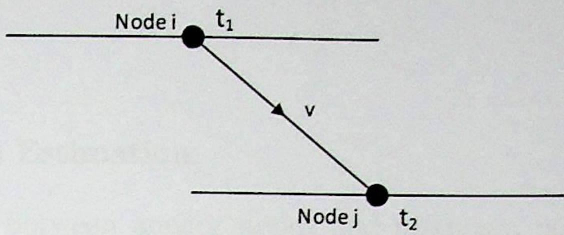


Figure 2.2: Time of Arrival

From Figure 2.2;

$$distance_{ij} = (t_2 - t_1) \times v \tag{2.1}$$

- *Time Difference of Arrival (TDoA)*

TDoA [2] uses two different signals with different velocities to determine the distance between sensor nodes.

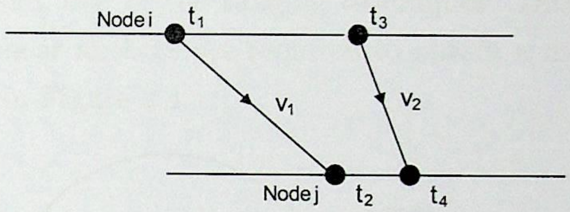


Figure 2.3: Time Difference of Arrival

From Figure 2.3;

$$distance_{ij} = (v_1 - v_2) \times (t_4 - t_2 - t_3 + t_1) \tag{2.2}$$

- *Angle of Arrival (AoA)*

AoA [2] uses an array of spatially separated antennas to discover the signal propagation direction and thereby determine the distance between sensor nodes. This techniques is bit expensive compared with others.

- *Received Signal Strength (RSS)*

RSS [2] uses the concept that signal strength deteriorates with the distance traveled. Almost all the sensor nodes have a Received Signal Strength Indicator (RSSI) which can be used to measure the amplitude of an incoming signal and in turn estimate the position of the sensor node. The Friis transmission equation expresses the ratio of the received power P_r to the transmission power P_t as 2.15.

$$\frac{P_r}{P_t} = G_t G_r \frac{\lambda^2}{(4\pi)^2 R^2} \tag{2.3}$$

Where G_t and G_r are antenna gains of transmitting and receiving antennas respectively.

2.3.1.2 Location Estimation

Once the distances between anchor nodes and unknown nodes are found using ranging techniques, it's time to determine the location or the position of the unknown nodes relative to the anchors. For this, either Triangulation, Trilateration / Multilaration or Min-Max algorithm can be used.

- *Trilateration*

The unknown node should be somewhere along the circumference of a circle centered at the anchor's position with a radius equal to the sensor - anchor distance which was found using one of the ranging techniques. Distance measurements from three non-collinear anchors are required to obtain a unique location in 2D space as illustrated in Figure 2.4.

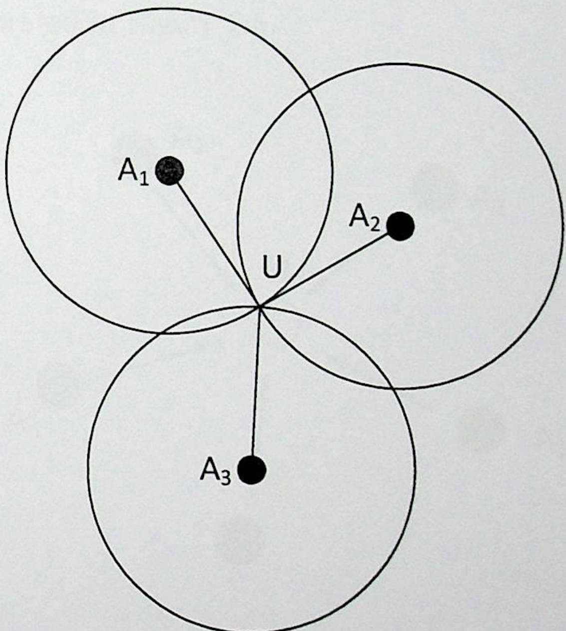


Figure 2.4: Trilateration

Let (x_u, y_u) be the location of the unknown node u and (x_j, y_j) be the known location of anchor node j. d_{uj} is the distance between them. Here $j = 1, 2, 3$ Then;

$$
\begin{aligned}
(x_u - x_1)^2 + (y_u - y_1)^2 &= d_{u1}^2 \\
(x_u - x_2)^2 + (y_u - y_2)^2 &= d_{u2}^2 \\
(x_u - x_3)^2 + (y_u - y_3)^2 &= d_{u3}^2
\end{aligned}
$$

Coordinates of the unknown node u can be calculated solving the above equations using the following matrix operation.

$$A = -2 \times \begin{bmatrix} x_1 - x_3 & y_1 - y_3 \\ x_2 - x_3 & y_2 - y_3 \end{bmatrix} \tag{2.4}$$

$$B = \begin{bmatrix} d_{u1}^2 - d_{u3}^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2 \\ d_{u2}^2 - d_{u3}^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \end{bmatrix} \tag{2.5}$$

$$P = \begin{bmatrix} x_u \\ y_u \end{bmatrix} = A^{-1}B \tag{2.6}$$

- ***Maximum Likelihood Estimation***

The unknown node should be somewhere along the circumference of a circle centered at the anchor's position with a radius equal to the sensor - anchor distance which was found using one of the ranging techniques. Distance measurements from at least three non-collinear anchors are required to obtain a unique location in 2D space as illustrated in Figure 2.5.

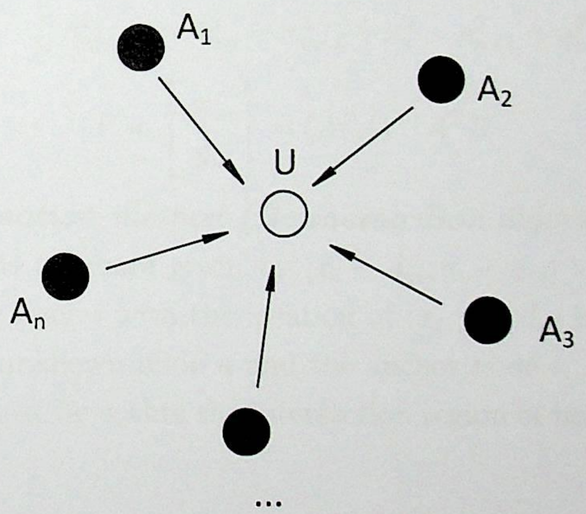


Figure 2.5: Maximum Likelihood Estimation

Let (x_u, y_u) be the location of the unknown node u and (x_j, y_j) be the known location of anchor node j. d_{uj} is the distance between them. Then;

$$
\begin{aligned}
(x_u - x_1)^2 + (y_u - y_1)^2 &= d_{u1}^2 \\
(x_u - x_2)^2 + (y_u - y_2)^2 &= d_{u2}^2 \\
&\vdots \\
(x_u - x_n)^2 + (y_u - y_n)^2 &= d_{un}^2
\end{aligned}
$$

where n is the number of anchors in the network.

Coordinates of the unknown node u can be calculated using the following matrix operation.

$$
A = -2 \times \begin{bmatrix} x_1 - x_n & y_1 - y_n \\ x_2 - x_n & y_2 - y_n \\ \vdots & \vdots \\ x_{n-1} - x_n & y_{n-1} - y_n \end{bmatrix} \tag{2.7}
$$

$$
B = \begin{bmatrix} d_{u1}^2 - d_{un}^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_{u2}^2 - d_{un}^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ d_{u(n-1)}^2 - d_{un}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix} \tag{2.8}
$$

$$
P = \begin{bmatrix} x_u \\ y_u \end{bmatrix} = (A^T A)^{-1} A^T B \tag{2.9}
$$

- ***Min-Max Estimation method (Bounding Box algorithm)***

Min-Max [17] builds a square given by $[x_i - d_{ui}, y_i - d_{ui}] \times [x_i + d_{ui}, y_i + d_{ui}]$ around each anchor node i with the location of (x_i, y_i). d_{ui} is the estimated distance between the unknown node u and the anchor node i. Estimated position of unknown node must be within the intersection region of boxes as illustrated in Figure 2.6.

Let number of anchors be N and the vertices of the intersection region be;

$$
V = (l, b), (r, b), (l, t), (r, t) \tag{2.10}
$$

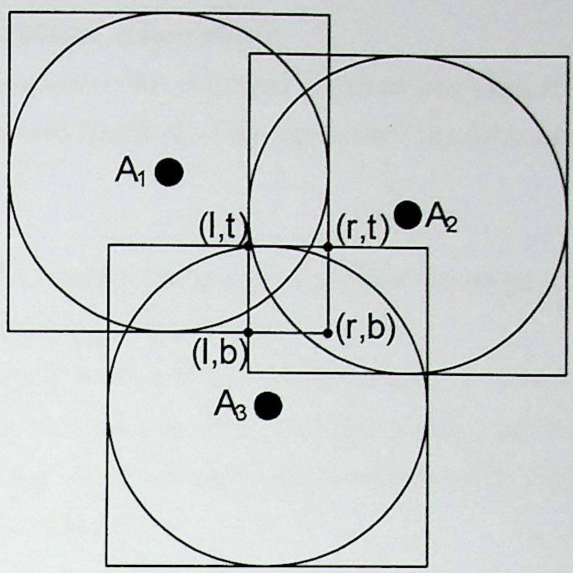


Figure 2.6: Min-Max Estimation method

where

$$l = \max_{i}^{N}(x_i - d_{ui}) \tag{2.11}$$
$$r = \max_{i}^{N}(x_i + d_{ui})$$
$$b = \max_{i}^{N}(y_i - d_{ui})$$
$$t = \max_{i}^{N}(y_i + d_{ui})$$

Then the estimated location of unknown node u is the center of the intersection area.

$$(x_u, y_u) = (\frac{l+r}{2}, \frac{b+t}{2}) \tag{2.12}$$

2.3.2 Range Free Localization

Range free algorithms use estimated distances based on connectivity information instead of physical distance or angle ranging measurements in locating the unknown nodes. Although range based techniques can achieve precise positioning, they require costly hardware facilities and high energy consumption. Therefore range free localization methods have attracted significant interest.

Typical Range free localization algorithms include DV-Hop [9], Centroid [10], Convex [11], MDS-MAP [8] and APIT [12].

- *Centroid Localization Algorithm*

Bulusu *et al.* [10] proposed the centroid localization algorithm which is a range free, connectivity-metric method. The algorithm implementation can be divided into three main steps.

Step 1: Each anchor node transmits a packet containing its location, to all the sensor nodes within its radio range.
Step 2: Each unknown node collects all the location information received from anchor nodes (which are one hop away) within a fixed period of time.
Step 3: Each unknown node calculates its position as the centroid of the received anchor positions given by 2.13

$$(x_u; y_u) = (\frac{\sum x_i}{n}, \frac{\sum y_i}{n}) \tag{2.13}$$

where $(x_u; y_u)$ is the estimated localization of the unknown node u, $(x_i, y_i)$is the location of anchor node i and n is the number of anchor nodes which are one hop away from the unknown node u.
The centroid localization algorithm is simple but the location accuracy is poor.

- *Convex Localization Algorithm*

Doherty *el at.* [11] proposed the convex position estimation which is based on connectivity-induced constraints. The main idea is to use the known locations of anchors to find the unknown locations of sensor nodes by applying proximity constraints imposed by known connections within a convex set. Provided that the network connectivity can be represented as a set of convex position constraints, the mathematical models can be used to generate feasible positions for the nodes in the network.

- *MDS- MAP Localization Algorithm*

MDS- MAP is a localization method based on multidimensional scaling (MDS) [8]. It uses connectivity information combined with known positions for certain anchor to estimate the locations of unknown nodes in the network. However, MDS-MAP is an inherently centralized algorithm and is therefore of limited utility in many applications.

- *APIT Localization Algorithm*

The basic idea of Approximate Point In Triangulation (APIT) localization algorithm [12] can be summarized as follows.

First, each anchor node transmits its location to all the unknown nodes within its radio range. The unknown node to be localized receives this information. Secondly, all the possible triangles are formed by connecting every three of these anchors. Thirdly, test whether the unknown node to be localized is within the triangle or not. This process should be carried out for all the triangles one by one. Finally, Center of Gravity (CoG) of the intersection of all the overlap triangles within which the unknown node resides, will be calculated. This CoG will be considered as the estimated location of the unknown node.

DV-Hop localization algorithm will be discussed in detail in the following Section.

2.4 The DV-Hop algorithm

The DV-Hop (Distance Vector Hop) Localization Algorithm [9] is a distributed, connectivity-based, and range free algorithm. Therefore the DV-Hop algorithm does not require high cost of hardware facilities and energy consumption required by range-based approaches. Although range based techniques can achieve precise positioning, accuracy of these techniques such as RSSI and ToA are purely dependent on the atmospheric conditions such as humidity of the environment. However the performances of the DV-Hop algorithm does not dependent on such external conditions. On the other hand the logic behind the DV-Hop algorithm is very simple so that sensor nodes do not need much energy to run the algorithm. Due to these plus points of the DV-Hop localization algorithm, we selected this as the core of our research.

In a sensor network there are two kinds of sensor nodes - anchors whose locations are known and unknown nodes. The basic idea behind the DV-Hop algorithm is to represent the distance between an unknown node and an anchor as a product of the average Hop Size and Hop Count. The basic DV-Hop Algorithm runs in three major steps.

Step 1: Each anchor node broadcasts a packet throughout the network containing its location and a Hop Count value initialized to one. Each receiving node keeps the minimum Hop Count while discarding higher ones from a particular anchor node. Each receiving node will increase the Hop Count by one before passing the packet onwards. At the end of this process, each node in the network obtains the minimum Hop Count to every anchor node.

Step 2: Each anchor node estimates its Hop Size using the Hop Count values to the other anchors. Hop Size is estimated by anchor node i as follows;

$$Hop\ Size_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} hop_{ij}} \tag{2.14}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of anchors i and j respectively and hop_{ij} is the Hop Count between anchors i and j.

Each anchor calculates its Hop Size and broadcasts it to the network. Unknown nodes save the first value they receive, while transmitting this Hop Size to the neighbors. This mechanism results in most of the unknown nodes obtaining the Hop Size of the nearest anchor. At the end of this process, unknown nodes estimate the distance form each anchor as the product of Hop Size and the corresponding Hop Count.

Step 3: Unknown nodes estimate their locations using either trilateration / multilateration or maximum likelihood estimation after they receive three or more distance information.

Let (x_u, y_u) be the location of the unknown node u and (x_j, y_j) be the known location of anchor node j. d_{uj} is the distance between them. Then;

$$\begin{gathered}
(x_u - x_1)^2 + (y_u - y_1)^2 = d_{u1}^2 \\
(x_u - x_2)^2 + (y_u - y_2)^2 = d_{u2}^2 \\
\vdots \\
(x_u - x_n)^2 + (y_u - y_n)^2 = d_{un}^2
\end{gathered}$$

where n is the number of anchors in the network.

Coordinates of the unknown node u can be calculated using the following matrix operation.

$$A = -2 \times \begin{bmatrix} x_1 - x_n & y_1 - y_n \\ x_2 - x_n & y_2 - y_n \\ \vdots & \vdots \\ x_{n-1} - x_n & y_{n-1} - y_n \end{bmatrix} \tag{2.15}$$

$$B = \begin{bmatrix} d_{u1}^2 - d_{un}^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_{u2}^2 - d_{un}^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ d_{u(n-1)}^2 - d_{un}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix} \tag{2.16}$$

$$P = \begin{bmatrix} x_u \\ y_u \end{bmatrix} = (A^T A)^{-1} A^T B \tag{2.17}$$

Figure 2.7 presents an example with three anchor nodes $A1$, $A2$ and $A3$. Anchor $A1$, knowing its Euclidean distances ($50m$ and $110m$) and Hop Counts (two hops and six hops) to the other two anchor nodes, computes a Hop Size of $(50 + 110)/(2 + 6) = 20$, which represents the estimated distance of a hop in meters. In a similar fashion, $A2$ computes a Hop Size of 18.6 and $A3$ computes a Hop Size of 17.3.

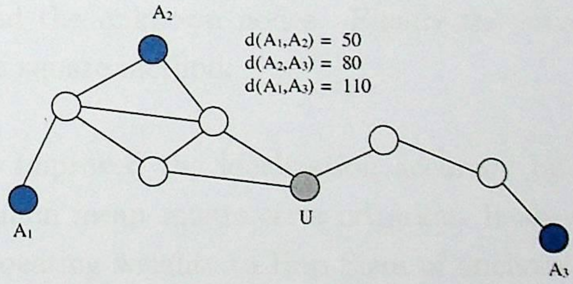


Figure 2.7: Example of DV-Hop Localization algorithm

Hop Sizes are propagated via controlled flooding to ensure that each unknown node will only use one Hop Size, typically from the closest anchor. For example, sensor node U in Figure 2.7 uses the Hop Size obtained from $A2$, that is, 18.6, to estimate its distances to the three anchors by multiplying the Hop Size with the Hop Counts (leading to distances 3×18.6 to $A1$, 2×18.6 to $A2$ and 3×18.6 to $A3$). Given these distances, triangulation can be used to determine the position of U.

2.5 Review of Improvements to the DV-Hop Localization Algorithm

Considering these facts, to improve the localization accuracy of DV-Hop Algorithm scholars have paid their attention in several aspects; modifying the Hop Size, deployment of anchors and other strategies.

2.5.1 Improvements using Modified Hop Size

• Wang *et al.* [18] extended the DV-Hop algorithm into 3D space and addressed its shortcomings to improve the performances. It modified the Hop Size and per-hop distance calculation to achieve improved localization accuracy.

• Chen *et al.* [19] proposed a method to improve the localization accuracy of DV-Hop algorithm by modifying the Hop Size using averages. . In addition it adopted the 2D hyperbolic localization scheme instead of traditional trilateration during the process of localization.

• Li *et al.* [20] used weight of anchors to improve the localization accuracy of DV-Hop algorithm. The weight of each anchor was a production of Hop Size weight and Distance weight. Here Hop Size weight reflected the accuracy of hop size provided by the anchors. The Distance weight reflected the distance between the anchor node and the unknown nodes. Finally the location was calculated using weighted least square method.

• Zhang *et al.* [21] improved the localization accuracy by modifying the Hop Size based on minimum mean square error criterion. It also suggested a further improvement by allocating weights to Hop Sizes of anchors in the reserve order of Hop Count between unknown nodes and anchors.

• Lee *et al.* [22] improved the accuracy of DV-Hop algorithm by introducing a weight which inversely proportional to the differences of Hop Counts. The weight also reflected the characteristic of the network.

• Hu *et al.* [23] introduced a threshold M, it used the weighted average Hop Size of anchor nodes within M hops to calculate the average Hop Size of unknown nodes. In order to further improve the algorithm, it estimated the location of anchors considering as unknown nodes using the estimated locations of unknown nodes which acted as anchors. Finally positioning results were corrected using that correction factor.

• Bao *et al.* [1] improved the localization accuracy of the DV-Hop algorithm by modifying the Hop Size of anchors with an error correction value. Then the Hop Sizes used by unknown nodes were modified with a weight.

• Xiang *et al.* [24] used residual weighting algorithm to estimate the position of unknown nodes by improving the calculating mode of average hop distance. In this process, for the position estimation with high residual error, a relatively low weight was assigned.

• Lin *et al.* [25] introduced an improved DV-Hop algorithm using three main approaches. Firstly, Hop Size among anchor nodes were modified using least squares method. Secondly, Hop Size used by unknown nodes were modified by weighting the N received Hop Sizes from anchor nodes. Finally, the iterative numerical method with the initial values of the estimated node location was presented by setting proper threshold.

• Chen *et al.* [26] presented an improved DV-Hop algorithm with four major steps. Firstly, some anchor nodes were placed at the border of monitoring area. Secondly, Hop Size between anchor nodes was modified, and the average Hop Size used by each unknown node for estimating its location was modified through weighting the received average one-hop distances from anchor nodes. Finally, the particle swarm optimization was used to correct the position estimated by the 2D hyperbolic localization algorithm, which made the result closer to the actual position.

2.5.2 Improvements using Controlled Deployment of Anchor Nodes

• Yin *et al.* [27] focused on how the placement and quantity of anchor nodes effect on localization error and suggested an anchor node placement strategy to improve the localization accuracy of DV-Hop algorithm.

• Zheng *et al.* [28] proposed an anchor node deployment strategy in order to reduce the localization error of DV-Hop algorithm. First, an anchor node was placed in the middle of square area and the other nodes are equally placed on the circle whose center was the center of square area and radius was half of the side length. That placement strategy was implemented with long range DV-Hop algorithm.

2.5.3 Improvements combining RSSI with the DV-Hop Algorithm

• Fang *et al.* [29] combined RSSI ranging technology with the basic DV-Hop algorithm to improve its performance.

• Ding *et al.* [30] proposed an RSSI based Hop Count calculation method, that could accurately reflect the distance between adjacent nodes and thereby the location error could be reduced than the DV-Hop algorithm.

• Yi *et al.* [31] improved the DV-Hop localization accuracy based on a regular moving anchor node and RSSI ranging technique. In the proposed RMADV-Hop algorithm, the moving anchor node virtually created an even distribution of anchors.

• Tian *et al.* [32] incorporated RSSI with DV-Hop to propose a new algorithm called RDV-Hop. The idea was to improve the localization accuracy of DV-Hop algorithm by combining the advantages of range based techniques to minimize the disadvantages of range free algorithms.

• Wang *et al.* [33] improved the localization accuracy of DV-Hop algorithm by incorporating the advantages of RSSI method with DV-Hop algorithm. A set of newly upgraded anchors will be generated using the original set of anchors with the help of RSSI method. On the other hand a pre-set priority level was given to each anchor where priority level for newly upgraded anchors were lower than that of original anchors. Anchors with higher priority had the highest precision. Finally, trilateration or minimum least square method was used to find the locations of unknown nodes relative to the selected anchors with highest priority.

2.5.4 Improvements using Other Techniques

• Kumar *et al.* [34] adopted a weighted least square algorithm in solving n equations for location estimation of unknown nodes using n anchors, which replaced the third step of DV-Hop algorithm. Further improvements was achieved by using extraneous information by the equations.

• Wang *et al.* [35] used location known regulated nodes or markers to rectify the location error of unknown nodes by generating an error vector and thereby improving the localization accuracy of DV-Hop algorithm. The regulated nodes could be placed either randomly or manually.

- Chen *et al.* [36] proposed an improved DV-Hop algorithm. The main principle of the improved scheme was estimation distance of the hops according to the number of neighbors in the same block. In order to reduce the localization error, it used weighted node distances to calculate the node's final coordinate.

- Xu *et al.* [37] proposed and improved DV-hop algorithm, especially focusing on non-uniform node distribution. The upper bound of the position area was determined and suitable anchors participating location were selected among the local position area. The Average Distance Per-Hop (ADPH) could be calculated using the suitable anchors in the local area, which was rarely affected by the non-homogeneous density.

- Zhou *et al.* [38] proposed NDV-Hop algorithm as an improved DV-Hop algorithm. Firstly, average hop distance was calculated by taking all the anchor nodes into account. Secondly, the position information of unknown nodes was amended by adopting error value of the anchor nodes. Finally, the position information that the out-of-scope node was revised by taking boundary coordinates as its potion information.

- Quanrui *et al.* [39] proposed a new liner approximation model for the relationship between Hop Count and the estimated distance based on the DV-Hop algorithm. Based on the new assumptions, Hop Size calculation was changed and estimated distance was calculated differently depending on the fact that communication range was known or not.

- Gao *et al.* [40] proposed the CDDV-Hop algorithm which combined the close degree model with basic DV-Hop algorithm to improve the localization accuracy.

- Liu *et al.* [41] proposed the VAH-DV-Hop algorithm to minimize the localization error of DV-Hop algorithm. The algorithm made the hops (from the unknown nodes to the anchor) multiplied by the average hop distance belonged to the corresponding anchor instead of getting the multiplication between hops (from the unknown nodes to the anchor) and average hop distance of the nearest anchor for the unknown node.

• Gui *et al.* [42] proposed a Selective 3 Anchor DV-Hop algorithm with the following idea. The unknown node first selected any three anchors to form a 3-anchor group. Then it calculated the candidate positions based on each 3-anchor group and finally according to the relation between candidate positions and minimum hop counts to anchors, the unknown node chose the best candidate position.

• Agashe *et al.* [43] studied the performance of DV-Hop algorithm in irregular shaped areas and worked out for localization accuracy for the same. In the proposed Optimum DV-Hop algorithm, a selection criteria for anchors in the process of locating the unknown nodes was proposed.

• Yan *et al.* [44] proposed a novel approach that used principal component analysis in order to eliminate the impact of multicollinearity and noise while reducing its localization error. Principal component analysis method is effective in rectify the defects in least square estimation caused by collinearity and noise.

• Li *et al.* [45] divided the network into small regions using the Voronoi diagram which prevented broadcast storm and reduced the overall network traffic. After that, under the constraints such as number of hops and the distance from the unknown node to anchors, the grid scanning method and iteration operation were used locate the unknown nodes.

• Mao *et al.* [46] proposed and improved DV-Hop algorithm called Area Division based semi-auto DV-Hop localization algorithm (ADBSA DV-Hop). Several ideas were employed to improve the localization accuracy such as semi-auto average size of per hop, area division and sticking to border.

2.6 Overview of Target Tracking in WSNs

Target tracking is the function of tracking a moving object in a WSN. To be able to track the moving object, at least three or more sensors in the network required to sense the object simultaneously. A tracking system can be divided mainly into three subsystems, namely sensing subsystem, tracking subsystem and communication subsystem [6]. The sensing subsystem is used to sense the object. The tracking subsystem runs the prediction based algorithm which is used to trace the path of the moving object. The communication subsystem shares the collected data among nodes and fusion centers.

Due to limited resources of sensors, tracking of a moving object has become a challenging problem in WSNs. Because of this challenging nature, such object tracking sensor networks are a spring of research opportunities.

In the first part of this Section we study few of those such research efforts on target tacking in WSNs.

2.6.1 Related Work

Many scholars have proposed different target tracking algorithms considering and prioritizing different aspects such as energy conservation and optimization, accuracy and communicational cost. Few of those algorithms are listed below.

• Tsai *et al.* [47] proposed a protocol to track a mobile object in a sensor network. The work was concentrated on how to query target tracks and obtain the target position effectively. The mobile user can obtain the tracking object position without broadcast query.

• Zhao *et al.* [48] proposed a novel weighted distance based node selection method for bearings-only sensors in WSNs. Based on the probability distribution function of predicted target state and the bearing error of a sensor itself, the sensor with minimum weighted distance was activated in the tracking process.

• Oh [49] developed a scalable real-time multi-target tracking algorithm that is autonomous and robust against transmission failures, communication delays and sensor localization error. The algorithm was based on a rigorous probabilistic model and an approximation scheme for the optimal Bayesian filter.

• Zhou *et al.* [50] considered the problem of collaborative target tracking in WSNs in a framework of quantized measurement fusion. First, the measurement in each local sensor was quantized by probabilistic quantization scheme and transmitted to a fusion center. Then, the quantized messages are fused and sequential importance resampling particle filtering was employed to estimate the target state.

• Yeow *et al.* [51] looked into efficient energy utilization of a target tracking sensor network by prediction a target's trajectory through experience. While this was not new, the chief novelty came in conserving energy through both dynamic spatial and temporal management of sensors while assuming minimal locality information.

• Teng *et al.* [52] proposed an algorithm that exploited measurements to simultaneously localize the detecting sensors and track the target. A general state evolution model was employed to describe the dynamical system with neither proper knowledge of the target moving manner nor precise location information of the sensors. The joint posterior distribution of the parameters of interest was updated online by incorporating the incomplete and inaccurate measurements between the target and each of the sensors into a Bayesian filtering framework. A variational approach was adopted in the framework to approximate the filtering distribution, thus minimizing the inter-cluster communication and the error propagation.

• Wang *et al.* [53] proposed a new tracking framework, called FaceTrack, which employed the nodes of a spatial region surrounding a target, called a face. Instead of predicting the target location separately in a face, authors estimated the target's moving toward another face. An edge detection algorithm was proposed to generate each face further in such a way that the nodes can prepare ahead of the target's moving.

• Lin *et al.* [54] proposed an adaptive energy efficient multisensory scheduling scheme for collaborative target tracking in WSNs. It calculated the optimal sampling interval to satisfy a specification of predicted tracking accuracy, selected the cluster of tasking sensors according to their joint detection probability and designated one of the tasking sensors as the cluster head for estimation update and sensor scheduling according to a cluster head for estimation update and sensor scheduling according to a cluster head energy measure function.

• Xu *et al.* [55] studied the problem of tracking signal emitting mobile targets using navigated mobile sensors based on signal reception. The mobile sensor controller utilizes the measurement collected by a WSN in terms of the mobile target signal's time of arrival (TOA). The mobile sensor controller acquired the TOA measurement information from both the mobile target and the mobile sensor for estimating their locations before directing the mobile sensor's movement to follow the target.

• Wang *et al.* [56] proposed a distributed energy optimization method for target tracking applications. Sensor nodes were clustered by maximum entropy clustering. Then the sensing field was divided for parallel sensor deployment optimization. For each cluster, the coverage and energy metrics were calculated by grid exclusion algorithm and Dijkstra's algorithm, respectively. Cluster heads perform parallel particle swarm optimization to maximize the coverage metric and minimize the energy metric.

• Masazade *et al.* [57] studied the problem target tracking based on energy reading of sensors. Authors minimized the estimation error by using an Extended Kalman Filter (EKF). The Kalman gain matrix is obtained as the solution to an optimization problem in which a sparsity promoting penalty function was added to the objective. By using a sparse Kalman gain matrix only a few sensors send their measurements to the fusion center, thereby saving energy.

• Oka *et al.* [58] proposed a novel algorithm for target tracking using signal strength measurements by a WSN in a power efficient manner. First, authors proposed a tandem incremental estimator that learns and tracks the radio environment of the network and provided that knowledge for the use of the tracking algorithm. Secondly, authors reduced the unbiased tracking error by exploiting the co-dependencies in the motion of several targets via a fully distributed and tractable particle filter.

• Wang *et al.* [59] presented an energy efficient selection of cooperative nodes. In the proposed method, the target detection probability was estimated by single node processing based on particle filter. Then an objective function for collaborative target tracking in WSNS was constructed according to the information utility and remaining energy of sensor nodes.

• Wang *et al.* [60] presented a new target tracking approach which avoids the instability problem and offered superior tracking performances. Authors first proposed an improved noise model which incorporates both additive noises and multiplicative noises in distance sensing. Then they use a maximum likelihood estimator for pre-localization to remove the sensing nonlinearity before applying a standard Kalman filter.

- Djuric *et al.* [61] presented particle filtering algorithms for tracking a single target using data from binary sensors. Authors proposed and implemented the tracking by employing auxiliary particle filtering and cost reference particle filtering. Further, authors extended the method to include estimation of constant parameters and derived the posterior Cramer-Rao bounds for the states.

The main objective of this literature review on tracking algorithms is to find a suitable method to use with DV-Hop algorithm to produce a novel tracking algorithm in WSNs. Though there are so many techniques presented for tracking a target in a WSN, most of them are centralized and dependent on range-based techniques to run the algorithm. Therefore concepts in the tracking algorithms based on these characteristics can not be incorporated with the DV-Hop algorithm. Ultimately we picked pre-localization technique presented in [60] to combine with the DV-Hop algorithm.

We present a novel tracking algorithm which uses some of the concepts in [60] in Chapter 5 and we implement and compare the results with [61].

Chapter 3

PERFORMANCE EVALUATION OF THE DV-HOP ALGORITHM AND ITS VARIATIONS

3.1 Introduction

This Chapter presents a comprehensive study of the DV-Hop localization algorithm. In Chapter 2 we carried out a detailed study of different localization techniques. Of these, we use the DV-Hop algorithm as the core localization technique for this research. The DV-Hop is a range free localization algorithm, and therefore does not need expensive hardware and high energy consumption. Further, it is a distributed algorithm so that the network does not need a central component to estimate the locations of senor nodes. The logic behind the DV-Hop algorithm is also a simple one which results in a low computational cost. Considering these facts, the DV-Hop algorithm is a good candidate for our application environment.

We introduced the DV-Hop algorithm and its existing improvements in Chapter 2. In this Chapter we analyze behavior of the DV-Hop algorithm and its improvements in depth. First we analyze the shortcoming of the DV-Hop algorithm and then the localization accuracy of the same with the variation of anchor nodes, total number of nodes and radio range. The behavior of the DV-Hop algorithm in different shaped environments is studied followed by a study in 3D environments. Finally we compare the performances of a few selected localization algorithms in literature. Our objective is the evaluation of these algorithms in a common set of comparable scenarios, which is not available in literature.

3.2 Simulation Environment

To evaluate the performance of the DV-Hop algorithm and its improvements, Matlab based simulations were used. The base scenario for simulation is one where both anchors and unknown nodes are uniformly distributed in a $100m \times 100m$ area, with each node having a radio range of $22m$. In literature most of the researchers have used a simulation environment such as this to run and simulate their algorithms [9], [26] and [20]. This is illustrated in Figure 3.1.

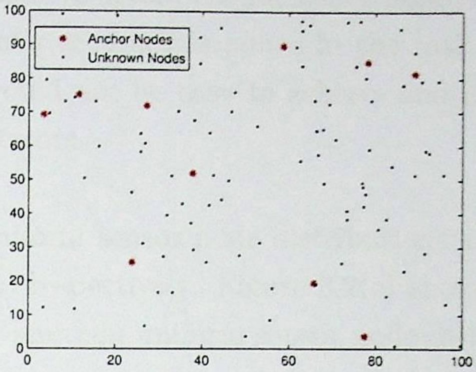


Figure 3.1: Distribution of sensor nodes in the monitoring area

Localization error and localization error variance are used to evaluate the accuracy and the stability of the network. In most of the simulations, a scenario of 200 nodes, each having a radio range of $22m$ in a $100m \times 100m$ area is used. Then, the DV-Hop algorithm is studied for its performance by varying one at a time, the anchor ratio (AR), the total number of nodes (N) and the radio range (R) of the nodes as summarized in Table 5.1. All the simulation results are averaged over 100 runs.

Table 3.1: Simulation Instances

Figure No	Total Number of Nodes (N)	Anchor Ratio(%) (AR)	Radio Range(m) (R)
3.6(a),(b)	200	variable 5-40	22
3.7(a),(b)	variable 100-400	10	22
3.8(a),(b)	200	10	variable 15-40

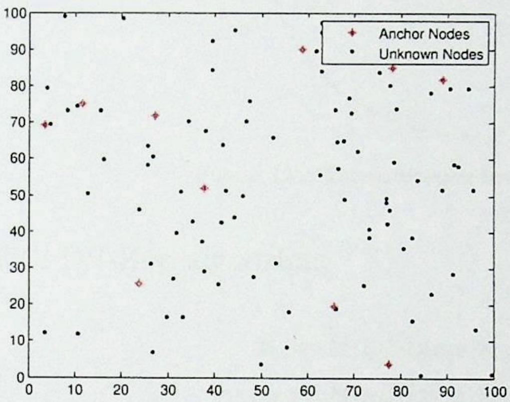
3.3 Analysis of Shortcomings of the DV-Hop Algorithm

Even though the DV-Hop algorithm is an attractive option for the localization of nodes in a wireless sensor network due to its simplicity, it suffers from poor accuracy. This is due to several fundamental reasons. This Section will analyze these facts in depth.

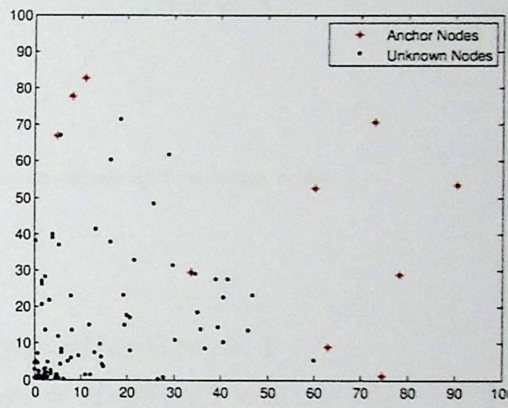
- *Estimation error of Average Hop Size:*

The accuracy of the DV-Hop localization algorithm depends on the distribution of sensor nodes. If the node distribution is such that the inter-node distances are nearly equal, the estimated average Hop Size of the network will be accurate resulting in a low localization error. On the other hand, if the node distribution is uneven, the algorithm's accuracy is poor. In the real environment, this even distribution of nodes would not be easy to achieve and hence may result in unacceptable positioning errors.

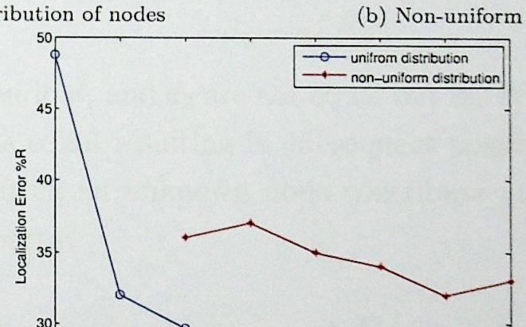
Uniform and non-uniform sensor node distributions are illustrated in Figure 3.2(a) and Figure 3.2(b) respectively. Figure 3.2(c) shows the localization error comparison for uniform and non-uniform sensor node distributions.



(a) Uniform distribution of nodes



(b) Non-uniform distribution of nodes



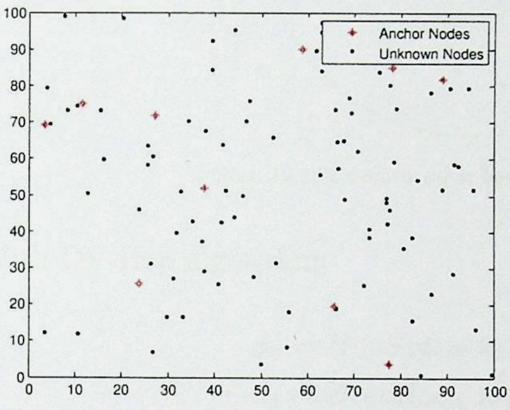
(c) Localization error with different node distributions

Figure 3.2: Effect of sensor node distribution in DV-Hop localization (Total Number of Nodes = 100, Radio Range = 50m, Simulations = 100 times)

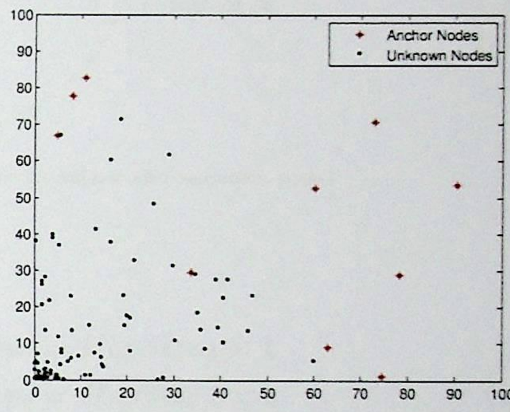
It is evident that localization error of the DV-Hop algorithm in a uniform sensor distribution is significantly less than that of a non-uniform distribution of sensors. In addition to average hop size estimation error, there may be connectivity gaps in the sensing area due to the non-uniformity of the sensor distribution, which leads to further errors and some nodes not being localized.

of sensor nodes. If the node distribution is such that the inter-node distances are nearly equal, the estimated average Hop Size of the network will be accurate resulting in a low localization error. On the other hand, if the node distribution is uneven, the algorithm's accuracy is poor. In the real environment, this even distribution of nodes would not be easy to achieve and hence may result in unacceptable positioning errors.

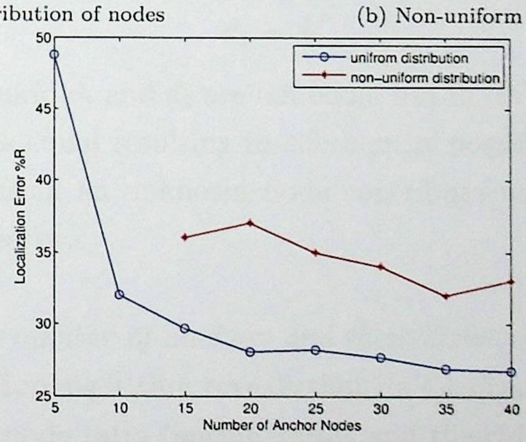
Uniform and non-uniform sensor node distributions are illustrated in Figure 3.2(a) and Figure 3.2(b) respectively. Figure 3.2(c) shows the localization error comparison for uniform and non-uniform sensor node distributions.



(a) Uniform distribution of nodes



(b) Non-uniform distribution of nodes



(c) Localization error with different node distributions

Figure 3.2: Effect of sensor node distribution in DV-Hop localization (Total Number of Nodes = 100, Radio Range = 50m, Simulations = 100 times)

It is evident that localization error of the DV-Hop algorithm in a uniform sensor distribution is significantly less than that of a non-uniform distribution of sensors. In addition to average hop size estimation error, there may be connectivity gaps in the sensing area due to the non-uniformity of the sensor distribution, which leads to further errors and some nodes not being localized.

- *Estimation error of per-hop distance:*

Imagine the scenario illustrated in Figure 3.3, with A being the nearest anchor for unknown nodes U_1 and U_2 which are within the communication range of A.

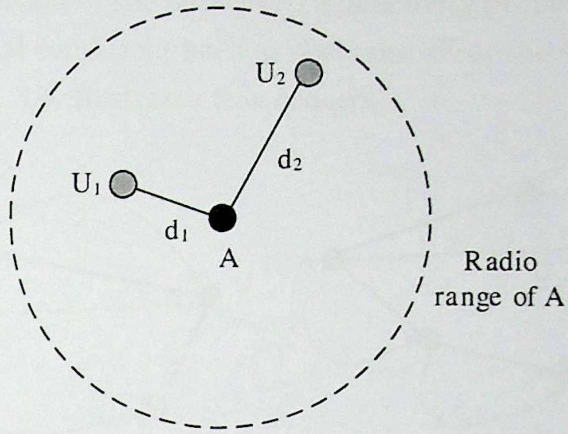


Figure 3.3: Communication between an anchor and unknown nodes

With DV-Hop algorithm;

$$\begin{aligned} d_1 &= HopSize_A \times Hops = HopSize_A \times 1 \\ d_2 &= HopSize_A \times Hops = HopSize_A \times 1 \\ &d_1 = d_2 \end{aligned} \tag{3.1}$$

In the actual scenario d_1 and d_2 are not equal but in DV-Hop estimation these two are considered as equal resulting in subsequent positioning errors. Anchors which are far away from an unknown node contribute more to this error than anchors which are nearby.

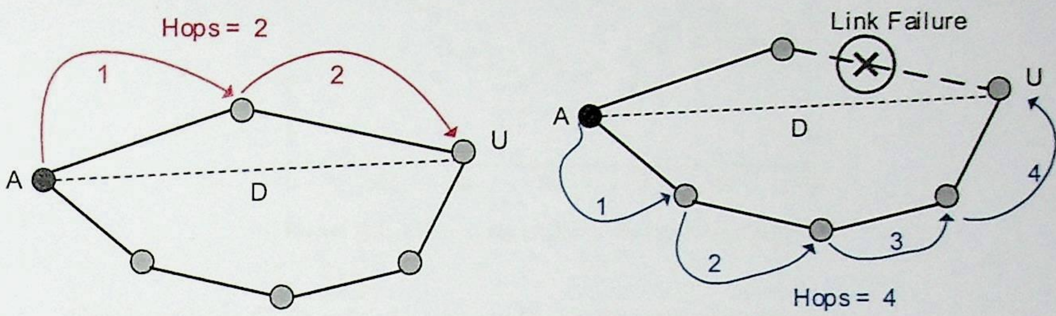
- *Dependency on number of anchors and their distribution:*

Analysis of the DV-Hop algorithm reveals that its localization accuracy depends on the anchor - to - node ratio (anchor ratio) and their distribution. When the anchor ratio of a network is high, localization accuracy is higher. Even distribution of anchors throughout the network also improves the localization accuracy. Since the cost of an anchor is significantly higher than a normal sensor node, it is not an option to increase the anchor node density in real life.

Figure 3.2(c) illustrates the localization error variation of DV-Hop algorithm with different number of anchor nodes while keeping the total number of nodes and the radio range of a sensor node a constant. It is evident that the localization error reduces with increasing number of anchors.

- *Connectivity problem:*

The DV-Hop algorithm estimates the locations of unknown nodes based on the connectivity information of the network. Connectivity problems due to node failures or environmental conditions such as obstacles affect the localization accuracy significantly. Figure 3.4 illustrates this concept.



(a) No link failure (b) Link failure

Figure 3.4: The effect of link failures in localization accuracy of DV-Hop algorithm

With DV-Hop algorithm;
when there are no link failures;

$$D = HopSize_u \times Hops = HopSize_u \times 2 \tag{3.2}$$

when the link is failed;

$$D = HopSize_u \times Hops = HopSize_u \times 4 \tag{3.3}$$

Therefore it can be seen that localization accuracy of the DV-Hop algorithm depends heavily on the proper connectivity among nodes.

- *Radio Range Irregularity:*

In the DV-Hop algorithm, it is assumed that the communication rage of all nodes are equal and regular. However in reality, the sensor nodes' communication is not a regular sphere in 3D space. Due to such radio range irregularities the localization accuracy deteriorates compared to what is expected theoretically. Figure 3.5 illustrates the radio range irregularity of sensor nodes.

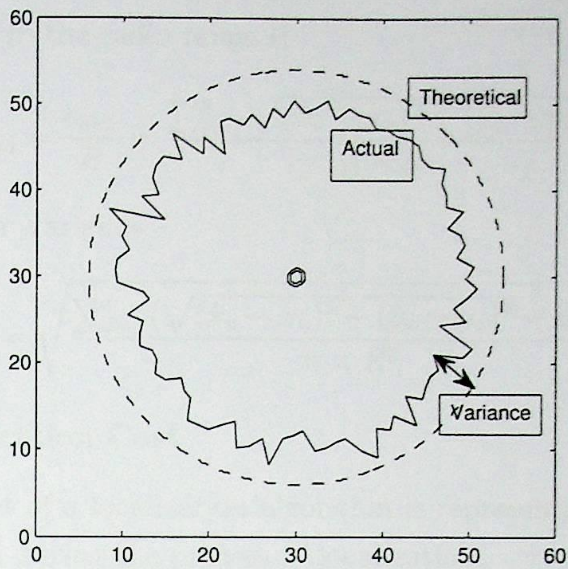


Figure 3.5: Illustration of Radio Range Irregularity

3.4 Evaluation Criteria for Localization Algorithms

In this Section, few matrices that are used in measuring the performances of localization algorithms are described.

3.4.1 Localization Accuracy

- Average Localization Error

$$e_{avg} = \frac{\sum_{k=1}^{n} \sqrt{(\hat{x}_u - x_u)^2 + (\hat{y}_u - y_u)^2}}{n} \tag{3.4}$$

- Root Mean Square Error

$$e_{rms} = \sqrt{\frac{\sum_{k=1}^{n} ((\hat{x}_u - x_u)^2 + (\hat{y}_u - y_u)^2)}{n}} \tag{3.5}$$

- Geometric Mean Error

$$e_{gm} = \sqrt[n]{\prod_{k=1}^{n} ((\hat{x}_u - x_u)^2 + (\hat{y}_u - y_u)^2)} \tag{3.6}$$

Where n is the number of unknown nodes, $(\hat{x}_u, \hat{y}_u)$ is the estimated location of unknown node u and (x_u, y_u) is the actual location of unknown node u.
For comparison of the performances of localization algorithms, e_{avg} is frequently used as a ratio of radio range R.

- Error normalized to the radio range R

$$e_{na} = \frac{e_{avg}}{R} = \frac{\sum_{k=1}^{n} \sqrt{(\hat{x}_u - x_u)^2 + (\hat{y}_u - y_u)^2}}{n \times R} \tag{3.7}$$

- Localization Error Variance

$$var = \sqrt{\frac{\sum_{k=1}^{n} (\sqrt{(\hat{x}_u - x_u)^2 + (\hat{y}_u - y_u)^2} - e_{avg})^2}{n \times R^2}} \tag{3.8}$$

3.4.2 Communication Cost

Communication cost of a localization algorithm is represented by the number of packets transmitted during the process of localization.

3.4.3 Computational Cost

Computational cost of a localization algorithm is represented as the computational time that takes for the process of localization.

3.5 Behavior Analysis of Localization Accuracy

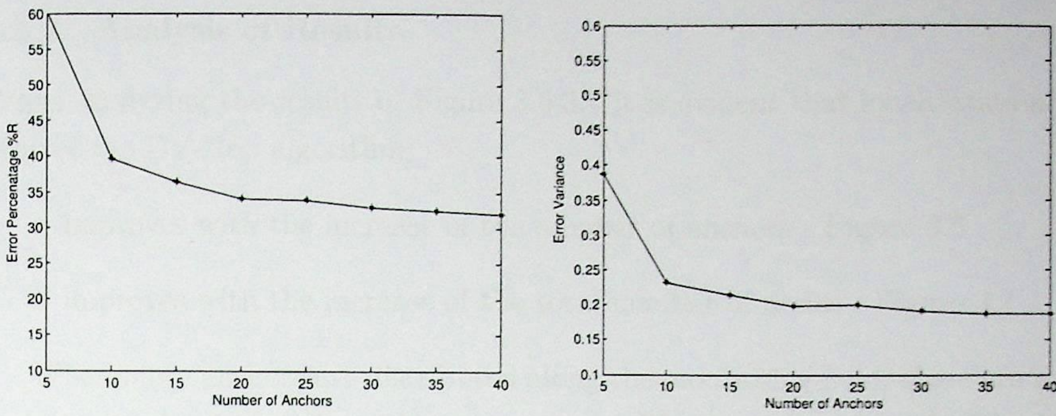
3.5.1 Simulation Scenarios

To evaluate the performance of the DV-Hop algorithm, Matlab based simulations were used. In the simulation environment both anchor and unknown nodes are distributed randomly according to a uniform pdf as show in Figure 3.1.

Localization error and localization error variance are used to evaluate the accuracy and the stability of the network. A scenario of 200 nodes, each having a radio range of $22m$ in a $100m \times 100m$ area is used. Then, the DV-Hop algorithm is studied for its performance by varying one at a time, the anchor ratio (AR), the total number of nodes (N) and the radio range (R) of the nodes as summarized in Table 5.1. All the simulation results are averaged over 100 runs.

Table 3.2: Simulation Instances

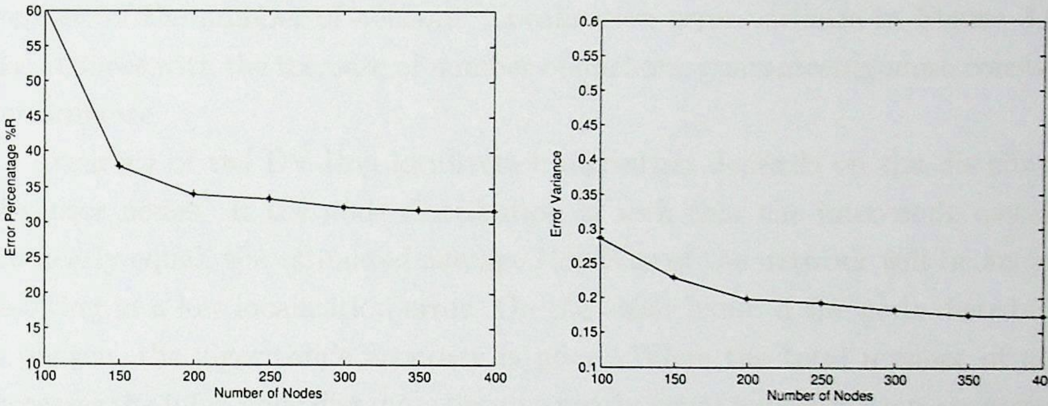
Figure No	Total Number of Nodes (N)	Anchor Ratio(%) (AR)	Radio Range(m) (R)
3.6(a),(b)	200	variable 5-40	22
3.7(a),(b)	variable 100-400	10	22
3.8(a),(b)	200	10	variable 15-40



(a) (N=200, R=22m)

(b) (N=200, R=22m)

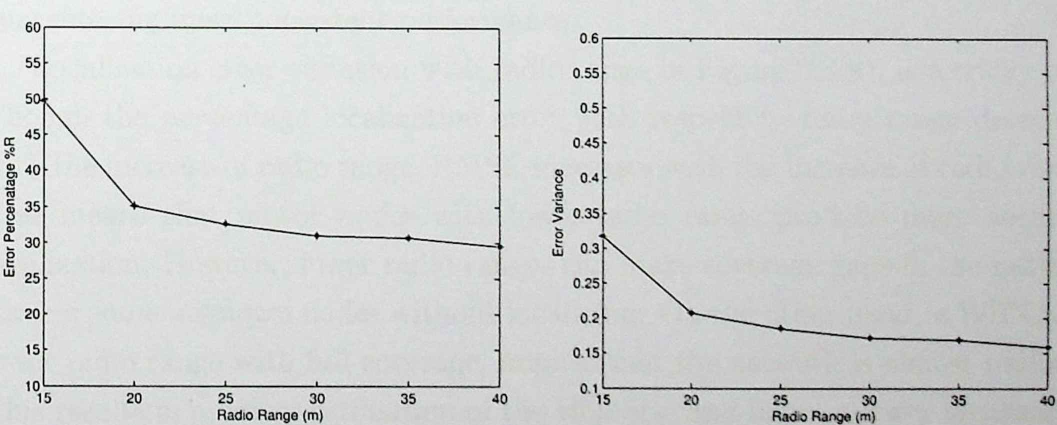
Figure 3.6: (a) Error Percentage with different number of anchors (b) Error Variance with different number of anchors



(a) (AR=10%, R=22m)

(b) (AR=10%, R=22m)

Figure 3.7: (a) Error Percentage with different number of sensor nodes (b) Error Variance with different number of sensor nodes



(a) (N=200, A=20)

(b) (N=200, A=20)

Figure 3.8: (a) Error Percentage with different radio range (b) Error Variance with different radio range

3.5.2 Analysis of Results

When analyzing the results of Figure 3.6,3.7 it is evident that localization accuracy of the DV-Hop algorithm;

- improves with the increase of the number of anchors - Figure 3.6
- improves with the increase of the total number of nodes - Figure 3.7

When more anchors are distributed along the monitoring field, there are more and more unknown nodes with anchors as neighbors (one-hop away). Therefore localization error results due to multiple hops, reduces resulting a much accurate location estimation. Hop Size estimation itself becomes more accurate with the increase of the number of anchors. Localization error variance in Figure 3.6(b) also reduces with the increase of number of anchors, guaranteeing more consistent performance.

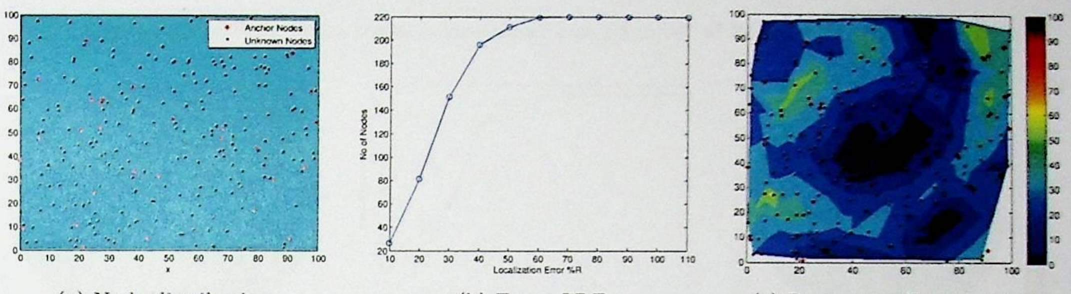
Accuracy of the DV-Hop localization algorithm depends on the distribution of sensor nodes. If the node distribution is such that the inter-node distances are nearly equal, the estimated average Hop Size of the network will be accurate resulting in a low localization error. On the other hand, if the node distribution is uneven, the algorithm's accuracy is poor. When the total number of nodes increases the inter-node distances become nearly equal to produce a more accurate estimation of Hop Size, resulting low localization error. Similarly, Localization error variance in Figure 3.7(b) also reduces with the increase of number of anchors, guaranteeing more consistent performance.

Localization error variation with radio range in Figure 3.8(a), is a tricky one. Though the percentage localization error with respect to radio range decreases with the increase of radio range, RMSE increases with the increase of radio range. This means that sensor nodes with lower radio range produce more accurate localization. However, lower radio ranges can make coverage gaps in the network leaving some unknown nodes without localizing. On the other hand, a WSN with lower radio range with full coverage, implies that the network is almost uniform. This results in a better estimation of the Hop Size and hence a lower localization error. A similar effect can be seen relative to the variation of localization error variance as in Figure 3.8(b).

3.6 Behavior Analysis of the DV-Hop in Different Shaped Environments

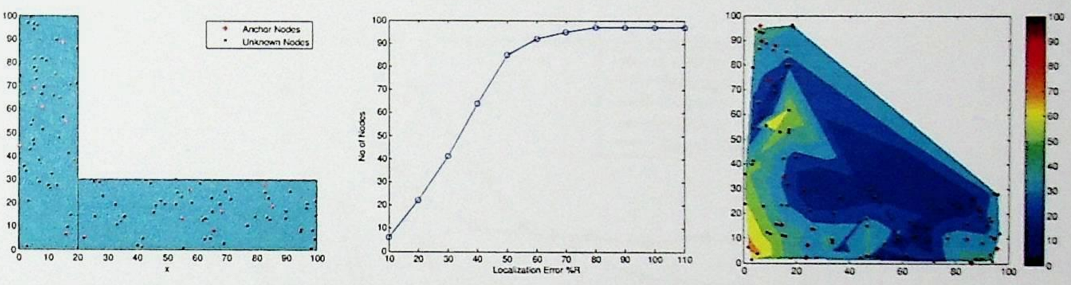
3.6.1 Simulation Scenarios

In this Section, we analyze the performance of DV-Hop algorithm in different shaped areas. In each area, node density (number of nodes/area) remains a constant.



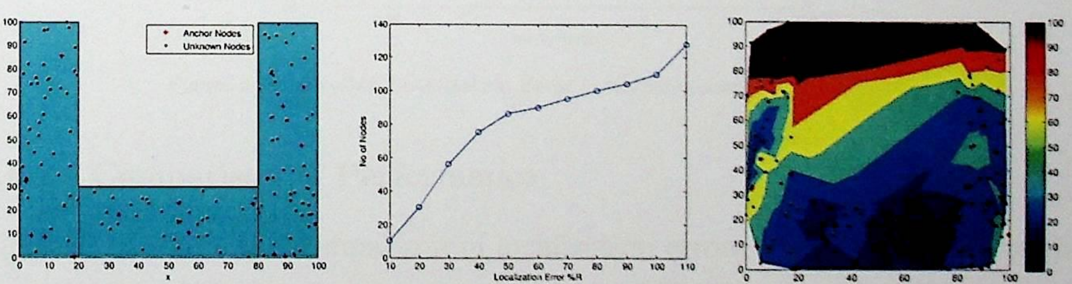
(a) Node distribution (b) Error CDF (c) Percentage Error contours

Figure 3.9: DV-Hop on a square shaped area (Number of Nodes = 250, Anchors = 30)



(a) Node distribution (b) Error CDF (c) Percentage Error contours

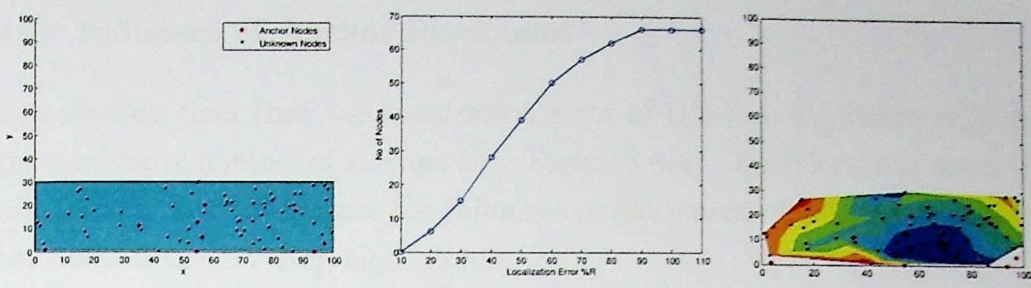
Figure 3.10: DV-Hop on a L shaped area (Number of Nodes = 110, Anchors = 13)



(a) Node distribution (b) Error CDF (c) Percentage Error contours

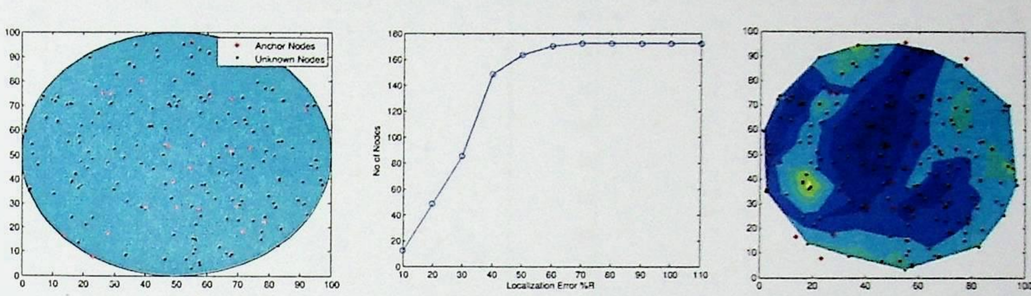
Figure 3.11: DV-Hop on a U shaped area (Number of Nodes = 145, Anchors = 17)

Figures 3.9, 3.10, 3.11, 3.12, 3.13 shows the node distribution in square shaped area, L shaped area, U shaped area, rectangular shaped area and O shaped area with localization error of each sensor node.



(a) Node distribution (b) Error CDF (c) Percentage Error contours

Figure 3.12: DV-Hop on a rectangular shaped area (Number of Nodes = 75, Anchors = 9)



(a) Node distribution (b) Error CDF (c) Percentage Error contours

Figure 3.13: DV-Hop on a O shaped area (Number of Nodes = 196, Anchors = 24)

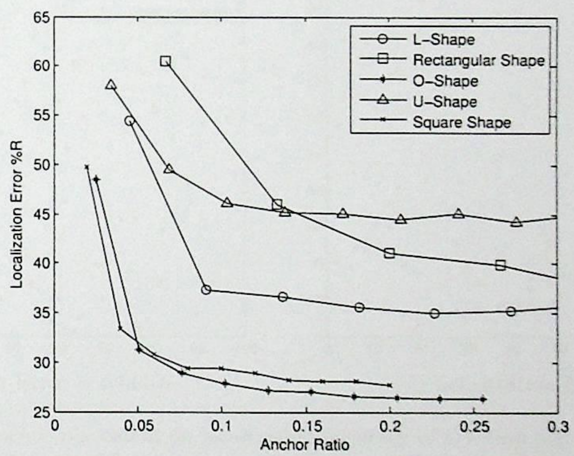


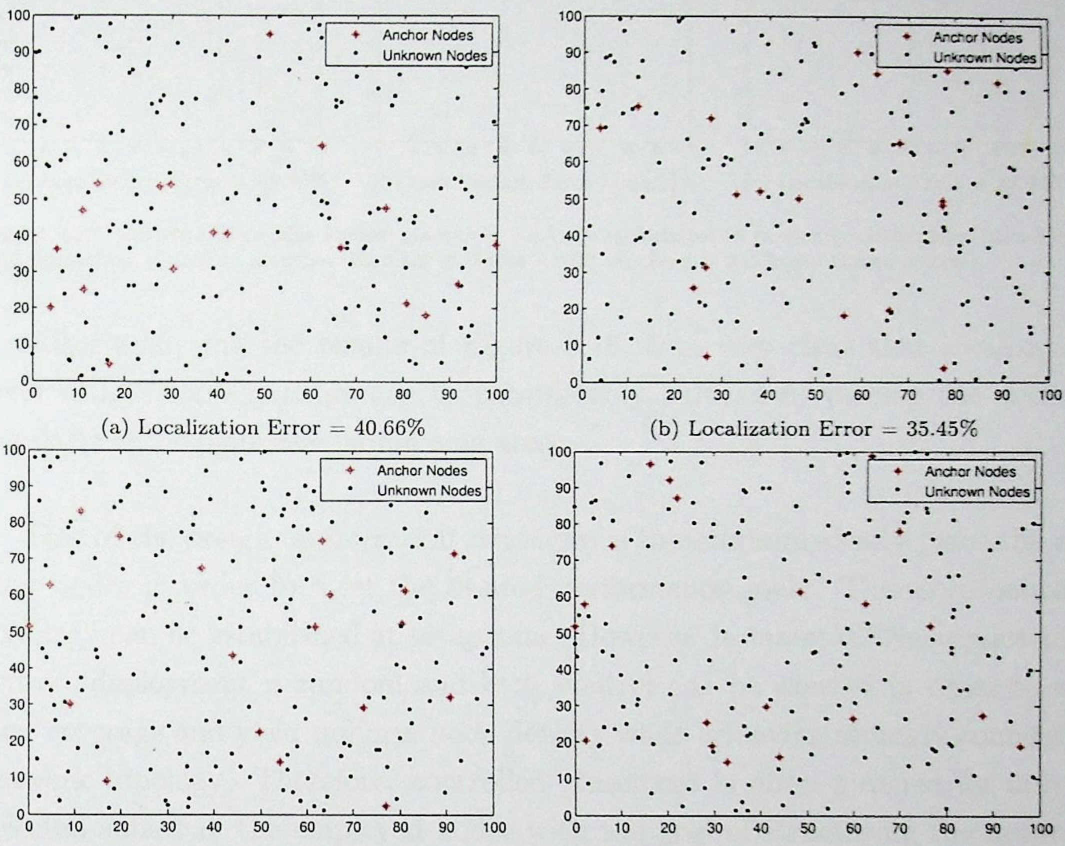
Figure 3.14: DV-Hop Localization Error in different shaped environments

3.6.2 Comparison of Performance

Figure 3.14 shows the comparison of localization errors when DV-Hop algorithm is applied in different shaped environments while keeping the node density a constant. It is evident that localization error in O shape and square shaped is minimum compared to the others.

3.7 Influence of Anchor Placement

It is already clear that the localization error of DV-Hop algorithm reduces with the increase of number of anchors as in Figure 3.6(a). Therefore, our main focus in this Section is to investigate the influence of placement of anchors on localization accuracy of the DV-Hop algorithm.



(a) Localization Error = 40.66%

(b) Localization Error = 35.45%

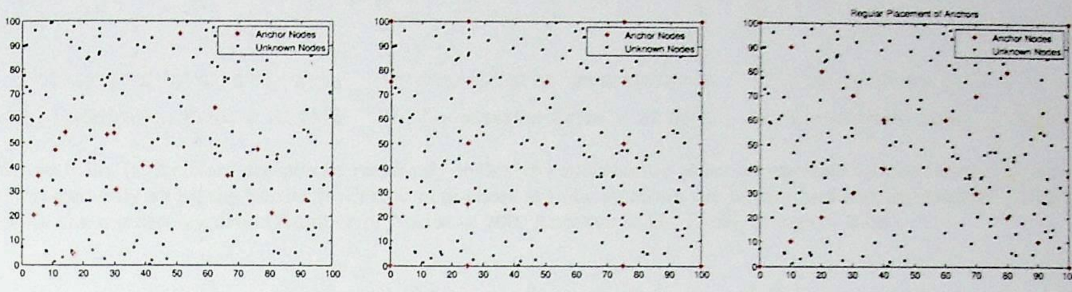
(c) Localization Error = 34.08%

(d) Localization Error = 33.72%

Figure 3.15: Influence of anchor placement on localization accuracy of DV-Hop algorithm (Number of Nodes = 200, Anchors = 20, Radio Range = 22m)

Figure 3.15 is used to illustrate the effect of the anchor placement on localization accuracy of the DV-Hop algorithm. Here the placement anchors is changed while keeping the placement of unknown nodes same in all the four configurations and evaluate the percentage localization error. Through simulations, it is evident that the localization error is associated with the placement of the anchors. The more regularly the anchors are placed, the lower the localization error. On the other hand, the more irregularly anchors are placed, the bigger the localization error. Therefore it is a fair conclusion to make that when anchors are placed regularly, localization error of DV-Hop algorithm reduces.

In order to further investigate this argument, we evaluate the localization error variation for few sensor networks with regular anchor placement in Figure 3.16 while keeping the known node distribution the same.



(a) Localization Error = 40.66% (b) Localization Error = 35.11% (c) Localization Error = 32.94%

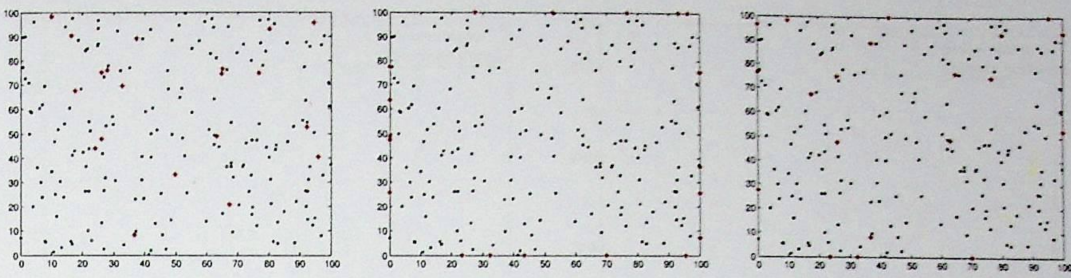
Figure 3.16: Influence of regular anchor placement (a)Anchors distributed randomly (b)Anchors placed on a grid (c)Anchors placed on diagonal (Number of Nodes = 200, Anchors = 20, Radio Range = 22m)

After analyzing the results of Figure 3.16, it is very clear that localization error of DV-Hop algorithm can be considerably reduced by placing the anchors regularly throughout the monitoring area.

One of the design optimization strategies is to deterministically place the anchor nodes in order to meet the desired performance goals. Therefore network topology can be established at setup time. However, in many WSNs applications sensors deployment is random and little control can be exerted in order to ensure coverage and yield uniform node density while achieving strongly connected network topology. Therefore, controlled placement is often pursued for only a selected subset of the employed nodes with the goal of structuring the network topology in a way that achieves the desired application requirements. On the other hand in some applications random distribution of anchors is the only feasible option.

Considering most of the practical applications, we combined the regular anchor placement with random deployment to produce a hybrid anchor placement scheme. Figure 3.17 shows the comparison of localization error of the three anchor placement schemes.

Optimal anchor placement is a challenging problem in DV-Hop based localization. Through simulations, Hybrid method is the best way to place the anchors. Therefore, our suggestion is to place a portion of anchors along the border of the monitoring area and rest to deploy randomly inside the monitoring area.



(a) Localization Error = 34.75% (b) Localization Error = 32.38% (c) Localization Error = 29.44%

Figure 3.17: (a)Anchors are placed randomly inside the monitoring area, no anchors on the border (b)Anchors are placed only along the border (c)Potion of anchors are placed along the border and rest are randomly deployed inside the monitoring area (Number of Nodes = 200, Anchors = 20, Radio Range = 22m)

3.8 Grid based Sensor Networks

Sensors can generally be placed in a monitoring area deterministically or randomly. The choice of the deployment scheme depends highly on the type of sensors, application and the environment that the sensors will operate in. Controlled node deployment is viable and often necessary when sensors are expensive or when their operation is significantly affected by their position. Controlled deployment is usually pursued for indoor applications of WSNs. Deploying sensors on a grid is such a controlled deployment scheme which is suitable for indoor environments.

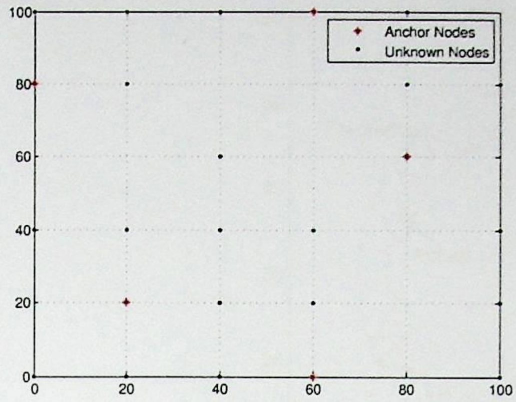
In this Section, we will evaluate the performance of DV-Hop algorithm in the process of locating the grid based sensor nodes. Though sensors are placed on a predefined grid, they are still placed on randomly selected grid points.

Figure 3.18 illustrates three scenarios of grid based networks with respective localization error. A mechanism to improve the localization error of grid based sensor networks will be presented in the next Section.

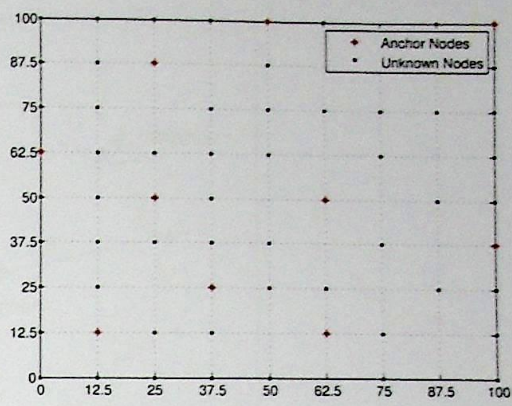
3.9 Effect of Radio Range Irregularity on DV-Hop

Radio irregularity is a common and non-negligible phenomenon in wireless sensor networks. It results in irregularity in radio range as shown in Figure 3.5.

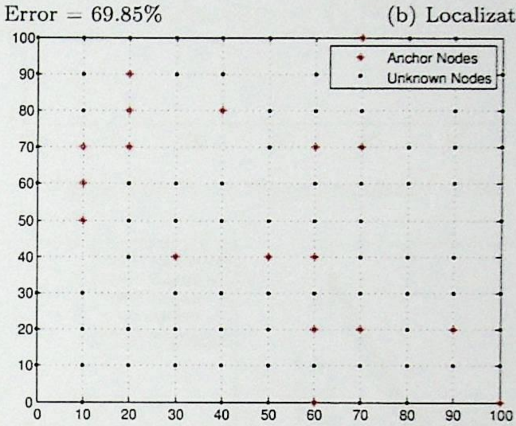
Radio irregularity is caused by two categories of factors: devices and the propagation media. Device properties include the antenna type (directional or omni-directional), the transmit power, antenna gains (at both the transmitter and receiver), receiver sensitivity, receiver threshold and the Signal-Noise Ratio (SNR). Media properties include the media type, the background noise and some



(a) Localization Error = 69.85%



(b) Localization Error = 46.66%



(c) Localization Error = 42.36%

Figure 3.18: (a)25 nodes on a grid (Anchors = 5, Radio Range = 30m) (b)64 nodes on a grid (Anchors = 10, Radio Range = 20m) (c)100 nodes on a grid (Anchors = 20, Radio Range = 15m)

other environmental factors, such as the temperature and obstacles within the propagation media.

We modeled the radio range irregularity by taking the difference between the actual radio range and the theoretical radio range, as a Gaussian random variable [62]. Variance between actual and the theoretical radio range is a critical measurement of the radio irregularity as shown in Figure 3.19. Therefore we apply this radio range irregularity model with different variances between actual and the theoretical radio range in DV-Hop localization process.

The DV-Hop localization algorithm assumes a spherical radio range. The study in Figure 3.20 shows that the performance of such DV-Hop algorithm degrades when the radio range becomes irregular.

The performance of the DV-Hop algorithm degrades when the variance of the radio range irregularity increases. This is mainly due to missing some of the di-

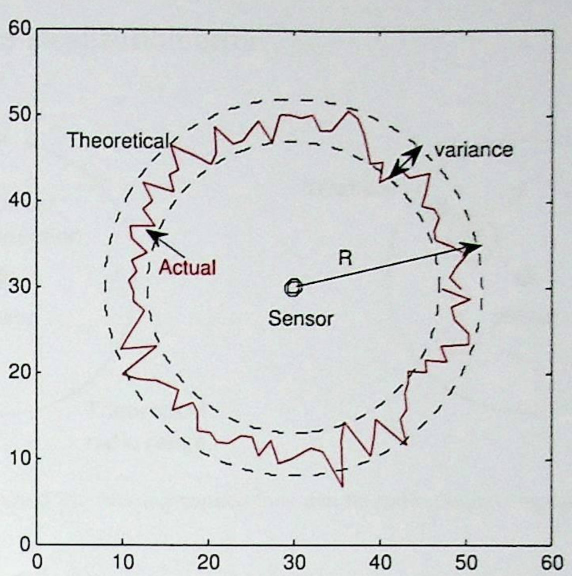
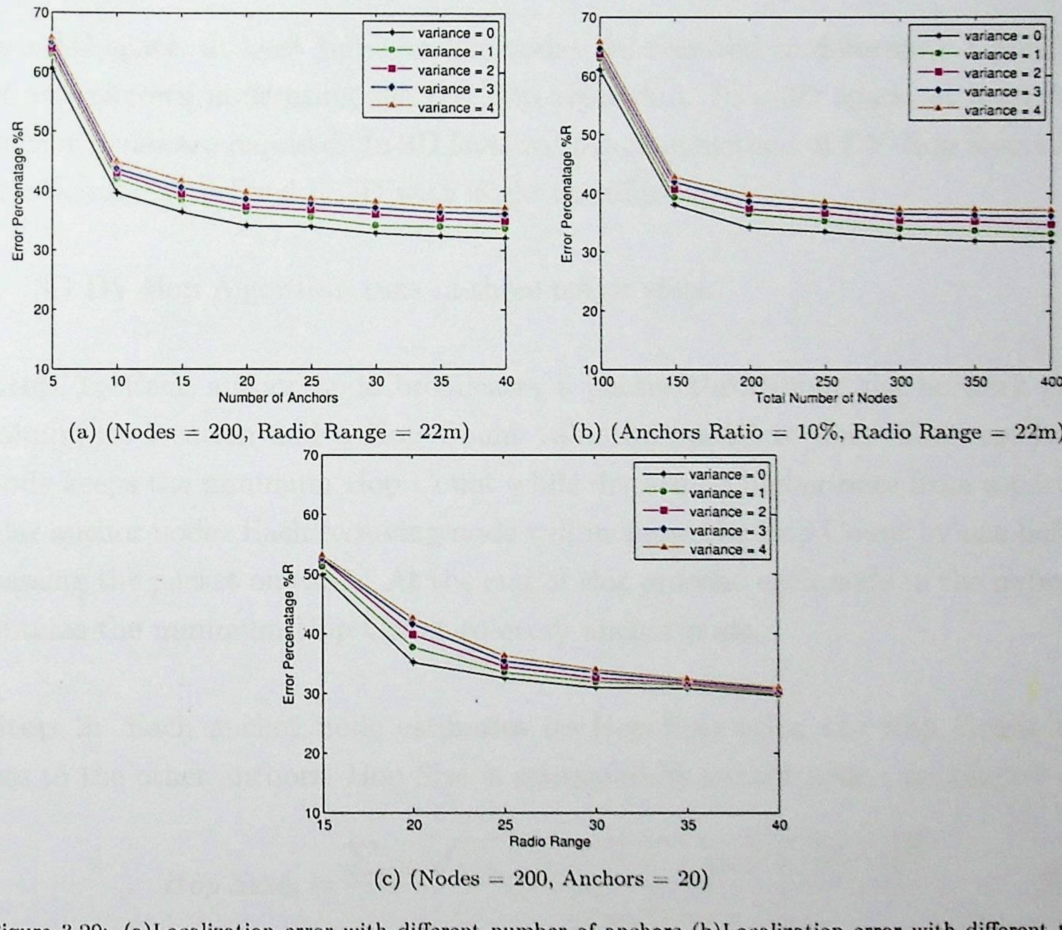


Figure 3.19: Radio Range Irregularity Model



(a) (Nodes = 200, Radio Range = 22m)

(b) (Anchors Ratio = 10%, Radio Range = 22m)

(c) (Nodes = 200, Anchors = 20)

Figure 3.20: (a)Localization error with different number of anchors (b)Localization error with different total number of nodes (c)Localization error with different radio ranges

rect links between a particular sensor node and its neighbors as shown in Figure 3.21. Therefore, number of hops between certain nodes can be increased and hence results a more localization error.

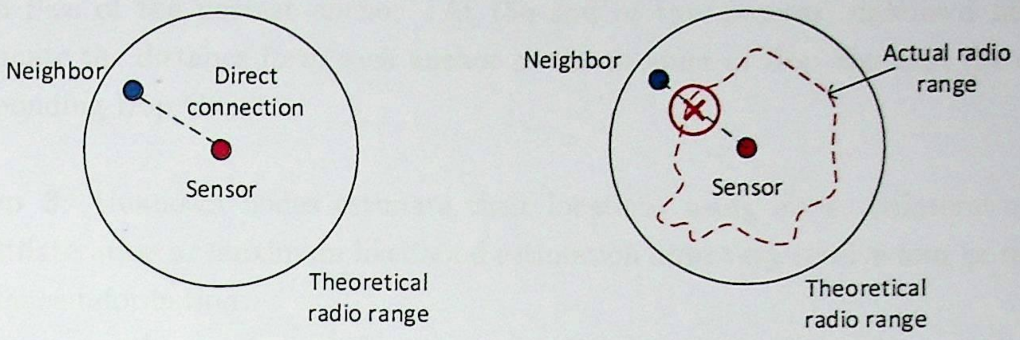


Figure 3.21: Missing connections due to radio range irregularity

3.10 DV-Hop in 3D Environments

In a 2D space, at least three anchor nodes are required to determine a position of an unknown node using the DV-Hop algorithm. In a 3D space, at least four anchor nodes are required. In 3D localization same method of DV-Hop algorithm can be used as defined in 2D with slight modification.

3D DV-Hop Algorithm runs in three major steps.

Step 1: Each anchor node broadcasts a packet throughout the network containing its location and a Hop Count value initialized to one. Each receiving node keeps the minimum Hop Count while discarding higher ones from a particular anchor node. Each receiving node will increase the Hop Count by one before passing the packet onwards. At the end of this process, each node in the network obtains the minimum Hop Count to every anchor node.

Step 2: Each anchor node estimates its Hop Size using the Hop Count values to the other anchors. Hop Size is estimated by anchor node i as follows;

$$Hop\ Size_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}}{\sum_{j \neq i} hop_{ij}} \tag{3.9}$$

where (x_i, y_i, z_i) and (x_j, y_j, z_j) are the coordinates of anchors i and j respectively and hop_{ij} is the Hop Count between anchors i and j.

Each anchor calculates its Hop Size and broadcasts it to the network. Unknown nodes save the first value they receive, while transmitting this Hop Size to the neighbors. This mechanism results in most of the unknown nodes obtaining the Hop Size of the nearest anchor. At the end of this process, unknown nodes estimate the distance form each anchor as the product of Hop Size and the corresponding Hop Count.

Step 3: Unknown nodes estimate their locations using either trilateration / multilateration or maximum likelihood estimation after they receive four or more distance information.

Let (x_u, y_u, z_u) be the location of the unknown node u and (x_j, y_j, z_j) be the known location of anchor node j. d_{uj} is the distance between them. Then;

$$\begin{gathered}(x_u - x_1)^2 + (y_u - y_1)^2 + (z_u - z_1)^2 = d_{u1}^2 \\ (x_u - x_2)^2 + (y_u - y_2)^2 + (z_u - z_2)^2 = d_{u2}^2 \\ \vdots \\ (x_u - x_n)^2 + (y_u - y_n)^2 + (z_u - z_n)^2 = d_{un}^2\end{gathered}$$

where n is the number of anchors in the network.

Coordinates of the unknown node u can be calculated using the following matrix operation.

$$A = -2 \times \begin{bmatrix} x_1 - x_n & y_1 - y_n & z_1 - z_n \\ x_2 - x_n & y_2 - y_n & z_2 - z_n \\ \vdots & \vdots & \\ x_{n-1} - x_n & y_{n-1} - y_n & z_{n-1} - z_n \end{bmatrix} \tag{3.10}$$

$$B = \begin{bmatrix} d_{u1}^2 - d_{un}^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 - z_1^2 + z_n^2 \\ d_{u2}^2 - d_{un}^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 - z_2^2 + z_n^2 \\ \vdots \\ d_{u(n-1)}^2 - d_{un}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 - z_{n-1}^2 + z_n^2 \end{bmatrix} \tag{3.11}$$

$$P = \begin{bmatrix} x_u \\ y_u \\ z_u \end{bmatrix} = (A^T A)^{-1} A^T B \tag{3.12}$$

Figure 3.22 shows the distribution of sensor nodes in a 3D space.

Figure 3.23 shows the localization error and error variance variation after

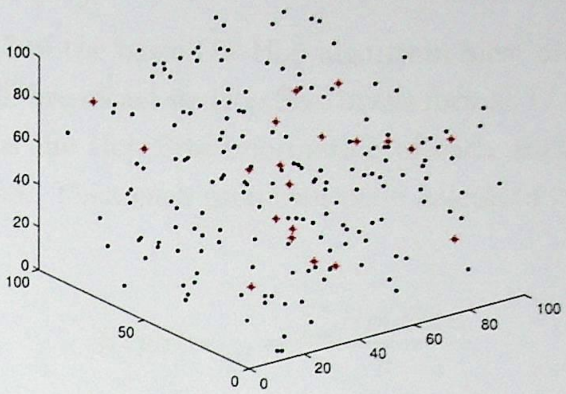
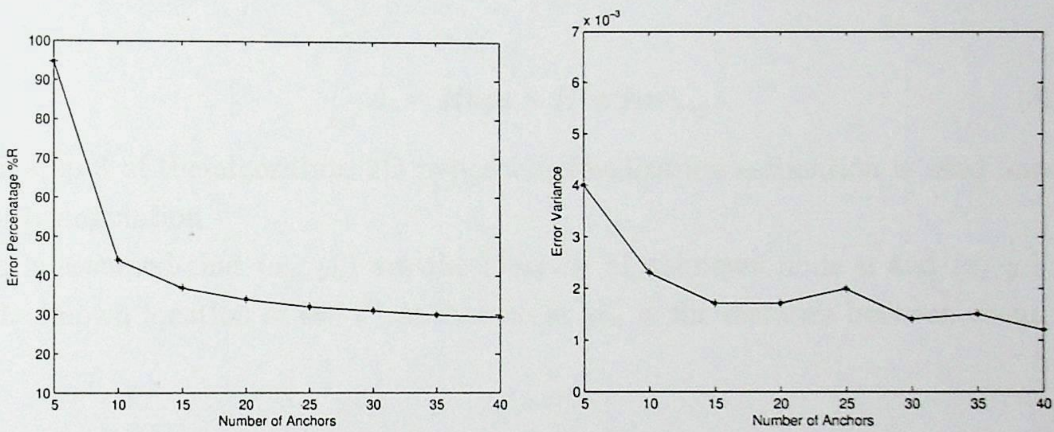


Figure 3.22: Sensor node distribution in a 3D environment

applying the 3D DV-Hop algorithm for the node distribution shown in Figure 3.22



(a) (b)

Figure 3.23: (a)Localization error with different number of anchors (b)Localization error variance with different number of anchors (Nodes = 200, Radio Range = 50m, Area = 100m×100m×100m)

3.11 Performance Analysis of Selected Improved DV-Hop Algorithms

In this Section we evaluate the performances of five selected improved DV-Hop algorithms from literature. Localization accuracy of each algorithm is compared through simulations.

3.11.1 An Improved DV-Hop Localization Algorithm for Wireless Sensor Networks (DV-Hop Average)

This algorithm is based on the work done in [19]. We refer to this algorithm as "DV-Hop Average".

Method

In [19], steps 2 and 3 of the basic DV-Hop algorithm have been changed. In step 2, all the anchors will broadcast its Hop Size in the format $\{ID, HopSize_i\}$. Each unknown node keeps the Hop Size information of each anchor while discarding the duplicate packets. Then each unknown node calculate its Average Hop Size as follows.

$$HopSize_{avg} = \frac{\sum HopSize_i}{n} \tag{3.13}$$

where n is the number of anchor nodes and $HopSize_i$ is the Hop Size of i^{th} anchor node.

After this, unknown nodes will calculate the distances to each anchor node as follows.

$$d_i = Hops \times HopSize_{avg} \tag{3.14}$$

In step 3 of the algorithm, 2D hyperbolic localization estimation is used instead of triangulation.

It is assumed that (x_u, y_u) are the location of unknown node u and (x_i, y_i) are the known location of the i^{th} anchor node. d_{ui} is the distance between them.

Method

In [19], steps 2 and 3 of the basic DV-Hop algorithm have been changed. In step 2, all the anchors will broadcast its Hop Size in the format $\{ID. HopSize_i\}$. Each unknown node keeps the Hop Size information of each anchor while discarding the duplicate packets. Then each unknown node calculate its Average Hop Size as follows.

$$HopSize_{avg} = \frac{\sum HopSize_i}{n} \tag{3.13}$$

where n is the number of anchor nodes and $HopSize_i$ is the Hop Size of i^{th} anchor node.

After this, unknown nodes will calculate the distances to each anchor node as follows.

$$d_i = Hops \times HopSize_{avg} \tag{3.14}$$

In step 3 of the algorithm, 2D hyperbolic localization estimation is used instead of triangulation.

It is assumed that (x_u, y_u) are the location of unknown node u and (x_i, y_i) are the known location of the i^{th} anchor node. d_{ui} is the distance between them.

$$(x_u - x_i)^2 + (y_u - y_i)^2 = d_{ui}^2 \tag{3.15}$$

$$x_i^2 + y_i^2 - 2x_u x_i - 2y_u y_i + x_u^2 + y_u^2 = d_{ui}^2$$

$$d_{ui}^2 - E_i = -2x_u x_i - 2y_u y_i + K$$

$$where; \quad E_i = x_i^2 + y_i^2 and K = x_u^2 + y_u^2$$

$$let; \quad Zc = [x_u, y_u, K]^T$$

$$G_c = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_n & -2y_n & 1 \end{bmatrix} h_c = \begin{bmatrix} d_{u1}^2 - E_1 \\ d_{u2}^2 - E_2 \\ \vdots \\ d_{un}^2 - E_n \end{bmatrix}$$

$$G_c Z_c = h_c$$

$$Zc = (G_c^T G_c)^{-1} G_c^T h^c$$

$$x_u = Z_c(1) \quad and \quad y_u = Z_c(2)$$

3.11.2 A New Improved DV-Hop Localization Algorithm (DV-Hop Marker)

This algorithm is based on the work done in [35]. We refer to this algorithm as "DV-Hop Marker".

Method

In [35], an additional phase has been added after the step 3 of basic DV-Hop algorithm. Analysis of the step 2 of the DV-Hop algorithm reveals that, for the unknown nodes from a certain small area always get the same hop-distance value and similar Hop Count. Hence they tend to have the similar location error. The additional phase is added based on this notion.

In this additional phase, some anchor nodes are selected as regulated nodes which are either placed manually or randomly. Instead of using these regulated nodes as conventional anchors, they are considered as unknown nodes to be located.

Let (x_i, y_i) be the calculated coordinate and (x_{ri}, y_{ri}) be the actual coordinate of the regulated node i. Then the error vector R_i can be found as follows.

$$R_i = \begin{bmatrix} x_i - x_{ri} \\ y_i - y_{ri} \end{bmatrix} \tag{3.16}$$

Then the regulated node broadcasts the packet $\{HopSize, R_i\}$ to neighbors. Each unknown node will compare the regulated node's Hop Size with its Hop Size received in step 2 of the basic DV-Hop localization. If both the Hop Size values are equal, the unknown node will change the coordinates as follows.

$$(x_u, y_u) = (x_m, y_m) - R_i \tag{3.17}$$

where (x_m, y_m) is the former location of unknown node m. If Hop Size values are not equal, the packet can be discard without modifying the coordinates.

3.11.3 A Weighted DV-Hop Localization Scheme for Wireless Sensor Networks (DV-Hop Weighted)

This algorithm is based on the work done in [20]. We refer to this algorithm as "DV-Hop Weighted".

Method

In [20], a weight has been given for each anchor node to improve the accuracy of the basic DV-Hop algorithm. The idea behind giving a weight for each anchor is to represent the impact of each anchor, in localizing the unknown nodes. The weight of each anchor node is a production of hop size weight and distance weight. Hop Size weight reflects the accuracy of Hop Size of the anchor and Distance weight reflects the distance between the anchor and the unknown node. The proposed algorithm is divided into three phases.

- Hop Count Computation

This phase is very similar to the step 1 of DV-Hop algorithm, where each unknown node find the minimum Hop Count to each anchor node.

- Hop Size and Weight Computation

This phase has been divided into three steps.

Step 1:

$$HopSize_{ij} = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{h_{ij}} \tag{3.18}$$
$$HopSize_i = \frac{\sum j \neq iHopSize_{ij}}{N-1}$$

Where (x_i, y_i) and (x_j, y_j) are coordinates of anchor nodes i and j. h_{ij} is the Hop Count between them. N is the number of anchor nodes. $HopSize_i$ is the Hop Size of anchor node i. Then, anchor node i calculates the Hop Size Weight of anchor node j.

$$Wd_{ij} = 1 - \frac{(HopSize_{ij} - HopSize_i)^2}{r^2} \tag{3.19}$$
$$if \quad Wd_{ij} < 0 \rightarrow Wd_{ij} = 0$$

where r is the radio range of the sensor node.
Finally each anchor node broadcasts the $\{HopSize_i, Wd_{ij}\}$ to all the nodes.

Step 2:
Unknown node i calculates the Hop Size and the Distance Weight Wh_j of anchor j relative to node i.

$$HopSize = \frac{\sum_{k=1}^{N} HopSize_k}{N} \tag{3.20}$$
$$Wh_j = \frac{1}{h_{ij}}$$

Step 3:
Unknown node i calculates the distance to the anchor node j and weight of anchor node j as follows.

$$d_{ij} = h_{ij} \times HopSize \tag{3.21}$$
$$W_{ij} = Wh_j \times \frac{\sum_{k=1}^{N} Wd_{jk}}{N}$$

- Location Estimation

Unknown nodes will calculate their locations using weighted least squares method.

$$f(x,y) = \min \sum_{i=1}^{n} w_i^2 (\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - d_i)^2 \tag{3.22}$$

where w_i is the weight of the anchor node i.

3.11.4 Improved DV-Hop Node Localization Algorithm in Wireless Sensor Networks (DV-Hop Improved)

This algorithm is based on the work done in [26]. We refer to this algorithm as "DV-Hop Improved".

Method

The algorithm in [26] has been divided in to four steps. In summary, (1) some anchor nodes were fixed at the border land of the monitoring regions. (2) The Hop Size calculation was modified, and the average one-hop distance used by each unknown node for estimating its location was modified. (3) The unknown nodes' positions were obtained by the 2D hyperbolic location algorithm. (4)Particle swarm optimization algorithm was used to correct the estimated position.

Step 1: Deploy Anchor Nodes

Some of the anchor nodes were fixed at the boarder of the monitoring area and rest were distributed randomly trough out the field. This avoided the concentration of anchors in an area and hence producing larger errors.

Step 2: Calculation of the Average Hop Distance of Unknown Nodes

The average hop distance of every anchor node $HopSize_i$ was calculated as follows.

$$HopSize_i = \frac{\sum_{j \neq i}(hop_{ij} d_{ij})}{\sum_{j \neq i} hop_{ij}^2}, \quad i \neq j \tag{3.23}$$

where hop_{ij} is the hop count between the anchor nodes i and j, d_{ij} is the straight-line distance between the anchor nodes i and j.

Then, average one-hop distance of unknown nodes $HopSize_u$ was modified through weighting the N received average one-hop distances from the anchor nodes by the following formula.

$$HopSize_u = \sum_{i=1}^{n} \lambda_i HopSize_i \tag{3.24}$$

$$\lambda_i = \frac{hop_i}{\sum_{k=1}^{n} hop_k}$$

where hop_i is the hop segment number between the unknown node u and the anchor node i in its routing table and n is the number of all anchor nodes.

Step 3: Position Determination of Unknown Nodes
2-D Hyperbolic location algorithm was used instead of the traditional trilateral or the multilateral position method to estimate the position of unknown nodes.

Let (x_i, y_i) be the coordinates of anchor node i and (x_u, y_u) be the coordinates of unknown node u.
The estimated distance d_{iu} between them is;

$$d_{iu}^2 = (x_i - x_u)^2 + (y_i - y_u)^2 \tag{3.25}$$

If $A_i = x_i^2 + y_i^2$ and $B_i = x_u^2 + y_u^2$, then;

$$d_{iu}^2 - A_i = -2x_i x_u - 2y_i y_u + B_u \tag{3.26}$$

This can be written in matrix form as;

$$GZ = h \tag{3.27}$$

where

$$Z = [x_u, y_u, B_u]^T \quad G = \begin{bmatrix} -2x1 & -2yi & 1 \\ -2x2 & -2y2 & 1 \\ \vdots & \vdots & \vdots \\ -2xn & -2yn & 1 \end{bmatrix} \tag{3.28}$$

$$h = \begin{bmatrix} d_{1u}^2 - A_1 \\ d_{2u}^2 - A_2 \\ \vdots \\ d_{nu}^2 - A_n \end{bmatrix}$$

Z can be obtained by using the least mean square estimation method:

$$Z = (G_T G)^{-1} G^T h \tag{3.29}$$

Then, coordinates of unknown node u are;

$$x_u = Z(1) \quad y_u = Z(2) \tag{3.30}$$

Step 4: Particle Swarm Optimization

Each particle has an objective function determined by the fitness value, and it also has a rate determined by the distance and direction of particle search. Each particle can update its velocity and position by;

$$\begin{aligned} v_u^{k+1} &= \omega v_u^k + c_1 rand_1^k (pbestu^k - X_u^k) \\ &+ c_2 rand_2^k (gbestu^k - X_u^k) \\ X_u^{k+1} &= X_u^k + v_u^{k+1} \end{aligned} \tag{3.31}$$

where v_u^k is the velocity vector, X_u^k is the current position vector of the particle,$pbest_u^k$ is the individual extreme of the particle u by $k - cycles$, and $gbest_u^k$ is the global extreme of all particles by $k - cycles$. In the typical PSO implementation,c1 and $c2$ are the acceleration factors and $rand_1^k$ and $rand_2^k$ are both random of [0,1]. k is the current number of iterations.

The inertia weight ω;

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter_{max}} \times k \tag{3.32}$$

where ω_{max} and ω_{min} are the initial inertia weight and the termination inertia weight respectively, $iter_{max}$ is the maximum number of iterations and k is the current number of iterations.

The fitness is designed as;

$$fitness(X_u) = \sum_{i=1}^{n} \alpha_i^2 (d_i - |X_u - A_i^T|)^2 \tag{3.33}$$

where A_i^T is the position of the anchor node and α_i is inversely proportional to the hop count between the unknown node u and the anchor node i.

3.11.5 Iterative Algorithm for Locating Nodes in WSN based on Modifying Average Hopping Distances (DV-Hop Iterative)

This algorithm is based on the work done in [63]. We refer to this algorithm as "DV-Hop Iterative".

Work reported in [26] is mainly based on the work carried out in this paper. The paper is originally published in Chinese and hence the method of this algorithm is not reported here. For simulation and comparison purposes we use the data set provided in [26] relative to [63].

3.12 Comparison through Simulations

3.12.1 Simulation Environment

To evaluate the performance of the improved algorithm, Matlab was used. In simulation environment both anchor and unknown nodes are distributed randomly according to a uniform pdf as show in Figure 3.1.

Number of anchors (n), total number of sensor nodes (N) and radio range(R) of the nodes are varied in order to study the performance of the algorithm. Localization error and localization error variance are used to evaluate the accuracy and the stability of the network. All the simulation results are averaged over 100 runs.

Matlab simulations are carried out to first examine the performance of the improved algorithms compared to the original DV-Hop. A scenario of 200 nodes, each having a radio range of $22m$ in a $100m \times 100m$ area is used. Then, the algorithms are studied for their performances by varying one at a time, the anchor ratio, the total number of nodes and the radio range of the nodes as summarized in Table 3.3. The localization error and the error variance are compared among them.

Figure 3.24, 3.25 and 3.26 show the comparison of performances of the improved DV-Hop algorithms.

Time Ration in 3.34 is a indicator how fast the localization of the nodes happens in a given network. Figure 3.27 shows the comparison of Time Ratio with

Table 3.3: Simulation Instances

Figure No	Total Number of Nodes	Anchor Ratio (%)	Anchor Range(m)
3.24	200	variable 5-40	22
3.25	variable 100-400	10	22
3.26	200	10	variable 15-40

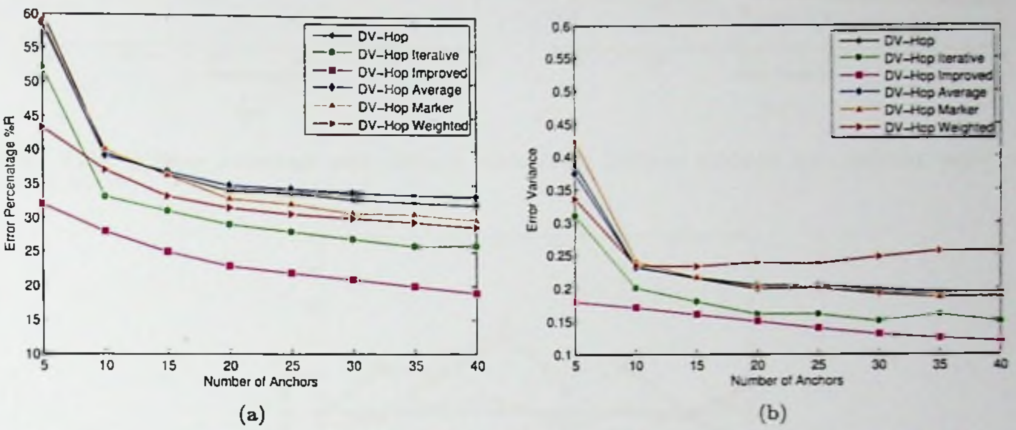


Figure 3.24: (a)Error percentage with different number of anchors (b)Error variance with different number of anchors (N=200,R=22m)

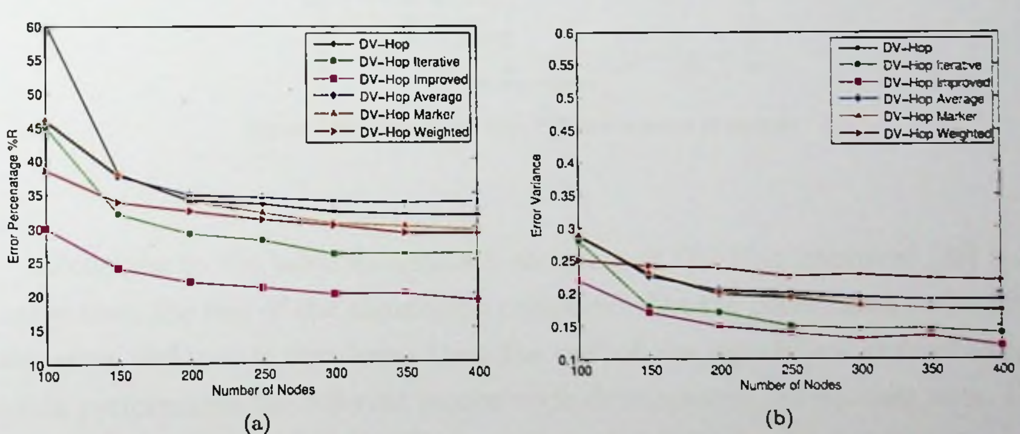


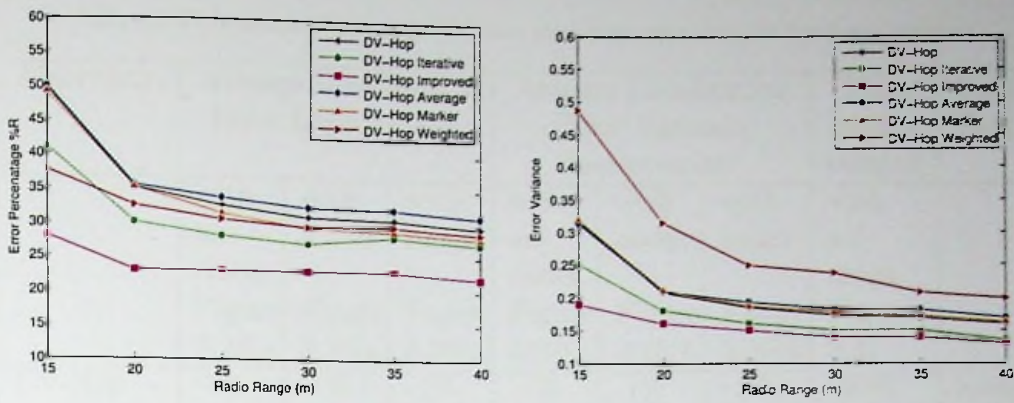
Figure 3.25: (a)Error percentage with different total number of nodes (b)Error variance with different total number of nodes (AR=10%,R=22m)

the varying number of anchors for different algorithms.

$$Time \quad Ratio = \frac{LocalizationTime_{GivenAlgorithm}}{LocalizationTime_{DV-Hop}} \tag{3.34}$$

3.12.2 Discussion

A summary of the performance of the compared algorithms is shown in the Table 3.4.



(a) (b)

Figure 3.26: (a)Error percentage with different radio range (b)Error variance with different radio range (N=200,A=20)

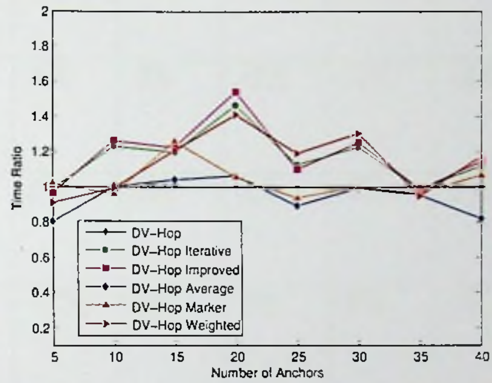


Figure 3.27: Time Ratio with different number of anchors

According to the table localization accuracy of DV-Hop Improved [26] is far better than the rest of the algorithms compared. On the other hand its localization error variance is also lower than the rest of the algorithms guaranteeing a stable performance for different sensor node distributions. In another note, DV-Hop Improved [26] does not need any additional distribution of packets compared to the DV-Hop algorithm ensuring the same communication cost. However, DV-Hop Improved [26] is around 10% slower than the DV-Hop algorithm due to the usage of the particle swarm optimization (PSO) algorithm and 2D Hyperbolic function in the process of localization.

DV-Hop Iterative [63] also characterizes the same behavior as the DV-Hop Improved [26], though its localization error is higher than that of DV-Hop Improved [26].

Table 3.4: Performance Comparison of selected algorithms relative to the DV-Hop algorithm

Algorithm	Average Localization Error Improvement			Average Localization Error Variance Improvement			Time Improvement	Com. Cost
	with anchors Figure 3.24(a)	with nodes Figure 3.25(a)	with range Figure 3.26(a)	with anchors Figure 3.24(b)	with nodes Figure 3.25(b)	with range Figure 3.26(b)	with anchors Figure 3.27	
DV-Hop Average [19]	-1%	-1%	-1%	-1%	-1%	-1%	-10%	lower
DV-Hop Marker [35]	3%	3%	1.5%	0	0	-1%	5%	higher
DV-Hop Weighted [20]	4%	4%	2%	-2%	-3%	-4%	15%	similar
DV-Hop Iterative [63]	6%	5%	3%	4%	4%	2%	18%	similar
DV-Hop Improved [26]	13%	13%	8%	7%	6%	3%	20%	similar

Localization error of the DV-Hop Weighted [20] is around 4% lower than that of the basic DV-Hop algorithm, but higher than that of both the DV-Hop Improved [26] and DV-Hop Iterative [63]. The most noticeable fact is that the localization error variance of DV-Hop Weighted [20] is greater than that of the basic DV-Hop algorithm. That means that the performance of the DV-Hop Weighted [20] is less consistent with different sensor node distributions, compared with basic DV-Hop algorithm. Therefore localization accuracy of the DV-Hop Weighted [20] can be more affected by the way of node distribution. Due to the weighting factor calculation process, computational complexity of the DV-Hop Weighted [20] increases compared to the basic DV-Hop algorithm. However DV-Hop Weighted [20] does not require any additional transmission of packets so that the communication cost remains same as the DV-Hop algorithm.

Localization error of the DV-Hop Marker [35] is around 3% lower than that of the basic DV-Hop algorithm, but greater than that of DV-Hop Improved [26], DV-Hop Iterative [63] and DV-Hop Weighted [20]. Localization error variance of the DV-Hop Marker [35] is almost similar as the DV-Hop algorithm. Due to the re-transmission of the Correction Factor to the network, DV-Hop Marker's [35] communication cost increases compared to the DV-Hop algorithm. However the network settling time increases only by around 5% compared to the DV-Hop algorithm. This Time for localization can be heavily dependent of the total number of nodes in the network, since re-transmission of the Correction Factor for a very large number of nodes may slow down the algorithm.

Localization error and error variance of the DV-Hop Average [19] is almost similar (if not a little high), compared with the DV-Hop algorithm. Most important fact about DV-Hop Average [19] is that its computational cost is lower than that of DV-Hop algorithm. DV-Hop Average [19] uses a common Hop Size for the whole network, so it does not need to compute different Hop Sizes for different nodes which results in a reduction of computational complexity. On the other hand, due to the usage of a common Hop Size for the entire network, DV-Hop Average [19] only requires to transmit the Hop Size once in the whole process of localization. Therefore communication cost of the DV-Hop Average [19] is lower than that of the DV-Hop algorithm.

Chapter 4

NOVEL APPROACHES FOR IMPROVED DV-HOP LOCALIZATION

Localization is a significant consideration in a Wireless Sensor Network (WSN), as it is essential to correlate sensor information with the geographic location it originates from. The DV-Hop algorithm, being a range-free, distributed algorithm, is an attractive option in this regard, due to its simplicity as discussed in Chapter 3. However it suffers from poor accuracy. In this Chapter, we present three algorithms to improve the localization accuracy of the DV-Hop algorithm. We examine the performance of the proposed algorithms through simulations in environments with varying node numbers, anchor ratios and radio range. The results show that the proposed algorithms have improved performance in localization accuracy and stability compared to the original DV-Hop algorithm.

4.1 Proposed Algorithm 1

Improved DV-Hop Algorithm Through Anchor Position Re-estimation

4.1.1 Overview

Examination of the DV-Hop algorithm indicates that the source of inaccuracy comes from the errors in the computation of distances d_{ij} used in the multilateration step. This error is caused in the Hop Size computation. The basic idea behind our approach is to use the errors in the estimation of known anchor locations using the DV-Hop algorithm to modify the Hop Size to reduce these errors.

The proposed algorithm can be divided in to the following four steps.

Step 1: Hop Size and Hop Count of the anchor nodes are found using Step 1 and Step 2 of the DV-Hop algorithm.

Step 2: Known positions of the anchor nodes are recalculated using Step 3 of the DV-Hop algorithm.

Step 3: Step 2 is iterated by modifying the Hop Size to minimize the average of anchor position errors. Through this, the optimum Hop Size Correction is obtained.

Step 4: Locations of unknown nodes are estimated with Step 3 of the DV-Hop algorithm using the modified Hop Size after applying the Hop Size Correction.

The algorithm is studied with a random uniformly distributed sensor bed as shown in Figure ??. Figure 4.1 summarize the proposed algorithm.

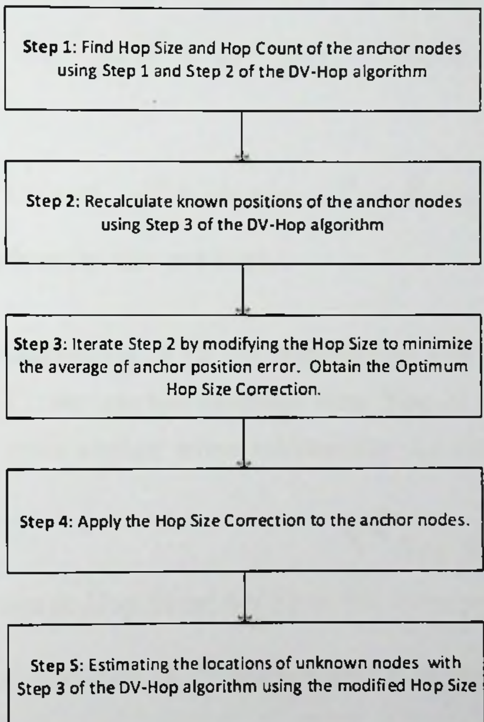


Figure 4.1: Algorithm 1

4.1.2 Recalculating the Anchor Positions (Step 2 of the Proposed Algorithm)

First the Hop Size of each anchor node is calculated by running the first two steps of the basic DV-Hop algorithm throughout the network. Then, each anchor node transmits its actual location, Hop Size and the Hop Count information to a central location. At the central location it simulates the distribution of anchors and the position of each anchor node is estimated using the remaining anchor nodes using the basic DV-Hop algorithm.

Let $(\hat{x}_i, \hat{y}_i)$ be the estimated location of anchor node i to be recalculated as an unknown node and (x_j, y_j) be the known location of anchor node j. $\hat{d}_{ij}$ is the distance between $(\hat{x}_i, \hat{y}_i)$ and (x_j, y_j). Then;

$$
\begin{gathered}
(\hat{x}_i - x_1)^2 + (\hat{y}_i - y_1)^2 = \hat{d}_{i1}^2 \\
(\hat{x}_i - x_2)^2 + (\hat{y}_i - y_2)^2 = \hat{d}_{i2}^2 \\
\vdots \\
(\hat{x}_i - x_{i-1})^2 + (\hat{y}_i - y_{i-1})^2 = \hat{d}_{i(i-1)}^2 \\
(\hat{x}_i - x_{i+1})^2 + (\hat{y}_i - y_{i+1})^2 = \hat{d}_{i(i+1)}^2 \\
\vdots \\
(\hat{x}_i - x_{n-1})^2 + (\hat{y}_i - y_{n-1})^2 = \hat{d}_{i(n-1)}^2
\end{gathered}
\tag{4.1}
$$

n is the number of anchors in the network.

Equation (4.1) is an adaptation of (2.15) considering the i^{th} anchor node as an unknown node. In (4.1), we use the Average Hop Size of the network instead of different hop sizes for each anchor when estimating the distances $\hat{d}_{ik}$.

$$
Average\ Hop\ Size(AHS) = \frac{\sum_{i=1}^{n-1} Hop\ Size_i}{n-1} \tag{4.2}
$$

where $Hop\ Size_i$ is the Hop Size of i^{th} anchor and n is the number of anchors in the network.

Coordinates of the anchor node i can be recalculated using the following matrix operation according to (2.15).

$$A = -2 \times \begin{bmatrix} x_1 - x_{n-1} & y_1 - y_{n-1} \\ x_2 - x_{n-1} & y_2 - y_{n-1} \\ \vdots & \vdots \\ x_{i-1} - x_{n-1} & y_{i-1} - y_{n-1} \\ x_{i+1} - x_{n-1} & y_{i+1} - y_{n-1} \\ \vdots & \vdots \\ x_{n-2} - x_{n-1} & y_{n-2} - y_{n-1} \end{bmatrix} \tag{4.3}$$

according to (2.16);

$$B = \begin{bmatrix} \hat{d}_{i1}^2 - \hat{d}_{i(n-1)}^2 - x_1^2 + x_{n-1}^2 - y_1^2 + y_{n-1}^2 \\ \hat{d}_{i2}^2 - \hat{d}_{i(n-1)}^2 - x_2^2 + x_{n-1}^2 - y_2^2 + y_{n-1}^2 \\ \vdots \\ \hat{d}_{i(n-2)}^2 - \hat{d}_{i(n-1)}^2 - x_{n-2}^2 + x_{n-1}^2 - y_{n-2}^2 + y_{n-1}^2 \end{bmatrix} \tag{4.4}$$

from (2.17);

$$\hat{P} = \begin{bmatrix} \hat{x}_i \\ \hat{y}_i \end{bmatrix} = (A^T A)^{-1} A^T B \tag{4.5}$$

The localization error of anchor i is calculated as follows.

$$e_{ai} = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \tag{4.6}$$

The average anchor position errors is given by,

$$e_{a,avg} = \frac{\sum_{i=1}^{n} e_{ai}}{n} \tag{4.7}$$

4.1.3 Iterative Hop Size Correction (Step 3 of the Proposed Algorithm)

The estimation error changes with the Average Hop Size of the network. If ΔAHS is the Hop Size Correction to the initial Average Hop Size given by (4.2), Figure 4.4 shows the variation of $e_{a,avg}$. The optimum Hop Size Correction is computed using the line search algorithm.

The optimum Hop Size Correction value $\Delta AHS_{optimum}$ obtained as above is now used to modify the Hop Sizes of each anchor node and unknown node as follows,

$$Hop\,Size_{i,optimum} = Hop\,Size_i + \Delta AHS_{optimum} \tag{4.8}$$

$$d_{ij} = Hop\,Size_{i,optimum} \times Hop\,Count_{ij} \tag{4.9}$$

where $Hop\,Size_i$ is the Hop Size calculated in Step 1 of the proposed algorithm. Proposed Algorithm 1 includes centralized computations, and therefore, removes some of the flexibility of the original DV-Hop algorithm. Therefore we propose the second algorithm described in the next Section which does not have any centralized computations.

4.2 Proposed Algorithm 2

4.2.1 Overview

This algorithm uses the known distance between the anchors to improve the Hop Size of each anchor node. The proposed algorithm can be divided in to the following four steps.

Step 1: Hop Size and Hop Count of the anchor nodes are found using Step 1 and Step 2 of the DV-Hop algorithm.

Step 2: Known distances among the anchors are recalculated using estimated Hop Sizes and Hop Counts.

Step 3: Step 2 is iterated by modifying the Hop Size to minimize the difference between the actual distance and the estimated distance among anchors. Through this, the optimum Hop Size is obtained.

Step 4: Locations of unknown nodes are estimated with Step 3 of the DV-Hop algorithm using the modified Hop Size.

The algorithm is illustrated in Figure 4.2. This algorithm maintains the distributed nature of the original DV-Hop algorithm.

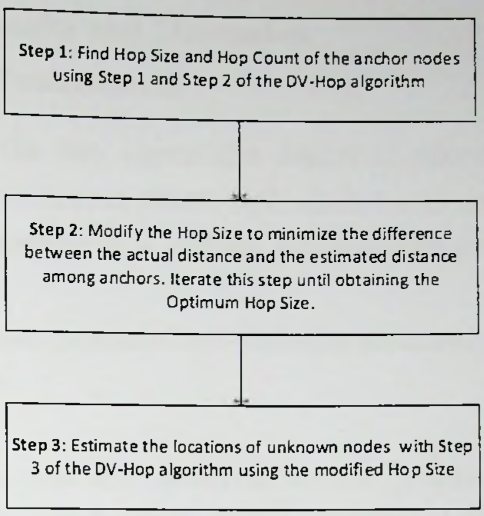


Figure 4.2: Algorithm 2

4.2.2 Recalculating the Distance between the Anchors

First the Hop Size of each anchor node is calculated through the first two steps of the DV-Hop algorithm throughout the network. Then the actual and estimated distance between anchors are calculated as follows.

Distance between anchor node i and j;

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4.10}$$

Estimated Distance between anchor node i and j;

$$\hat{D}_{ij} = HopSize_i \times hop_{ij} \tag{4.11}$$

Difference;

$$\Delta_{ij} = |D_{ij} - \hat{D}_{ij}| \tag{4.12}$$

$$Error_i = \sum_{j=1}^{N} \Delta_{ij} \tag{4.13}$$

Optimum Hop Size of i^{th} anchor, $HopSize_{i,optimum}$ is found by reducing the $Error_i$ using the Line Search Algorithm. This optimum Hop Size is used by unknown nodes to find their locations.

4.3 Simulation Results and Discussion

4.3.1 Simulation Environment

The performance of the two algorithms described above are evaluated using Matlab simulation environment where both anchor and unknown nodes are distributed randomly according to a uniform pdf as shown in Figure 3.1.

All the simulation results presented are averages obtained over 100 different sensor bed configurations.

Figure 4.3shows the actual and estimated anchor positions obtained at Step 2 of our Algorithms. The recalculated anchor positions show significant deviation form the actual positions.

Figure 4.4 shows the variation of the average anchor position error with the

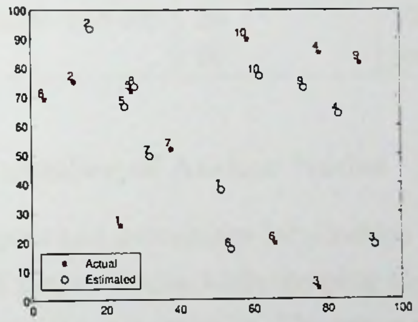


Figure 4.3: Actual and Estimated locations of anchors

Hop Size Correction. An optimum value for the Hop Size Correction is clearly observed in Figure 4.4.

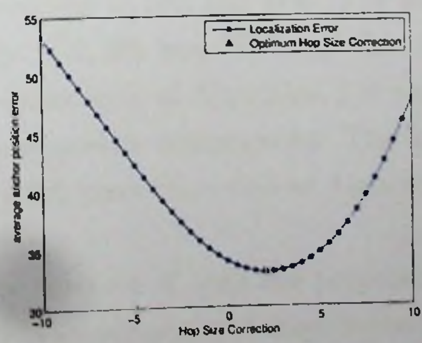


Figure 4.4: Optimum Hop Size Correction for AHS computed in initial step

4.4 Performance Evaluation

Simulations are carried out to first examine the performance of the proposed algorithm compared to the original DV-Hop. A scenario of 200 nodes, each having a radio range of $22m$ in a $100m \times 100m$ area is used. The algorithm is studied for its performance by varying one at a time, the anchor ratio, the total number of nodes and the radio range of the nodes as summarized in Table 5.1. The localization error and the error variance are compared with the original DV-Hop algorithm [9] and two other algorithms DV-Hop Weighted [20], DV-Hop Marker [35]. Finally, the new algorithms are compared for their communications and computational costs with the same prior studies.

Table 4.1: Simulation Instances

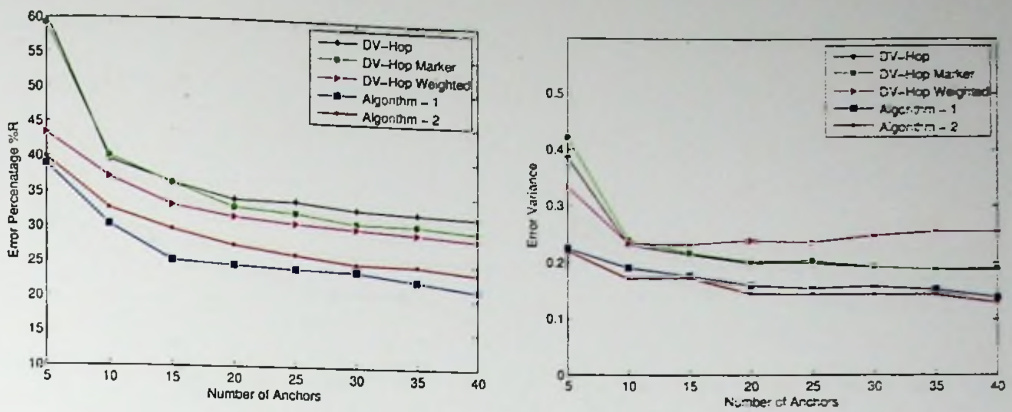
Figure No	Total Number of Nodes	Anchor Ratio (%)	Anchor Range(m)
4.5	200	variable 5-40	22
4.6	variable 100-400	20	22
4.7	200	20	variable 15-40

4.4.1 With variable number of Anchor Nodes

Percentage localization error and percentage localization error variance are shown with different number of anchor nodes while keeping the total number of nodes at 200 in Figure 4.5(a) and (b) respectively. The radio range of a sensor node is set to $22m$.

It is evident that both the proposed algorithms have better localization accuracy compared to the basic DV-Hop algorithm [9] as well as the previously published improved algorithms in [20] and [35]. The localization error of Algorithm 1 is approximately 11%, 9% and 7% lower than that of [9], [35] and [20] respectively. The localization error of Algorithm 2 is approximately 9%, 7% and 5% lower than the same respective comparisons. The localization error of Algorithm 1 is approximately 2% lower than that of Algorithm 2.

The localization error variance of both the proposed algorithms also show a clear improvement. The improvement obtained beyond 15 anchor nodes is observed to be insignificant. This corresponds to an Anchor Ratio (AR) of 7.5%.



(a) (b)

Figure 4.5: (a)Error percentage with different number of anchors (b)Error variance with different number of anchors (N=200,R=22m)

4.4.2 With variable Total number of Nodes

Percentage localization error and percentage localization error variance are shown with different total number of nodes while keeping the number of anchor nodes at 20 (10% AR) in Figure 4.6(a) and (b). The radio range of a sensor node is set to 22m.

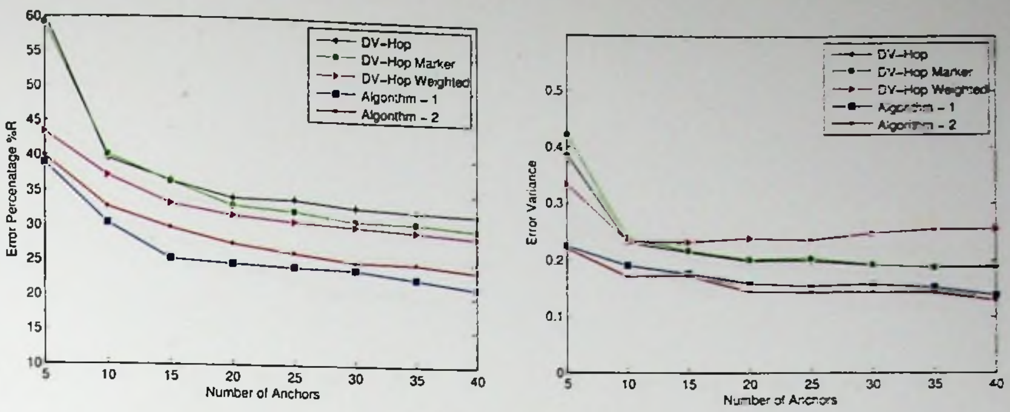
It is evident that both the proposed algorithms have improved localization accuracy compared to [9], [20] and [35]. Localization error of the Algorithm 1 is approximately 14%, 12% and 10% lower than that of [9], [35] and [20] respectively. The localization error of Algorithm 2 is approximately 8%, 5% and 4% lower than the same respective comparisons. The localization error of Algorithm 1 is approximately 8% lower than that of Algorithm 2.

Localization error variance is also clearly lower in both the proposed algorithms in comparison with [9], [20] and [35].

4.4.3 With variable Radio Range

Percentage localization error and percentage localization error variance are shown with different total radio ranges while keeping the total number nodes at 200 in Figure 4.7(a) and (b). Number of anchor nodes is 20 (10% AR).

It is evident that both the proposed algorithms have improved localization accuracy compared to [9], [20] and [35]. Localization error of the Algorithm 1



(a)

(b)

Figure 4.5: (a)Error percentage with different number of anchors (b)Error variance with different number of anchors (N=200,R=22m)

4.4.2 With variable Total number of Nodes

Percentage localization error and percentage localization error variance are shown with different total number of nodes while keeping the number of anchor nodes at 20 (10% AR) in Figure 4.6(a) and (b). The radio range of a sensor node is set to 22m.

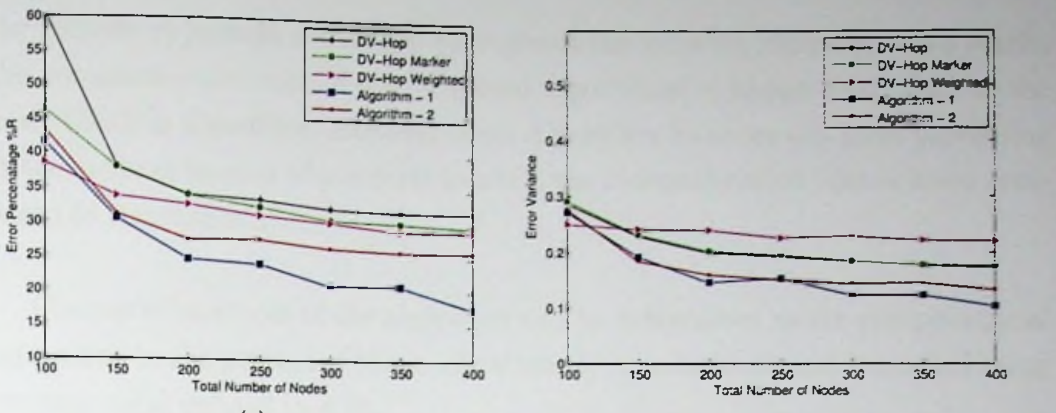
It is evident that both the proposed algorithms have improved localization accuracy compared to [9], [20] and [35]. Localization error of the Algorithm 1 is approximately 14%, 12% and 10% lower than that of [9], [35] and [20] respectively. The localization error of Algorithm 2 is approximately 8%, 5% and 4% lower than the same respective comparisons. The localization error of Algorithm 1 is approximately 8% lower than that of Algorithm 2.

Localization error variance is also clearly lower in both the proposed algorithms in comparison with [9], [20] and [35].

4.4.3 With variable Radio Range

Percentage localization error and percentage localization error variance are shown with different total radio ranges while keeping the total number nodes at 200 in Figure 4.7(a) and (b). Number of anchor nodes is 20 (10% AR).

It is evident that both the proposed algorithms have improved localization accuracy compared to [9], [20] and [35]. Localization error of the Algorithm 1

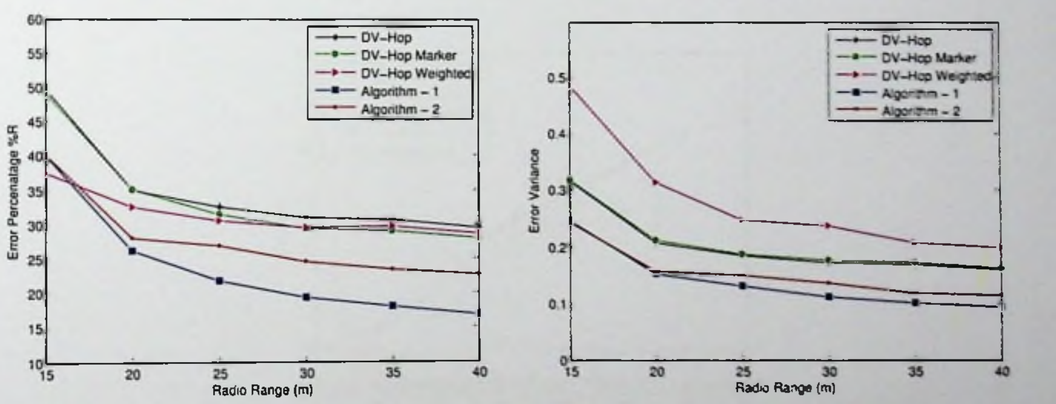


(a) (b)

Figure 4.6: (a)Error percentage with different total number of nodes (b)Error variance with different total number of nodes (AR=10%,R=22m)

is approximately 14%, 13% and 12% lower than that of [9], [35] and [20] respectively. The localization error of Algorithm 2 is approximately 8%, 7% and 6% lower than the same respective comparisons. The localization error of Algorithm 1 is approximately 6% lower than that of Algorithm 2.

Localization error variance is also clearly lower in both the proposed algorithms in comparison with [9], [20] and [35].



(a) (b)

Figure 4.7: (a)Error percentage with different radio range (b)Error variance with different radio range (N=200,A=20)

4.4.4 Communication and Computational Cost

Communication cost of the algorithm is determined by the number of packets exchanged inside the network in the process of localization. In both the proposed algorithms, an optimized Hop Size is found using an iterative method. Therefore the number of packets exchanged throughout the network increases. As a result, the communication cost of the proposed algorithms is higher than that of the basic DV-Hop algorithm. However since Algorithm 2 carries out local processing in the anchors instead of a central location the communication cost is lower compared to Algorithm 1.

Computational cost of the algorithm can be determined as the computational time taken in the process of the localization. Computational cost is studied using the time ratio defined in 4.14.

$$TimeRatio = \frac{LocalizationTime_{ProposedAlgorithm}}{LocalizationTime_{DV-HopAlgorithm}} \tag{4.14}$$

Due to the iterations, the proposed algorithms have higher computational cost than the DV-Hop algorithm as illustrated in Figure 4.8. While the computational cost of Algorithm 1 increases drastically with increasing anchor ratio, the same of Algorithm 2 is significantly lower in comparison. Comparing with Figures 4.5, 4.6 and 4.7, it is seen that the lower computational cost of Algorithm 2 is obtained with a sacrifice in performance. The increase in computational cost of Algorithm 2 in comparison to [9], [35] and [20] is approximately 4%.

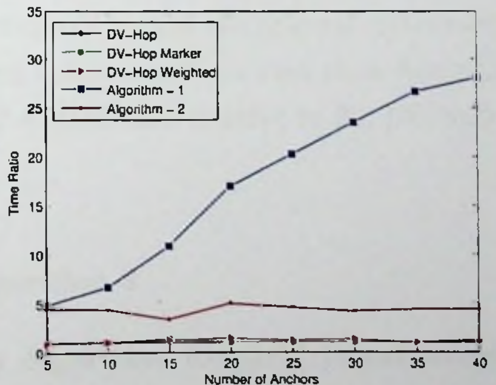


Figure 4.8: Time Ratio with different number of anchors

4.4.5 Conclusion

The two algorithms proposed in this Chapter are based on using anchor position re-estimation in order to obtain a better estimate of the Hop Size. Simulation results show that both the proposed algorithms provide improvements over the DV-Hop algorithm [9] and the improved versions presented in [20] and [35]. A performance comparison is provided in summary form in Table 4.2. Localization error variance also is lower in both the proposed algorithms, and thus they ensure more steady performance.

Table 4.2: Performance Comparison of Algorithms

Algorithm	Localization error as a percentage of R			
	Case 1 N=200 n=20 R=22m	**Case 2** N=200 n=30 R=22m	**Case 3** N=300 n=20 R=22m	**Case 4** N=200 n=20 R=40m
DV-Hop [9]	34.1166	32.8638	32.3024	29.5040
DV-Hop Marker [35]	32.8907	30.8675	30.5428	27.8509
DV-Hop Weighted [20]	31.5418	30.1692	30.1867	28.6347
Algorithm 1	24.5055	23.8068	20.9430	16.8087
Algorithm 2	27.5249	25.0581	26.4259	22.5786

The optimum Hop Size computation requires an iterative procedure. Thus, both algorithms need more computational power and time than the DV-Hop algorithms used for comparison. Also, the computational cost of Algorithm 1 increases with the number of anchor nodes.

Algorithm 1 introduces a centralized processing component into the DV-Hop algorithm, while Algorithm 2 retains the original distributed nature. Thus, Algorithm 1 incurs a higher communications cost than Algorithm 2. The increase in computational cost of Algorithm 2 relative to [9], [20] and [35] is approximately 4%.

4.5 Proposed Algorithm 3

Improved DV-Hop Algorithm for Grid based sensor networks

4.5.1 Overview

In this Section we propose a DV-Hop based localization scheme that can be used for localizing grid based sensor networks. We assume that the sensor network is deployed in a controlled manner, where the sensors are fixed randomly on a regular grid as shown in Figure 4.9. Furthermore we assume that grid distance (D) is known and radio range (R) of each sensor node is $D < R < 2D$.

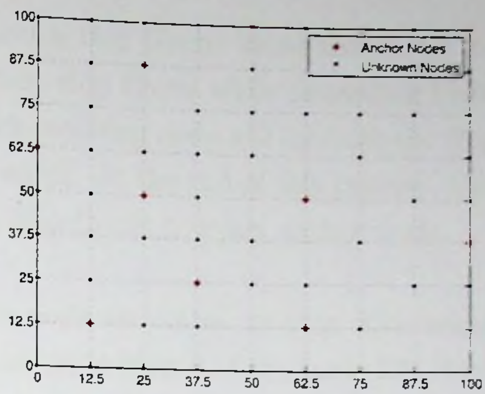


Figure 4.9: Sensors deployed on a grid in a random manner

Since all the nodes are placed on grid points, the distance between an anchor and an unknown node should be a known multiple of the grid distance D. As shown in Figure 4.11,

$$d_{ij} = one \quad of\{1, \sqrt{2}, 2, \sqrt{5}, \sqrt{8}, 3, \sqrt{10}, \sqrt{13}, \sqrt{18}, 4, \sqrt{17}, ...\} \times D \tag{4.15}$$

where d_{ij} is the distance between anchor node i and unknown node j. We use this knowledge about d_{ij} to improve the performance of DV-Hop algorithm in grid based networks.

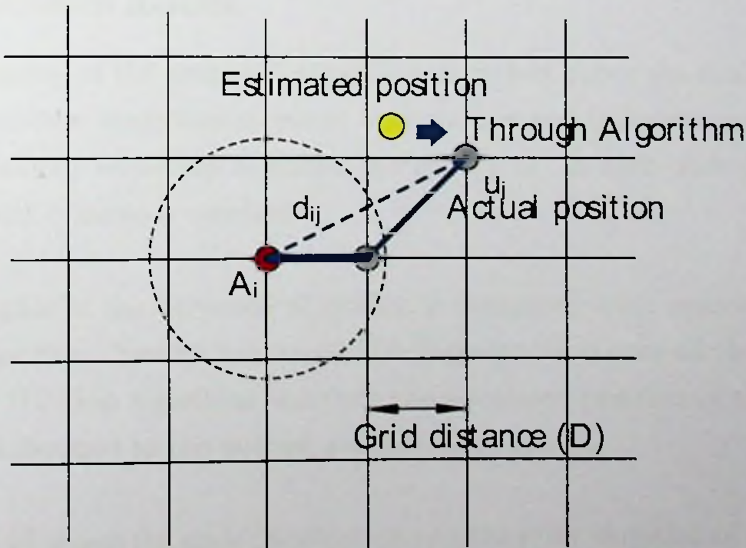


Figure 4.10: Grid based DV-Hop algorithm

The algorithm runs in three major steps as follows.

Step 1: Each anchor node broadcasts a packet throughout the network containing its location and a Hop Count value initialized to one. Each receiving node keeps the minimum Hop Count while discarding higher ones from a particular anchor node. Each receiving node will increase the Hop Count by one before passing the packet onwards. At the end of this process, each node in the network obtains the minimum Hop Count to every anchor node.

Step 2: Each anchor node estimates its Hop Size using the Hop Count values to the other anchors as in Step 2 of the basic DV-Hop algorithm in 2.14and broadcasts it to the network. Unknown nodes save the first value they receive, while transmitting this Hop Size to the neighbors. This mechanism results in most of the unknown nodes obtaining the Hop Size of the nearest anchor. At the end of this process, unknown nodes estimate the distance form each anchor as the product of Hop Size and the corresponding Hop Count. The resulted distance between an anchor and an unknown node will be rounded to the nearest value of 4.15.

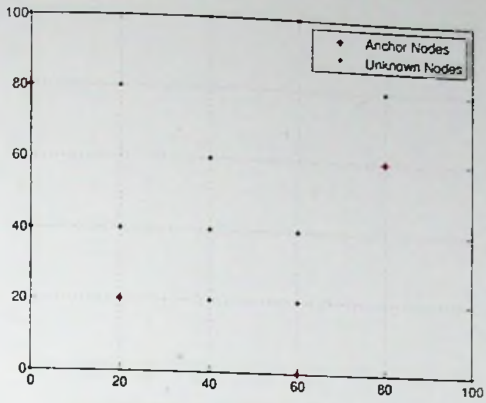
Step 3: Unknown nodes estimate their locations using either trilateration / multilateration or maximum likelihood estimation after they receive three or more distance information as in Step 3 of basic DV-Hop algorithm.

4.5.2 Simulation Results

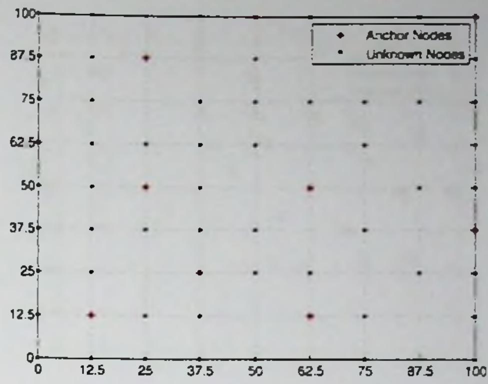
The performance of the proposed algorithm described above are evaluated using Matlab simulation environment where both anchor and unknown nodes are distributed randomly according as shown in Figure 4.11. In each node distribution, anchor density remains a constant.

Performance of the proposed algorithm is compared with nearest grid point DV-Hop algorithm. Nearest grid point DV-Hop algorithm runs all the three steps of the basic DV-Hop algorithm and then the estimated position of the unknown node will be dragged to the nearest available grid point.

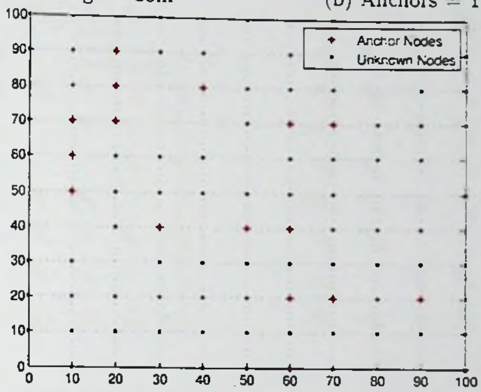
Figure 4.12 shows the node distribution and the error variation of a grid based WSN.



(a) Anchors = 5, Radio Range = 30m



(b) Anchors = 10, Radio Range = 20m



(c) Anchors = 20, Radio Range = 15m

Figure 4.11: (a)25 nodes on a grid (b)64 nodes on a grid (c)100 nodes on a grid

4.5.3 Discussion

Through simulations it is very clear that the proposed algorithm can localize more nodes with zero localization error. Figure 4.13 shows the summary of the simulation results gathered for various grid based topologies.

In our WSN based application, if the controlled node distribution is possible on a even grid and the grid distance is known, the proposed algorithm can be used to localize the unknown nodes of the network.

4.6 Localization in Emergency Environments

In this research, we tried to overcome challenges for localization mentioned in Section 1.4 in an emergency situation by improving, modifying the DV-Hop algorithm. First we improve the accuracy of the DV-Hop algorithm while proposing three novel enhancements to the original DV-Hop algorithm. The effect of link failures, node failures and radio range irregularity on DV-Hop localization accu-

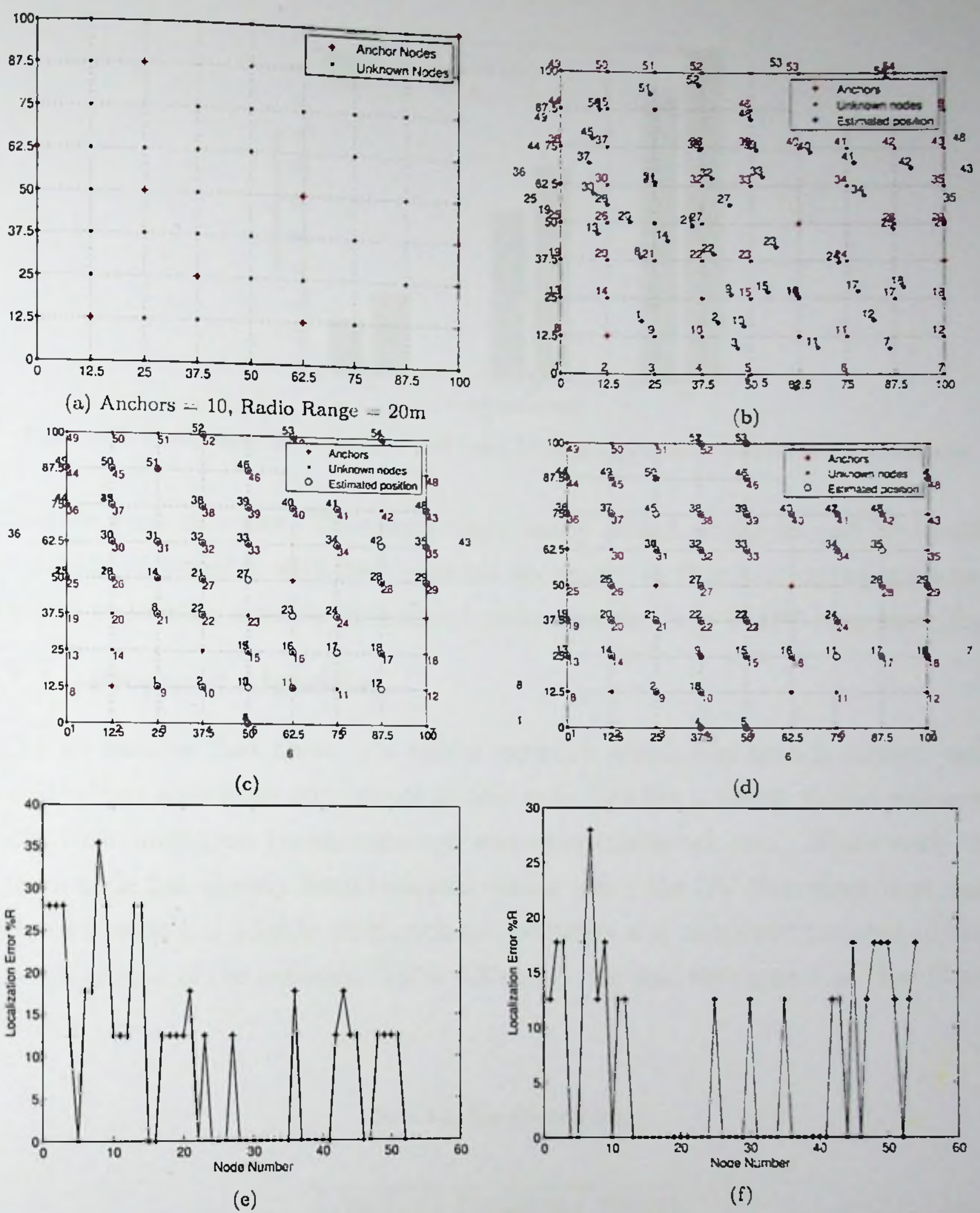


Figure 4.12: (a)64 nodes on a grid (b)DV-Hop algorithm (c)Nearest grid point DV-Hop algorithm (d)Proposed DV-Hop based algorithm (e)Localization error of each node when Nearest grid point DV-Hop algorithm used (f)Localization error of each node when proposed DV-Hop based algorithm used

racy is studied. Now we propose a new DV-Hop based algorithm to locate newly introduced nodes to an existing sensor network.

4.7 Introduction of New Nodes to the Network

Introducing new sensor nodes to the existing network is a common requirement in WSN related applications. Especially nodes with unknown locations can be newly added randomly for additional information gathering during an emergency

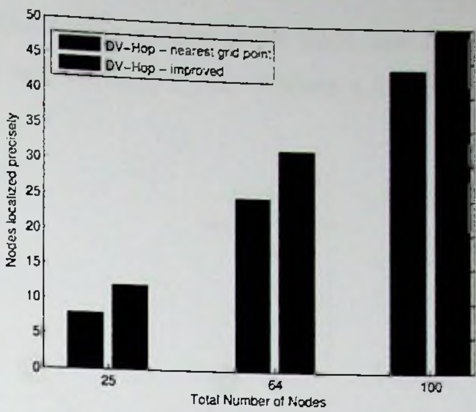


Figure 4.13: Performance of the proposed grid based DV-Hop algorithm in different node distributions

situation such as a fire. Therefore these newly added nodes should be localize as quickly as possible with best possible accuracy. In this Section we propose a method to localize a newly introduced node with the help of DV-Hop algorithm.

4.7.1 Proposed Algorithm

First we assume that there is a sensor network which has been localized using the DV-Hop algorithm and our objective is to localize a newly added unknown node with minimum communication and computational cost. Since each unknown node has already been localized earlier using the DV-Hop algorithm, each unknown node has a table with anchors' locations and minimum number of hops to each anchor of the network. Table 4.3 shows this and we name it as Hop Count Table.

Table 4.3: Hop Count Table

Anchor	Location	Hops
A_1	(x_1, y_1)	h_1
A_2	(x_2, y_2)	h_2
$\vdots$	$\vdots$	$\vdots$
A_n	(x_n, y_n)	h_n

Once a new node is introduced to the network, first it sends a packet to its neighbors (one hop away) asking the Hop Count Table. Newly introduced node keeps the first received Hop Count Table while discarding others unless its from an anchor. This mechanism ensures that the Hop Count Table of the nearest neighbor will get to the newly added node. The newly added node will save the Hop Count Table while adding one to the hop count column. If an anchor is a

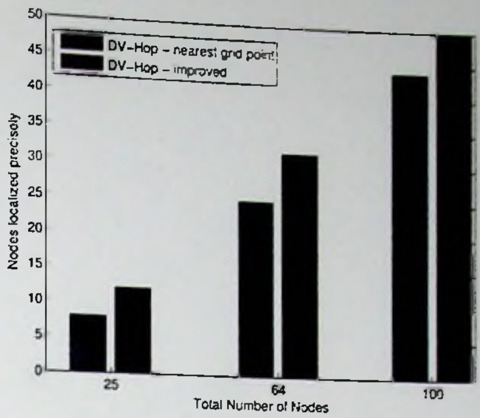


Figure 4.13: Performance of the proposed grid based DV-Hop algorithm in different node distributions

situation such as a fire. Therefore these newly added nodes should be localize as quickly as possible with best possible accuracy. In this Section we propose a method to localize a newly introduced node with the help of DV-Hop algorithm.

4.7.1 Proposed Algorithm

First we assume that there is a sensor network which has been localized using the DV-Hop algorithm and our objective is to localize a newly added unknown node with minimum communication and computational cost. Since each unknown node has already been localized earlier using the DV-Hop algorithm, each unknown node has a table with anchors' locations and minimum number of hops to each anchor of the network. Table 4.3 shows this and we name it as Hop Count Table.

Table 4.3: Hop Count Table

Anchor	Location	Hops
A_1	(x_1, y_1)	h_1
A_2	(x_2, y_2)	h_2
$\vdots$	$\vdots$	$\vdots$
A_n	(x_n, y_n)	h_n

Once a new node is introduced to the network, first it sends a packet to its neighbors (one hop away) asking the Hop Count Table. Newly introduced node keeps the first received Hop Count Table while discarding others unless its from an anchor. This mechanism ensures that the Hop Count Table of the nearest neighbor will get to the newly added node. The newly added node will save the Hop Count Table while adding one to the hop count column. If an anchor is a neighboring node to the newly added node then the hop count column of that respective anchor will not be added one. Figure 4.14 illustrates this algorithm.

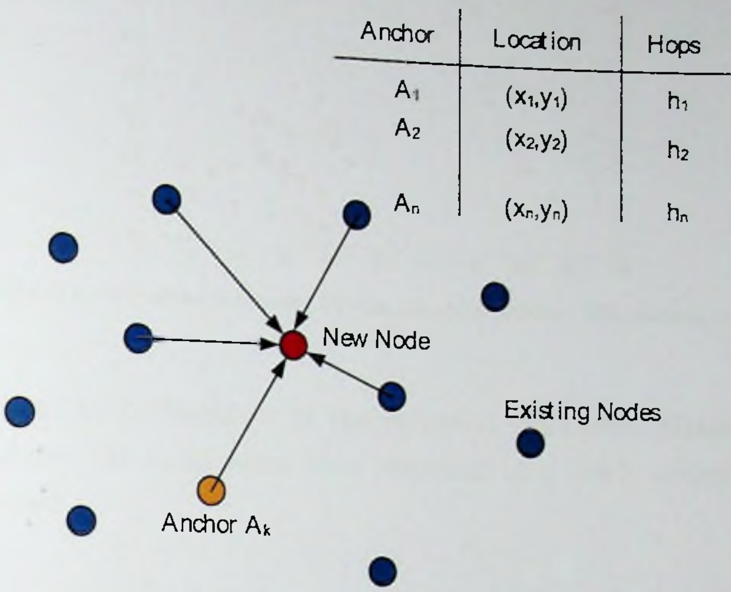


Figure 4.14: Localizing a newly added node

Table 4.4 shows the saved Hop Count Table of the newly added node.

Table 4.4: Hop Count Table

Anchor	Location	Hops
A_1	(x_1, y_1)	$h_1 + 1$
A_2	(x_2, y_2)	$h_2 + 1$
$\vdots$	$\vdots$	$\vdots$
A_k	(x_k, y_k)	h_k
$\vdots$	$\vdots$	$\vdots$
A_n	(x_n, y_n)	$h_n + 1$

Once the Hop Count Table is finalized, the newly added node runs the third step of the basic DV-Hop algorithm to estimate its location.

4.7.2 Simulation Results

Figure 4.15 shows a network with a newly introduced node. Localization error of this newly introduced node is needed to be analyzed.

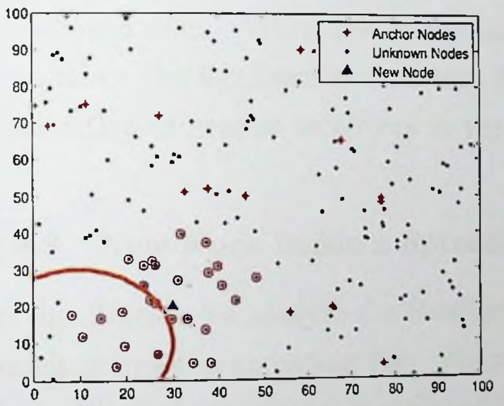
When analyzing the results of the 4.16,it is very clear that the localization error of the newly introduced node goes in line with the localization error of the existing network. On the other hand, the proposed mechanism for localizing a new node is very much cost effective in terms of computational and communication vise over re-localizing the whole network.

4.8 Introducing a New Node in an Emergency Situation

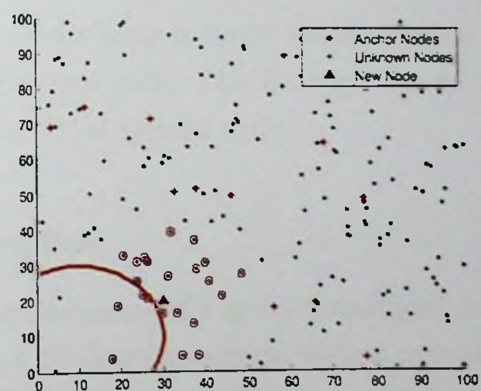
In this Section we introduce a new node in an emergency situation where some of the existing nodes are destroyed due to a spreading fire. Through simulations we analyze the performance of the proposed algorithm for localizing the newly introduced node in a node destroying environment.

4.8.1 New Node outside the Fire

Figure 4.17 shows a scenario with a fire outside the newly added node. Due to the fire several nodes have been destroyed.



(a) Before fire



(b) After fire

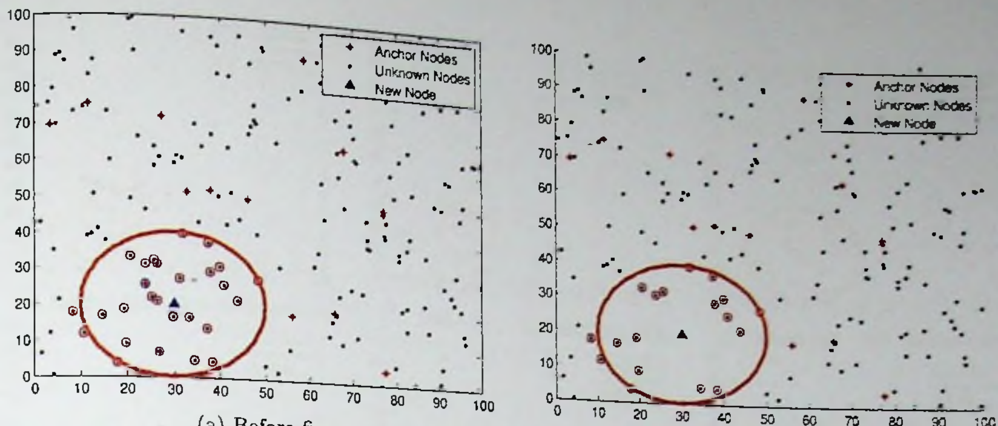
Figure 4.17: Localizing a new node in a node destroying environment (Nodes = 200, Radio Range = 22m)

4.8.2 New Node inside the Fire

Figure 4.18 shows a scenario of localizing a new node which is inside a fire. Due to the fire several nodes have been destroyed.

Table 4.5 shows the localization error variation in these two scenarios.

According to the results it is clear that when the new node is outside the fire, node destruction due to fire will not affect the localization accuracy of the newly



(a) Before fire

(b) After fire

Figure 4.18: Localizing a new node in a node destroying environment (Nodes = 200, Radio Range = 22m)

Table 4.5: Localization accuracy of a new node in a node destroying environment

	Actual	No fire	Outside fire Figure 4.17	Inside fire Figure 4.18
Location	(30,20)	(31.26, 26.03)	(31.26, 26.03)	(35.41,26.28)
Error %R	-	28.03	28.03	37.70

introduced node. When the newly introduced node is inside the fire region, it may affect the localization accuracy of the new node. This happens due to the destruction of nearest neighbors to the newly introduced node.

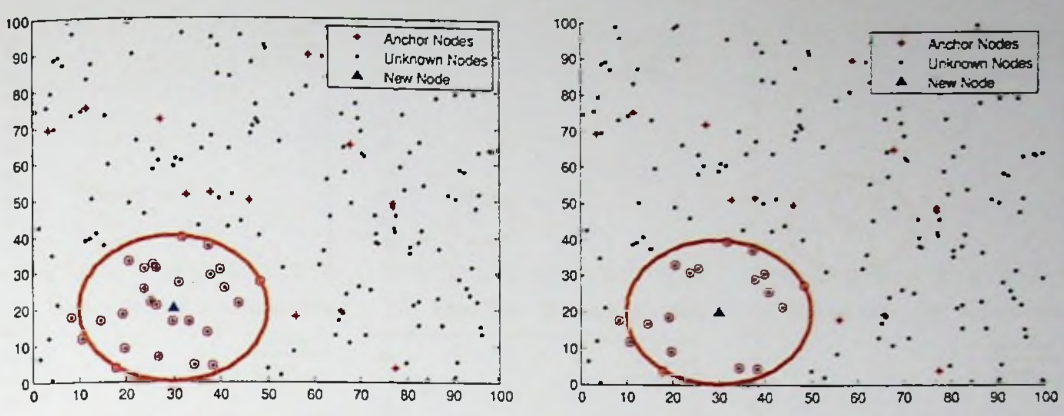
4.8.3 New Node inside a Spreading Fire

In this Section we analyze the localization accuracy of a newly introduced node which is inside a spreading fire. Figure 4.19 shows a scenario with a spreading fire until all the neighboring nodes of the newly introduced node are destroyed.

Table 4.6 shows the percentage localization error of a newly introduced node according to the magnitude of a fire.

4.8.4 Discussion

In the above two sections we discussed regarding a novel mechanism for localizing a newly introduced node to an existing network, with the help of DV-Hop algorithm. After analyzing the simulation results, it is very clear that the proposed algorithm can localize a newly introduced node will the same localization accuracy as the existing network. Furthermore, a scenario of an emergency situation



(a) Before fire (b) After fire

Figure 4.18: Localizing a new node in a node destroying environment (Nodes = 200, Radio Range = 22m)

Table 4.5: Localization accuracy of a new node in a node destroying environment

	Actual	No fire	Outside fire Figure 4.17	Inside fire Figure 4.18
Location	(30,20)	(31.26, 26.03)	(31.26, 26.03)	(35.41,26.28)
Error %R	-	28.03	28.03	37.70

introduced node. When the newly introduced node is inside the fire region, it may affect the localization accuracy of the new node. This happens due to the destruction of nearest neighbors to the newly introduced node.

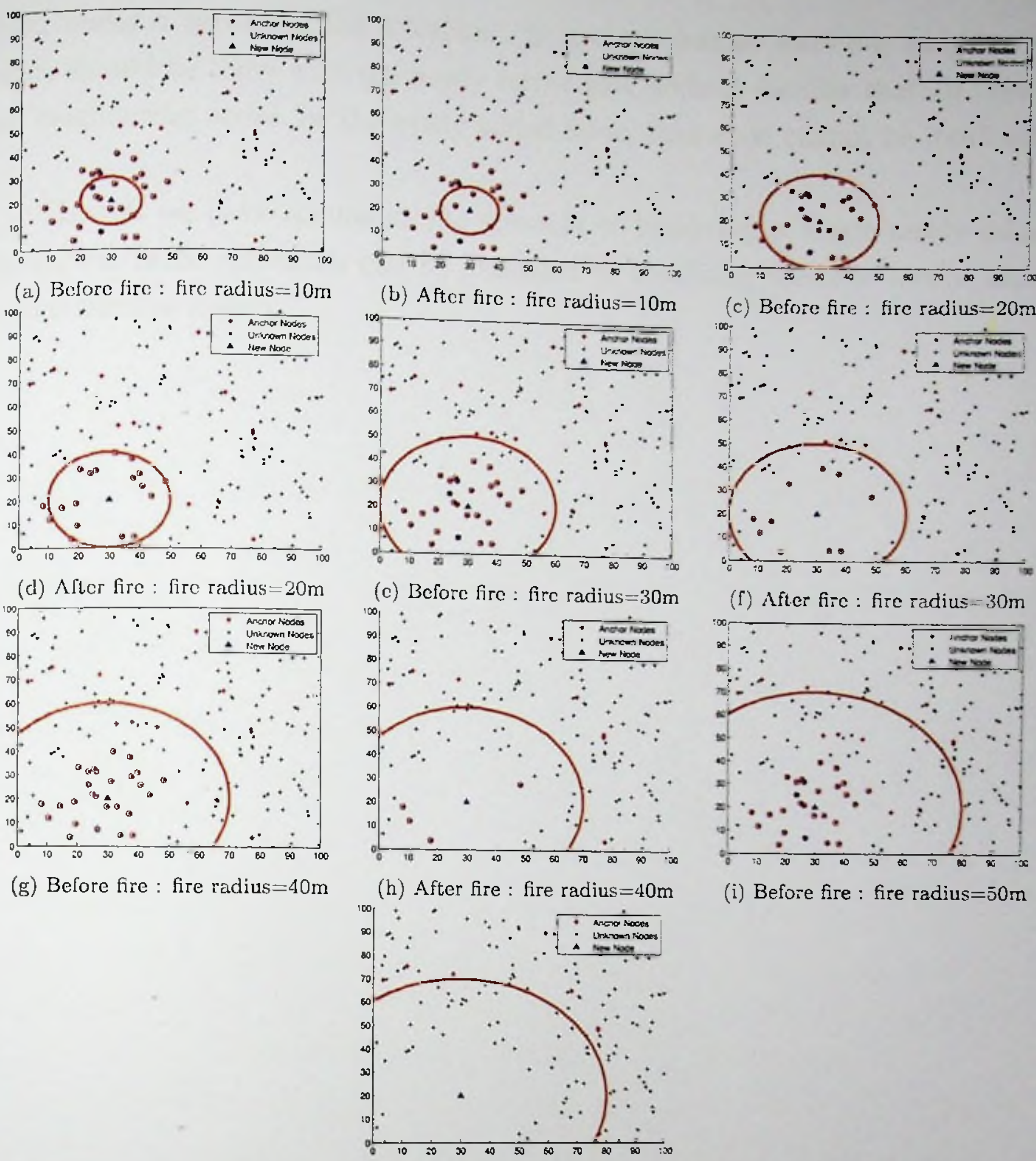
4.8.3 New Node inside a Spreading Fire

In this Section we analyze the localization accuracy of a newly introduced node which is inside a spreading fire. Figure 4.19 shows a scenario with a spreading fire until all the neighboring nodes of the newly introduced node are destroyed.

Table 4.6 shows the percentage localization error of a newly introduced node according to the magnitude of a fire.

4.8.4 Discussion

In the above two sections we discussed regarding a novel mechanism for localizing a newly introduced node to an existing network, with the help of DV-Hop algorithm. After analyzing the simulation results, it is very clear that the proposed algorithm can localize a newly introduced node will the same localization accuracy as the existing network. Furthermore, a scenario of an emergency situation



(a) Before fire : fire radius=10m
(b) After fire : fire radius=10m
(c) Before fire : fire radius=20m
(d) After fire : fire radius=20m
(e) Before fire : fire radius=30m
(f) After fire : fire radius=30m
(g) Before fire : fire radius=40m
(h) After fire : fire radius=40m
(i) Before fire : fire radius=50m
(j) After fire : fire radius=50m

Figure 4.19: Localizing a new node inside a spreading fire (Nodes = 200, Anchors = 20, Radio Range = 22m, New node = (30,20))

Table 4.6: Localization accuracy of a new node in a spreading fire (New node at (30,20))

Radius of Fire (m)	Estimated Location	Error %R
10	(31.26,26.03)	28.03
20	(35.41,26.28)	37.70
30	(35.41,26.28)	37.70
40	(35.41,26.28)	37.70
50	Unable to locate	-

was simulated and localization accuracy of a newly introduced node was evaluated under a spreading fire situation. It is clear that at least one neighboring node should be there with the newly introduced node to localize that. If there's no neighboring nodes for the newly added node, that node cannot be localized.

Though we have simulated the scenario of localizing a single newly added node, the same algorithm can be extended to localizing multiple newly added nodes simultaneously.

Chapter 5

TARGET TRACKING

5.1 Introduction

The objective of this Chapter is to present a novel tracking algorithm based on the DV-Hop algorithm. In the previous two Chapters we discussed in depth the DV-Hop localization algorithm and its improvements. In this Chapter we implement the target tracking functionality in an existing network which is already localized. Later in the Chapter we propose a novel tracking algorithm which combines the DV-Hop algorithm with Kalman filtering. The DV-Hop algorithm is used for pre-localization of the target and measurement conversion. Then the Kalman filter is used to recursively update the target state.

Before presenting the novel tracking algorithm we first implement an existing target tracking algorithm [61] by particle filtering in a binary sensor network. In a binary sensor network, the deployed sensors measure a signal of interest and if its level is above a predefined threshold, they report to the fusion center with a signal that identifies them; otherwise they are silent. Therefore, binary sensors do not need much internal processing. However this is a centralized algorithm. We use this algorithm as a reference to compare the performance with the proposed novel tracking algorithm. Finally we compare the performances of the algorithms using Matlab based simulations.

5.2 Target Tracking by Particle Filtering in Binary Sensor Networks

Djuric *et. al* [61] present a particle filtering algorithm for tracking a single target using data from binary sensors. The sensors transmit signals that identify them to a central unit if the target is in their neighborhood; otherwise they do not transmit anything. The central unit uses a model for the target movement in the sensor field and estimates the target's trajectory, velocity, and power using the received data. In Section 5.2.1 we introduce and implement this algorithm.

Simulation results show the performance of the implemented algorithm.

5.2.1 Network Description

In a binary sensor network, the deployed sensors measure a signal of interest and if its level is above a predefined threshold, they report to the fusion center with a signal that identifies them; otherwise they are silent.

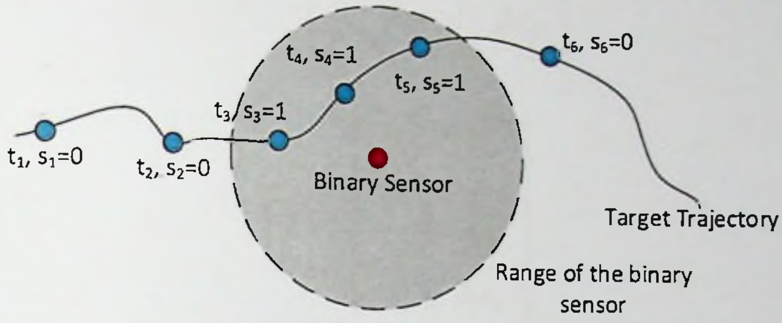


Figure 5.1: A binary sensor with a moving target

Figure 5.1 shows a binary sensor having a circular coverage. When the target is outside the range of the sensor, the received signal is below the set threshold. and the sensor does not transmit anything (instants t_1, t_2, t_6). During the time when the target is inside the range of the sensor, the received signal is above the threshold, and the sensor transmits a "one" to the fusion center (instants t_3, t_4, t_5). When at a given time the fusion center does not receive a signal from a particular sensor, this implies that the sensor transmits a "zero.".

The network consists of N binary sensors that may be deployed randomly, deterministically, or both. In all cases, we assume that the fusion center knows the locations of all the binary sensors and that their locations remain fixed for all time.

5.2.2 Mathematical Models

The standard model for target movement is described as follows.

$$X_k = [x_k \quad \dot{x}_k \quad y_k \quad \dot{y}_k] \tag{5.1}$$

where (x_k, y_k) are the position of the target at time t_k and $(\dot{x}_k, \dot{y}_k)$ are the velocities of the target along the x and y directions at time t_k respectively.

From the nearly constant-velocity model [64];

$$X_{k+1} = F_k X_k + G_k u_k \tag{5.2}$$

where

$$F_k = \begin{bmatrix} 1 & \Delta t_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

$$G_k = \begin{bmatrix} \frac{\Delta t_k^2}{2} & 0 \\ \Delta t_k & 0 \\ 0 & \frac{\Delta t_k^2}{2} \\ 0 & \Delta t_k \end{bmatrix} \tag{5.4}$$

where $\Delta t_k = t_{k+1} - t_k$ is the sampling time, $u_k = [u_x \quad u_y]^T$ is a white Gaussian noise with zero mean and covariance matrix Q_u. u_x and u_y represent the effect of noise acceleration of the moving target along the x and y axes respectively. Furthermore we assume that u_x and u_y are uncorrelated. i.e.,

$$Q_u = \begin{bmatrix} \sigma_{u_x}^2 & 0 \\ 0 & \sigma_{u_y}^2 \end{bmatrix} \tag{5.5}$$

The received power can be modeled for the n^{th} sensor as follows.

$$y_{n,k} = g_n(X_k) + v_{n,k} = \frac{\psi d_0^{\alpha}}{\|r_n - l_k\|^{\alpha}} + v_{n.k} \tag{5.6}$$

$n = 1, 2, \ldots\ldots, N$

where,

- N is the total number of sensors.
- $g_n(\cdot)$ is a function that models the received signal power by the n^{th} sensor.
- $v_{n,k}$ is a noise process independent from u_k and independent from noise samples of other sensors.
- r_n is the position of the n^{th} sensor.

- $l_k = [x_k \quad y_k]^T$ is the location of the target at time t_k.
- $\|r_n - l_k\|^\alpha$ denotes the Euclidean distance between r_n and l_k.
- ψ is the emitted power of the target measured at a reference distance d_0.
- α is an attenuation parameter that depends on the transmission medium and is considered to be known and the same for all sensors.

In the model, it is assumed that $v_{n,k} \sim N(\mu_v, \sigma_v^2)$ where $\mu_v = \sigma^2$ with σ^2 being the known power of the background measurement noise of one sample and $\sigma_v^2 = \frac{2\sigma^4}{L}$, with L being the number of samples used to obtain the measured power.

The n^{th} sensor measures the received power $y_{n,k}$, processes it locally and transmits a single binary digit to the fusion center according to the following rule.

1. The sensor compares the actual observed power level $y_{n,k}$ with a threshold γ. If the sensed value is below γ, it remains silent.
2. If the sensed value is greater than γ, the sensor transmits its identification code to the fusion center.

Therefore, the sensors in the network send signals to the fusion center only if the received power is greater than the sensor thresholds.

The received signal from the n^{th} sensor at the fusion center is modeled as follows.

$$z_{n,k} = \beta_n s_{n,k} + \epsilon_{n,k} \tag{5.7}$$

where

$$s_{n,k} = \left\{ \begin{array}{lll} 1 & if & y_{n,k} > \gamma \\ 0 & if & y_{n,k} \leq \gamma \end{array} \right\} \tag{5.8}$$

and $\epsilon_{n,k}$ is the observation noise, and β_n is a known attenuation coefficient associated with the n^{th} sensor.

In this analysis we neglect the observation noise. i.e.,

$$z_{n,k} = s_{n,k} \tag{5.9}$$

The objective is to track the evolving state $X_{0:k} = (X_0, X_1, ..., X_k)$ using the observations $Z_{1:k} = (z_{1,1:k}, ..., z_{N,1:k})$, where $z_{i,1:k}$ represents the observations up to time t_k of the i^{th} sensor.

5.2.3 Tracking Algorithm

In this Section we describe the target tracking algorithm associated with [61].

Recall that according to the theory of particle filtering, we track the a posteriori distribution of $X_{0:k}$, $p(X_{0:k}|Z_{1:k})$ by approximating it with a random measure χ_k composed of particles $x_k^{(m)}$ and weights $w_k^{(m)}$, where m is an index, and which we denote by $\chi_k = \{X_{0:k}^{(m)}, w_k^{(m)}\}_{m=1}^{M}$ with M being the number of particles. At every time instant t_k, the particle filter carries out the following operations [61]:

1. Selection of most promising particle streams
2. Particle propagation
3. Computation of particle weights
4. State estimation

Particle filter attempts to draw from an importance function which is as close as possible to the optimal one. To that end, the selection of most promising particles is carried out by sampling from a multinomial distribution where the number of possible outcomes is M and the probabilities of the respective outcomes are $\tilde{w}_t^{(m)}, m = 1, 2...., M$ and

$$\tilde{w}_k^{(m)} \propto p(Z_k|\mu_k^{(m)})w_{k-1}^{(m)} \tag{5.10}$$

where $\mu_k^{(m)}$ is some parameter that characterizes $X_k^{(m)}$ given $X_{k-1}^{(m)}$.
We have,

$$p(Z_k|\mu_k^{(m)}) = \prod_{n=1}^{N} p(z_{n,k}|\mu_k^{(m)}) \tag{5.11}$$

For the factors $p(z_{n,k}|\mu_k^{(m)})$, we can write,

$$p(z_{n,k}|\mu_k^{(m)}) = P(s_{n,k} = 0|\mu_k^{(m)}) + P(s_{n,k} = 1|\mu_k^{(m)}) \tag{5.12}$$

$$P(s_{n,k} = 1|\mu_k^{(m)}) = Q\left(\frac{\gamma - g_n(\mu_k^{(m)}) - \mu_v}{\sigma_v}\right) \tag{5.13}$$

$$P(s_{n,k} = 0|\mu_k^{(m)}) = 1 - Q\left(\frac{\gamma - g_n(\mu_k^{(m)}) - \mu_v}{\sigma_v}\right) \tag{5.14}$$

where $Q(\cdot)$ denotes the complementary function.

The initial set of particles $X_0^{(m)}, m = 1, 2...., M$, are drawn from a prior distribution $\pi(X_0)$, and the weights of the particles are set to $\frac{1}{M}$. Suppose now that at time instant $t_k - 1$, we have the random measure $\chi_{k-1} = \left\{X_{0:k-1}^{(m)}, w_{k-1}^{(m)}\right\}_{m=1}^{M}$. Then the steps of a particle filter recursion can be implemented as follows.

1) **Selection of most promising particle streams:**
For selection of most promising particles, the conditional mean of $X_k^{(m)}$ given $X_{k-1}^{(m)}$ is used as a characterizing parameter of every stream,

$$\mu_k^{(m)} = E\left(X_k|X_{k-1}^{(m)}\right) \tag{5.15}$$

The conditional means are computed from,

$$\mu_k^{(m)} = F_k X_{k-1}^{(m)} \tag{5.16}$$

This is followed by computation of the weights according to 5.10 and their normalization. Finally, a set of indices $\{i_m\}$ are drawn from the probability mass function (pmf) represented by the normalized weights.

2) **New particle generation:**
The first two elements of the four-dimensional state X_k represent the location of the target in a two-dimensional space, and the components of the velocity in this space. That implies that the generation of $X_k^{(m)}$ requires drawing only two-dimensional random variables. The generation can be carried out by first,

propagating the velocity components one step ahead using the joint distribution $p(\dot{x}_k, \dot{y}_k|\dot{x}_{k-1}, \dot{y}_{k-1})$ and second, computing the locations according to

$$x_k^{(m)} = x_{k-1}^{(i_m)} + \frac{\Delta t_k}{2}\left(\dot{x}_k^{(m)} + \dot{x}_{k-1}^{(i_m)}\right) \tag{5.17}$$

$$y_k^{(m)} = y_{k-1}^{(i_m)} + \frac{\Delta t_k}{2}\left(\dot{y}_k^{(m)} + \dot{y}_{k-1}^{(i_m)}\right) \tag{5.18}$$

3) **Weight computation:**

The newly generated particles are assigned weights according to

$$w_k(m) \propto \frac{p\left(Z_k|X_k^{(m)}\right)}{p\left(Z_k|\mu_k^{(i_m)}\right)} \tag{5.19}$$

4) **State estimation:**

Once the weights are normalized, one can use χ_k to compute estimates of the unknown states. The minimum mean square error (MMSE) estimate is obtained form,

$$\hat{X}_k = \sum_{m=1}^{M} w_k^{(m)} X_k^{(m)} \tag{5.20}$$

5.2.4 Simulation Results

In this Section we present Matlab based simulations that illustrate the performances of the above algorithm. We considered a scenario where the examined network consisted of $N = 121$ sensors deployed in a monitoring field with dimensions $100m \times 100m$.

Simulation parameters;

- Attenuation parameter $\alpha = 2.5$
- Reference power parameter to $\psi = 5000$ at $d_0 = 1m$
- Radio range of the sensors is $22m$
- Threshold $\gamma = 2$.

- The covariance matrix of the state noise process, Qu =diag$\{0.05, 0.01\}$ and the measurement noise mean, $\mu_v = 1$ variance, $\sigma_v^2 = 0.01$.
- Sampling interval $\Delta t_k = 1s$.
- Number of particles $M = 1000$
- The prior for the target's location and velocity was a Gaussian distribution with mean $X_0 = [0 \quad 0.01 \quad 0 \quad 0.01]^T$ and covariance matrix Ξ =diag$\{10, 0.01, 10, 0.01\}$

We experimented with two sensor networks, with deterministically and randomly deployed sensors as shown in Figure 5.2. It can be seen that the algorithm track the target's trajectory closely.

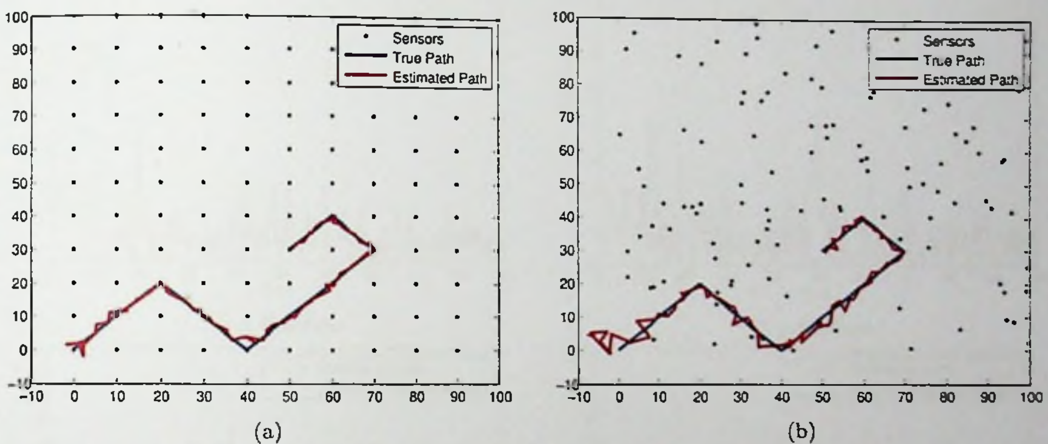


Figure 5.2: A realization of a target trajectory (Nodes = 121, Radio Range = 22m) (a)Deterministically deployed sensor network (b)Randomly deployed sensor network

In Figure 5.3, we display the root mean square errors (RMSEs) of the location estimate of the target through the algorithm. The RMSEs in this experiment were obtained by averaging 100 different realizations. Visual inspection of Figure 5.2 (a) and (b) show that accuracy of tracking is better in the deterministically deployed sensor network compared to the one with randomly deployed network. The difference in performance depends on various factors including on the number of sensors in the field, their constellation, the actual trajectory of the target, the strength of the transmitted signal by the target and the sampling interval.

The RMSEs of the location coordinates and velocity components of the target are shown in the Figure 5.4.

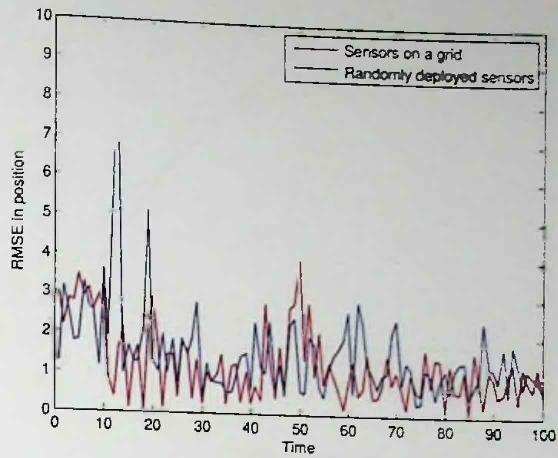
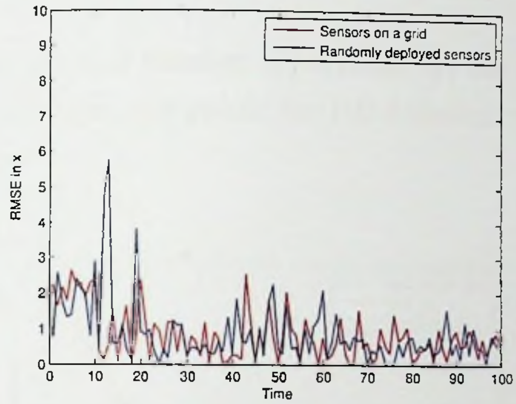
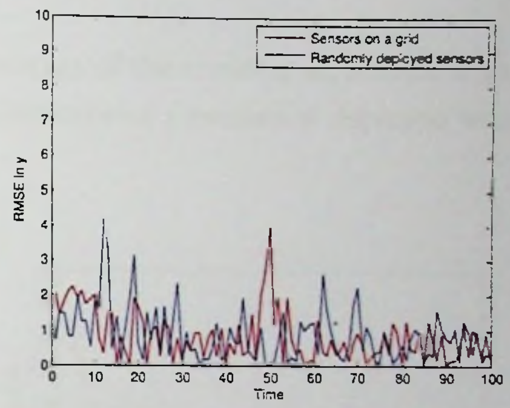


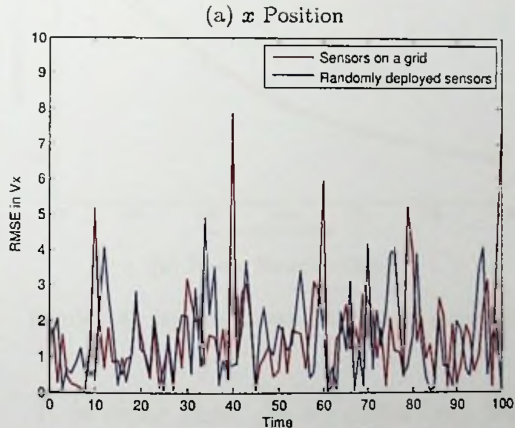
Figure 5.3: RMSEs of the location estimates of the target (Nodes = 121, Radio Range = 22m)



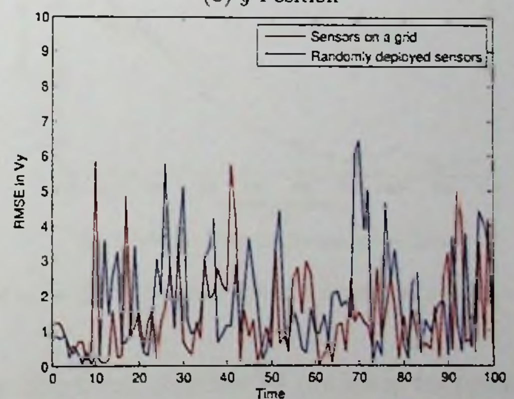
(a) x Position



(b) y Position



(c) $\dot{x}$ Velocity in x direction



(d) $\dot{y}$ Velocity in y direction

Figure 5.4: RMSEs of location coordinates and velocity components (Nodes = 121, Radio Range = 22m)

Figure 5.5 shows the performances of the algorithm expressed by the cumulative distribution functions (CDFs) of RMSEs. Again 100 different realizations were used in the experiment. Through figures it is evident that around 70% of nodes have a RMSE less than 2.

We further studied the impact of the total number of nodes, velocity of the

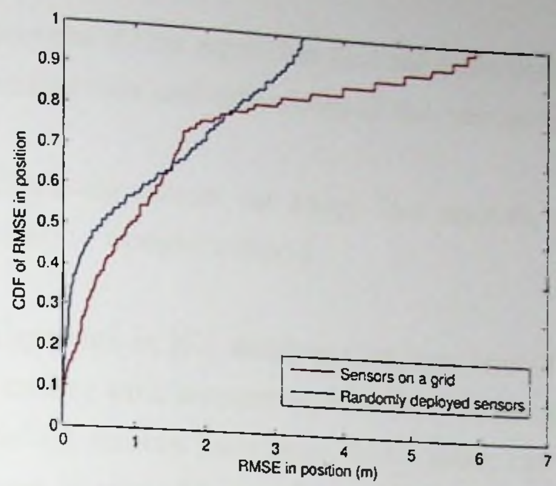
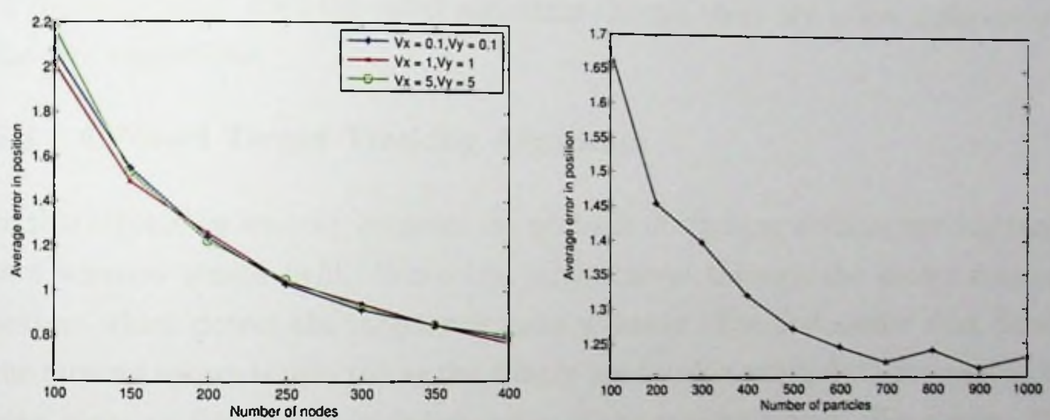


Figure 5.5: CDF of RMSEs (Nodes = 121, Radio Range = 22m)

target and number of particles, on the accuracy of the tracking algorithm. Figure 5.6 shows the result for 100 different realizations of a randomly deployed sensor field.



(a) Radio Range = 22m

(b) Nodes = 200, Radio Range = 22m

Figure 5.6: (a)RMSE with different total number of nodes (b)RMSE with different number of particles

It is clear that, as we expected, when the total number of nodes is increasing in the network, localization accuracy of the tracking algorithm is improved. On the other hand, the velocity of the target has very little effect on the accuracy of the tracking algorithm. Hence for same network realization, moving targets with different velocities shows almost same RMSE. Further, as we expected when the number of particles of the particle filter increases, localization accuracy of the tracking algorithm also improves.

We use this algorithm as a reference for performance comparison of the proposed novel tracking algorithm in the following Section. Though there is no direct link between the concepts of this algorithm and the proposed novel algorithm, we can list down few similarities and differences of the two as follows.

Similarities: Both algorithms use range free sensors, same target motion model and the same state update method.

Differences: Algorithm in [61] assumes that locations of the existing sensors in the network are known with zero error, where as in the novel algorithm those locations to the existing sensors have been found using the DV-Hop algorithm. Algorithm in [61] uses Particle filtering for recursive update of the target state, where as the novel algorithm uses Kalman filter of the same.

We use this algorithm for performance comparison in this research basically because of its similarity of the network layout. The network is a range free one with the same target motion model. Therefore we can compare the performances of this algorithm with the novel algorithm though there are a few differences in the two algorithms.

5.3 A Novel Target Tracking Algorithm

In this algorithm we only consider the problem of tracking a single moving target in a wireless sensor field. When the target moves through the sensor network, sensors which detect the target will make a cluster. The first sensor that detects the moving target is selected as the cluster head inside which data processing will take place to find the state information of the moving target. We assume that the existing sensor network is already localized using the DV-Hop algorithm and each sensor node in the network has the Hop Count Table. Further we assume that there is no transmission delays. Using the distance information inside the cluster head, it will estimate the state information of the target and transmit it to a fusion center. Further, we assume that all the sensors are of the same type and have the same noise characteristics. Figure 5.7 shows the basic idea behind this algorithm.

During this algorithm we combine the DV-Hop localization algorithm with Kalman filtering. The DV-Hop algorithm is used for pre-localization of the target and measurement conversion. The converted measurement and its associated noise statistics are then used in a standard Kalman filter for recursive update of

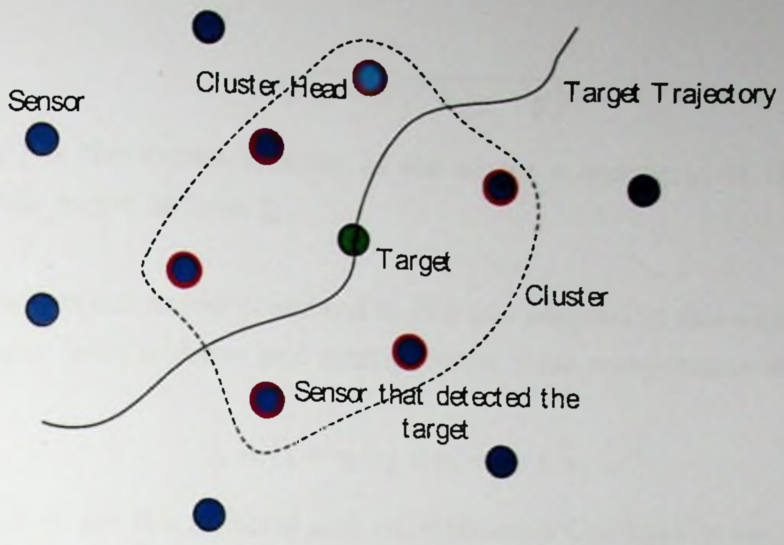


Figure 5.7: Localizing a newly added node

the target state. Figure 5.8 illustrates these main steps of the algorithm.

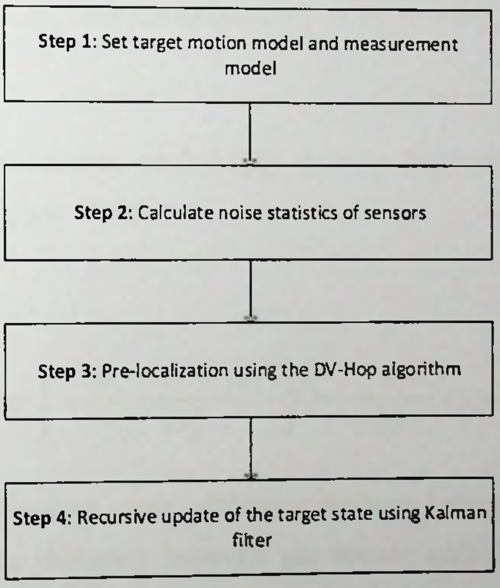


Figure 5.8: Novel tracking algorithm

We use the same target motion model presented in Section 5.2.2 in this algorithm.

5.3.1 Measurement Model

Let z_i be the distance measurement to the target obtained by the sensor i at time t_k and r_i be the actual distance between the sensor i and the target at time t_k.

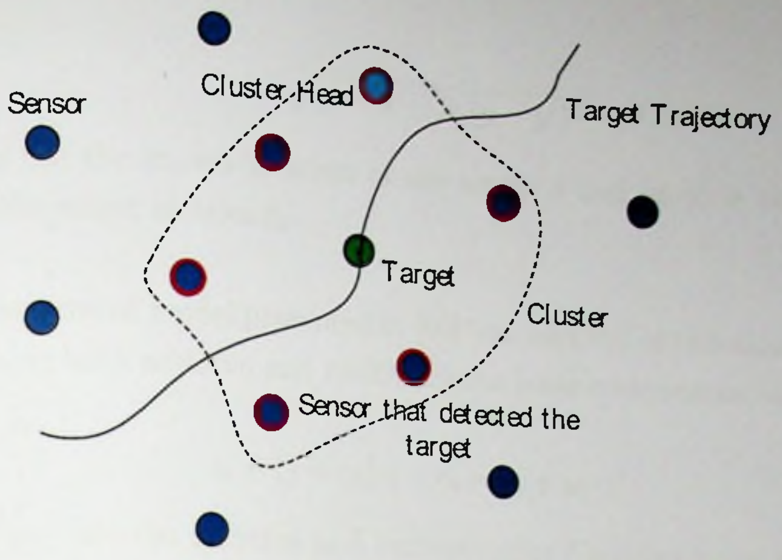


Figure 5.7: Localizing a newly added node

the target state. Figure 5.8 illustrates these main steps of the algorithm.

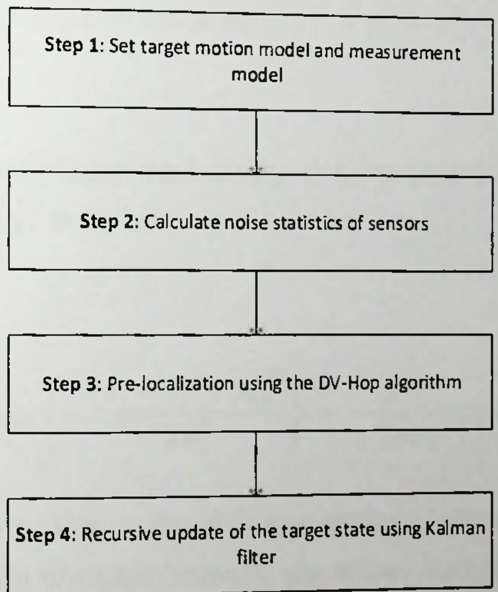


Figure 5.8: Novel tracking algorithm

We use the same target motion model presented in Section 5.2.2 in this algorithm.

5.3.1 Measurement Model

Let z_i be the distance measurement to the target obtained by the sensor i at time t_k and r_i be the actual distance between the sensor i and the target at time t_k.

$$r_i = \sqrt{(x - x^i)^2 + (y - y^i)^2} \tag{5.21}$$

where (x^i, y^i) is the known location of the sensor i and (x, y) is the unknown location of the target at time t_k.

The measurement model presented in [60] was adopted in this algorithm. The model contains both additive and multiplicative noise components as follows.

$$z_i = (1 + \gamma_i) r_i + n_i = r_i + u_i \tag{5.22}$$

where n_i and γ_i are the additive and multiplicative Gaussian noise components of sensor i with means μ_n and μ_γ and covariances σ_n^2 and σ_γ^2, respectively. These two components are uncorrelated.

The total noise of the sensor i is denoted by $u_i = n_i + r_i \gamma_i$. It is also Gaussian with mean $\mu_i = \mu_n + r_i \mu_\gamma$ and covariance $\sigma_i^2 = \sigma_n^2 + \sigma_\gamma^2 r_i^2$, which are dependent on actual distance r_i.

From 5.22, the conditional probability density function (PDF) of the measurement z_i, given (x, y) is as follows.

$$p(z_i|x, y) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{(x_i - r_i - \mu_i)^2}{2\sigma_i^2}\right\} = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{[r_i - (z_i - \mu_i)]^2}{2\sigma_i^2}\right\} \tag{5.23}$$

Due to the multiplicative noise, different sensors have different noise magnitudes depending on the distance between the sensor and the target. We assume that there are n_k ($n_k \geq 3$) sensors have detected the target at time t_k, and all the measurements are gathered at the cluster head. Let Z_k denote the measurements with the same time stamps from all the n_k sensors.

$$Z_k = \{z_1(k) \quad z_2(k) \quad \cdots \quad z_{n_k}(k)\} \tag{5.24}$$

Therefore the problem for the cluster head is to estimate the target state X_k, denoted by $\hat{X}_{k|k}$, given the measurements Z_j, where $j = 0, 1, ..., k$.

$$r_i = \sqrt{(x - x^i)^2 + (y - y^i)^2} \tag{5.21}$$

where (x^i, y^i) is the known location of the sensor i and (x, y) is the unknown location of the target at time t_k.

The measurement model presented in [60] was adopted in this algorithm. The model contains both additive and multiplicative noise components as follows.

$$z_i = (1 + \gamma_i)r_i + n_i = r_i + u_i \tag{5.22}$$

where n_i and γ_i are the additive and multiplicative Gaussian noise components of sensor i with means μ_n and μ_γ and covariances σ_n^2 and σ_γ^2, respectively. These two components are uncorrelated.

The total noise of the sensor i is denoted by $u_i = n_i + r_i\gamma_i$. It is also Gaussian with mean $\mu_i = \mu_n + r_i\mu_\gamma$ and covariance $\sigma_i^2 = \sigma_n^2 + \sigma_\gamma^2 r_i^2$, which are dependent on actual distance r_i.

From 5.22, the conditional probability density function (PDF) of the measurement z_i, given (x, y) is as follows.

$$p(z_i|x, y) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{(z_i - r_i - \mu_i)^2}{2\sigma_i^2}\right\} = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{[r_i - (z_i - \mu_i)]^2}{2\sigma_i^2}\right\} \tag{5.23}$$

Due to the multiplicative noise, different sensors have different noise magnitudes depending on the distance between the sensor and the target. We assume that there are n_k ($n_k \geq 3$) sensors have detected the target at time t_k, and all the measurements are gathered at the cluster head. Let Z_k denote the measurements with the same time stamps from all the n_k sensors.

$$Z_k = \{z_1(k) \quad z_2(k) \quad \cdots \quad z_{n_k}(k)\} \tag{5.24}$$

Therefore the problem for the cluster head is to estimate the target state X_k, denoted by $\hat{X}_{k|k}$, given the measurements Z_j, where $j = 0, 1, ..., k$.

$$r_i = \sqrt{(x - x^i)^2 + (y - y^i)^2} \tag{5.21}$$

where (x^i, y^i) is the known location of the sensor i and (x, y) is the unknown location of the target at time t_k.

The measurement model presented in [60] was adopted in this algorithm. The model contains both additive and multiplicative noise components as follows.

$$z_i = (1 + \gamma_i)r_i + n_i = r_i + u_i \tag{5.22}$$

where n_i and γ_i are the additive and multiplicative Gaussian noise components of sensor i with means μ_n and μ_γ and covariances σ_n^2 and σ_γ^2, respectively. These two components are uncorrelated.

The total noise of the sensor i is denoted by $u_i = n_i + r_i\gamma_i$. It is also Gaussian with mean $\mu_i = \mu_n + r_i\mu_\gamma$ and covariance $\sigma_i^2 = \sigma_n^2 + \sigma_\gamma^2 r_i^2$, which are dependent on actual distance r_i.

From 5.22, the conditional probability density function (PDF) of the measurement z_i, given (x, y) is as follows.

$$p(z_i|x, y) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{(x_i - r_i - \mu_i)^2}{2\sigma_i^2}\right\} = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{[r_i - (z_i - \mu_i)]^2}{2\sigma_i^2}\right\} \tag{5.23}$$

Due to the multiplicative noise, different sensors have different noise magnitudes depending on the distance between the sensor and the target. We assume that there are n_k ($n_k \geq 3$) sensors have detected the target at time t_k, and all the measurements are gathered at the cluster head. Let Z_k denote the measurements with the same time stamps from all the n_k sensors.

$$Z_k = \{z_1(k) \quad z_2(k) \quad \cdots \quad z_{n_k}(k)\} \tag{5.24}$$

Therefore the problem for the cluster head is to estimate the target state X_k, denoted by $\hat{X}_{k|k}$, given the measurements Z_j, where $j = 0, 1, ..., k$.

$$r_i = \sqrt{(x - x^i)^2 + (y - y^i)^2} \tag{5.21}$$

where (x^i, y^i) is the known location of the sensor i and (x, y) is the unknown location of the target at time t_k.

The measurement model presented in [60] was adopted in this algorithm. The model contains both additive and multiplicative noise components as follows.

$$z_i = (1 + \gamma_i)r_i + n_i = r_i + u_i \tag{5.22}$$

where n_i and γ_i are the additive and multiplicative Gaussian noise components of sensor i with means μ_n and μ_γ and covariances σ_n^2 and σ_γ^2, respectively. These two components are uncorrelated.

The total noise of the sensor i is denoted by $u_i = n_i + r_i\gamma_i$. It is also Gaussian with mean $\mu_i = \mu_n + r_i\mu_\gamma$ and covariance $\sigma_i^2 = \sigma_n^2 + \sigma_\gamma^2 r_i^2$, which are dependent on actual distance r_i.

From 5.22, the conditional probability density function (PDF) of the measurement z_i, given (x, y) is as follows.

$$p(z_i|x, y) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{(x_i - r_i - \mu_i)^2}{2\sigma_i^2}\right\} = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{[r_i - (z_i - \mu_i)]^2}{2\sigma_i^2}\right\} \tag{5.23}$$

Due to the multiplicative noise, different sensors have different noise magnitudes depending on the distance between the sensor and the target. We assume that there are n_k $(n_k \geq 3)$ sensors have detected the target at time t_k, and all the measurements are gathered at the cluster head. Let Z_k denote the measurements with the same time stamps from all the n_k sensors.

$$Z_k = \{z_1(k) \quad z_2(k) \quad \cdots \quad z_{n_k}(k)\} \tag{5.24}$$

Therefore the problem for the cluster head is to estimate the target state X_k, denoted by $\hat{X}_{k|k}$, given the measurements Z_j, where $j = 0, 1, ..., k$.

5.3.2 Target Tracking Algorithm

In this Section we explain our proposed tracking algorithm in detail. First we explain how to find the noise characteristics of sensor nodes using least square algorithm. Then we explain the DV-Hop based pre-localization process of the proposed tracking algorithm. This will be followed by a Kalman filter for recursive estimation of the target state.

5.3.2.1 Noise Statistics Computation

We need to find a proper method to estimate the noise statistics of the sensors. Mean and covariance of the additive and the multiplicative noise of sensors can be estimated through experiments as follows.

Suppose we run m tests with target at different positions in the sensor field. Denote the actual position of target to be r_i and we assume that r_i is known. N measurement samples $z_i^j, j = 1,, N$ are collected for N different positions of target, where N is a large number.
Empirical estimates of the mean and variance can be found as follows.

$$\bar{\mu}_i = \frac{1}{N}\sum_{j=1}^{N} z_i^j, \quad \bar{\sigma}_i^2 = \frac{1}{N-1}\sum_{j=1}^{N}(z_i^j - \bar{\mu}_i) \tag{5.25}$$

where $i = 1, \ldots.., m$. Using the ergodicity of the stationary process z_i^j and the independence between the additive and multiplicative noises, when $N \to \infty$, we have,

$$\bar{\mu}_i \to E\left\{z_i^j\right\} = r_i + r_i\mu_\gamma + \mu_n \tag{5.26}$$

$$\bar{\sigma}_i^2 \to E\left\{(z_i^j - E\left\{z_i^j\right\})^2\right\} = r_i^2\sigma_\gamma^2 + \sigma_n^2 \tag{5.27}$$

Define approximation errors as;

$$e_1(i) = \bar{\mu}_i - (r_i + r_i\mu_\gamma + \mu_n) \tag{5.28}$$

$$e_2(i) = \bar{\sigma}_i^2 - (r_i^2\sigma_\gamma^2 + \sigma_n^2) \tag{5.29}$$

We can use the least squares algorithm to determine the estimates of $\mu_\gamma, \mu_n, \sigma_\gamma^2$ and σ_n^2. We can minimize J_1 and J_2 defined as follows.

$$J_1 = \sum_{i=1}^{m} e_1^2(i), \quad J_2 = \sum_{i=1}^{m} e_2^2(i) \tag{5.30}$$

The minimizations of J_1 and J_2 are given by,

$$\frac{\partial J_1}{\partial \mu_\gamma} = 0, \quad \frac{\partial J_1}{\partial \mu_n} = 0, \quad \frac{\partial J_1}{\partial \sigma_\gamma^2} = 0, \quad \frac{\partial J_2}{\partial \sigma_n^2} = 0 \tag{5.31}$$

Through calculations,

$$\begin{bmatrix} \hat{\mu}_\gamma \\ \hat{\mu}_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} r_i^2 & \sum_{i=1}^{m} r_i \\ \sum_{i=1}^{m} r_i & m \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{m} r_i(\bar{\mu}_i - r_i) \\ \sum_{i=1}^{m} (\bar{\mu}_i - r_i) \end{bmatrix} \tag{5.32}$$

$$\begin{bmatrix} \hat{\sigma}_\gamma^2 \\ \hat{\sigma}_n^2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} r_i^4 & \sum_{i=1}^{m} r_i^2 \\ \sum_{i=1}^{m} r_i^2 & m \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{m} r_i^2 \bar{\sigma}_i^2 \\ \sum_{i=1}^{m} \bar{\sigma}_i^2 \end{bmatrix} \tag{5.33}$$

5.3.2.2 Pre-localization using the DV-Hop algorithm

In this Section, we convert the distance measurements gathered at the cluster head, to a position information of the target using the DV-Hop algorithm. We assume that means and covariances are known. This measurement conversion process is called pre-localization.

First, we assume that the existing sensor network is already localized using the DV-Hop algorithm. That means each node in the network knows its Hop Count Table 4.3. Further, each node in the network has an estimated Hop Size associated with it. Once a moving target is detected by the sensors in the network, following two step process will be carried out to pre-localize the target.

Step 1: Node that detects the moving target first (target is within the communication range) at time t_k becomes the cluster head and its Hop Size ($HopSize_{CH}$) is assigned as the Hop Size of the moving target at time t_k.

Step 2: Cluster head estimates the location of the target at time t_k using either trilateration / multilateration or maximum likelihood estimation using the distance information in the Hop Count Table as mentioned in Section 4.7.

Let (x_u, y_u) be the location of the target at time t_k and (x_j, y_j) be the known location of the anchor node j. $HopSize_j$ is the estimated Hop Size of the node j. Then;

$$\begin{aligned}
(x_u - x_1)^2 + (y_u - y_1)^2 &= d_{u1}^2 \\
(x_u - x_2)^2 + (y_u - y_2)^2 &= d_{u2}^2 \\
&\vdots \\
(x_u - x_n)^2 + (y_u - y_n)^2 &= d_{un}^2
\end{aligned}$$

where n is the number of anchors in the network.

Coordinates of the target at time t_k can be calculated using the following matrix operation.

Here,

$$d_{ui} = \begin{cases} HopSize_{CH} \times Hops_{ui} + 1 & : if \quad Cluster \quad Head \neq i^{th} \quad anchor \\ HopSize_{CH} & : if \quad Cluster \quad Head = i^{th} \quad anchor \end{cases} \tag{5.34}$$

Then,

$$A = -2 \times \begin{bmatrix} x_1 - x_n & y_1 - y_n \\ x_2 - x_n & y_2 - y_n \\ \vdots & \vdots \\ x_{n-1} - x_n & y_{n-1} - y_n \end{bmatrix} \tag{5.35}$$

$$B = \begin{bmatrix} d_{u1}^2 - d_{un}^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_{u2}^2 - d_{un}^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ d_{u(n-1)}^2 - d_{un}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix} \tag{5.36}$$

$$P = \begin{bmatrix} x_u \\ y_u \end{bmatrix} = (A^T A)^{-1} A^T B \tag{5.37}$$

After pre-localization, we have an estimation for the position of the target at time t_k as follows.

$$
\begin{aligned}
\bar{Z}_k &= [\bar{x}_k \quad \bar{y}_k]^T \\
\bar{x}_k &= x_u \\
\bar{y}_k &= y_u
\end{aligned}
\tag{5.38}
$$

5.3.2.3 Kalman Filtering

One the pre-localization is done, we can represent the pre-localized estimation of the target at time t_k as follows.

$$
\bar{Z}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} X_k + v_k = CX_k + v_k \tag{5.39}
$$

where v_k is the converted measurement noise.

Now it is time to specify the characteristics of v_k before the converted measurement $\bar{Z}_k$ can be used in Kalman filtering.

Assume that there are n ($n \geq 3$) which have detected the moving target simultaneously at time t_k and the measurement noises of different sensors are mutually independent. Defining, $Z = \{z_i, i = 1, 2, ..., n\}$, let's denote $p(Z|x, y)$ the jointly conditional probability density function of Z given (x, y) as follows. Further, since the noises of individual sensors are mutually independent.

$$
p(Z|x, y) = \prod_{i=1}^{n} p(z_i|x, y) \tag{5.40}
$$

where $p(z_i|x, y)$ is given in 5.23.

σ_i and μ_i depend on the actual distance between the target and sensor i, r_i which are unknown and are related to the unknown parameters (x, y). Therefore we make the assumption that the difference between z_i and r_i is very small and therefore σ_i^2 can be approximately replaced with $\sigma_{z_i}^2 = \sigma_n^2 + z_i^2\sigma_\gamma^2$. Then we can rewrite 5.23 as,

$$
p(z_i|x, y) \simeq \frac{1}{\sqrt{2\pi}\sigma_{z_i}} \exp\left\{ -\frac{[(1 + \mu_\gamma)r_i - \check{z}_i]^2}{2\sigma_{z_i}^2} \right\} \tag{5.41}
$$

where $\check{z}_i = z_i - \mu_n$ is the calibrated measurement.

Using the Bayes rule, we have the following posterior probability distribution of (x, y),

$$p(x, y|Z) = \frac{p(Z|x, y)p_a(x, y)}{p(Z)} \tag{5.42}$$

where $p_a(x, y)$ is the prior probability density function of (x, y) known by the sensor nodes. Here we assume that $p_a(x, y)$ is uniform within the monitoring field. Therefore we can represent $p(x, y|Z)$ can be represented as,

$$p(x, y|Z) = \alpha p(Z|x, y) \tag{5.43}$$

where α is independent of (x, y).

From 5.40 and 5.41;

$$p(Z|x, y) = \frac{1}{(2\pi)^{\frac{n}{2}} \prod_{i=1}^{n} \sigma_{z_i}} \exp\left\{-\sum_{i=1}^{n} \frac{[(1+\mu_\gamma)r_i - \check{z}_i]^2}{2\sigma_{z_i}^2}\right\} \tag{5.44}$$

Let,

$$f(x, y) = \sum_{i=1}^{n} \frac{[(1+\mu_\gamma)r_i - \check{z}_i]^2}{2\sigma_{z_i}^2} \tag{5.45}$$

Then,

$$p(x, y|Z) = \frac{\alpha}{(2\pi)^{\frac{n}{2}} \prod_{i=1}^{n} \sigma_{z_i}} \exp(-f(x, y)) \tag{5.46}$$

After pre-localization,

$$f(x, y) \simeq f(\bar{x}, \bar{y}) + \frac{1}{2} \begin{bmatrix} x - \bar{x} \\ y - \bar{y} \end{bmatrix}^T H(\bar{x}, \bar{y}) \begin{bmatrix} x - \bar{x} \\ y - \bar{y} \end{bmatrix} \tag{5.47}$$

where $H(\bar{x}, \bar{y})$ is the Hessian matrix given by,

$$H(\bar{x}, \bar{y}) = \begin{bmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial x \partial y} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix}_{x=\bar{x}, y=\bar{y}} \tag{5.48}$$

Then,

$$p(x,y|Z) \simeq \beta \exp\left\{-\frac{1}{2}\begin{bmatrix} x-\bar{x} \\ y-\bar{y} \end{bmatrix}^T H(\bar{x},\bar{y}) \begin{bmatrix} x-\bar{x} \\ y-\bar{y} \end{bmatrix}\right\} \tag{5.49}$$

where β is a constant. That means v_k is approximately a Gaussian distribution with zero mean and covariance matrix of

$$R_k = H^{-1}(\bar{x}_k, \bar{y}_k) \tag{5.50}$$

Final step of the algorithm is to use the Kalman filter to update the target state using the pre-localization estimation and associated noises as follows.

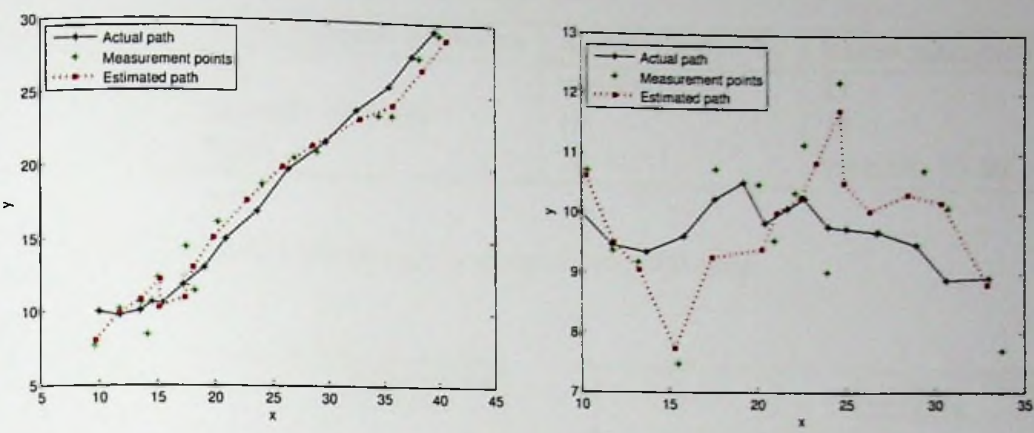
$$\hat{X}_{k+1|k} = F_k \hat{X}_{k|k} \tag{5.51}$$
$$P_{k+1|k} = F_k P_{k|k} F_k^T + G_k Q_w G_k^T \tag{5.52}$$
$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1}(\bar{Z}_{k+1} - C\hat{X}_{k+|k}) \tag{5.53}$$
$$P_{k+1|k+1} = P_{k+1|k} - K_{k+1} S_{k+1} K_{k+1}^T \tag{5.54}$$
$$S_{k+1} = C P_{k+1|k} C^T + R_{k+1} \tag{5.55}$$
$$K_{k+1} = P_{k+1|k} C^T S_{k+1}^{-1} \tag{5.56}$$

The initial estimates are given as $\hat{X}_{0|0} = \bar{X}_0$ and $P_{0|0} = P_0$ for some large positive definite P_0.

5.3.3 Simulation Results

In this Section we present Matlab based simulations that illustrate the performances of the proposed tracking algorithm. We considered a scenario where the examined network consisted of $N = 200$ sensors deployed randomly in a monitoring field with dimensions $100m \times 100m$. Additive noise is Gaussian and white with zero mean and covariance of 0.01 and multiplicative noise is Gaussian and white with zero mean and covariance of 0.01 for every sensor. Sampling interval is $1s$. The initial state of the target is $X_0 = [0 \quad 0 \quad 1 \quad 1]$. The covariance matrix Q_w of the target noisy acceleration is given by $Q_w = \text{diag}\{0.01 \quad 0.01\}$.

In Figure 5.9, we can see two target tracing scenarios and it can be seen that the algorithm track the target's trajectory closely. In trajectory 1 of Figure 5.9 (a), target moves along an almost straight line (non-maneuvering target tracking), where as in trajectory 2 of Figure 5.9 (b), target moves along a path with several sudden turns (maneuvering target tracking). Estimated path in Figure 5.9 (a) is closer to the original path than in Figure 5.9 (b). Therefore it is evident that the proposed algorithm is more suitable for non-maneuvering target tracking.



(a) Trajectory 1

(b) Trajectory 2

Figure 5.9: A realization of a target trajectory (Nodes = 200, Radio Range = 22m)

Figure 5.10 shows the performances of the algorithm expressed by the cumulative distribution functions (CDFs) of RMSEs. 100 different realizations were used in the experiment.

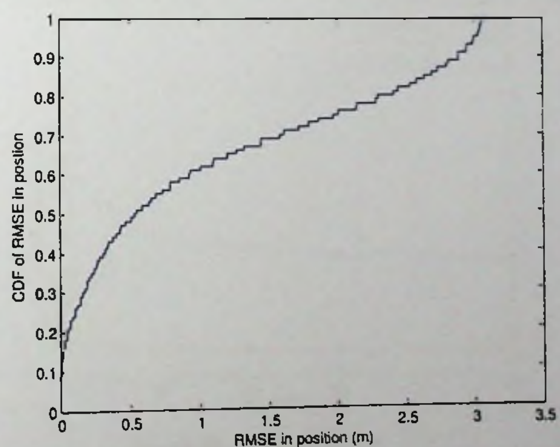


Figure 5.10: CDF of RMSEs (Nodes = 200, Radio Range = 22m)

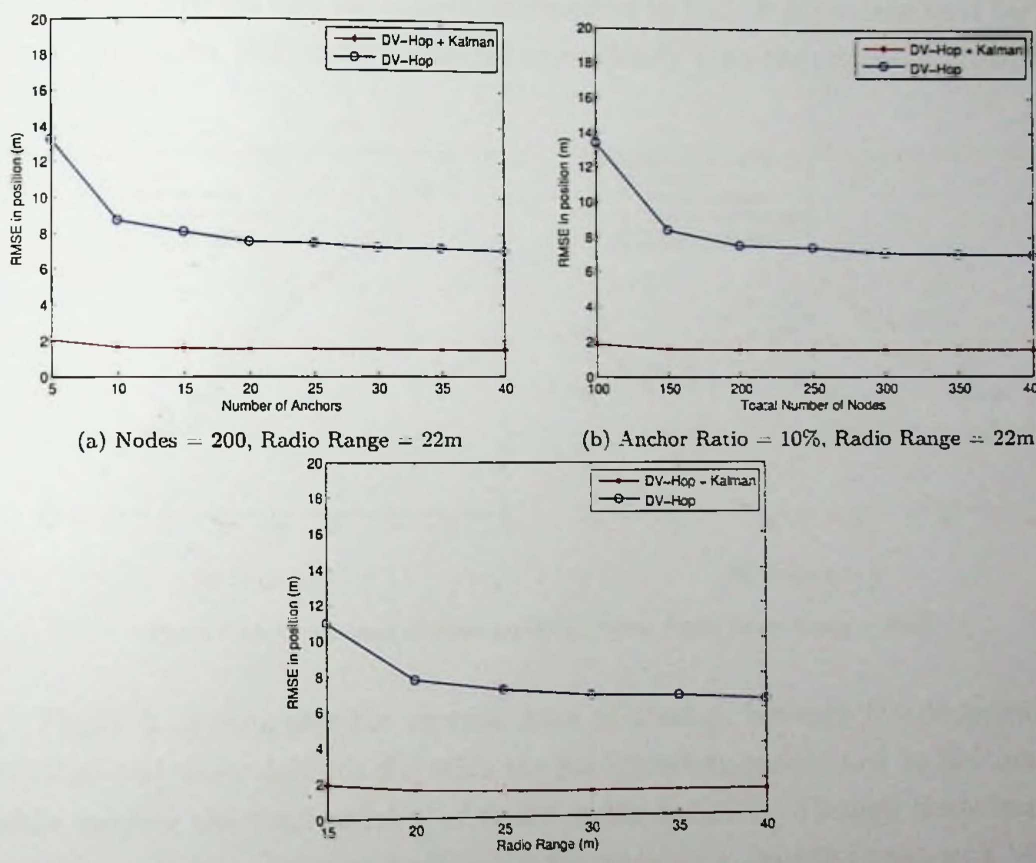
We further studied the impact of the number of anchors, total number of nodes and different radio ranges of sensors, on the accuracy of the tracking algorithm.

Then, the proposed algorithm is studied for its performance by varying one at a time, the anchor ratio (AR), the total number of nodes (N) and the radio range (R) of the nodes as summarized in Table 5.1.

Table 5.1: Simulation Instances

Figure No	Total Number of Nodes	Anchor Ratio(%)	Radio Range(m)
5.11(a)	200	variable 5-40	22
5.11(b)	variable 100-400	10	22
5.11(c)	200	10	variable 15-40

All the simulation results are averaged over 100 runs.



(a) Nodes = 200, Radio Range = 22m

(b) Anchor Ratio = 10%, Radio Range = 22m

(c) Nodes = 200, Anchors = 20

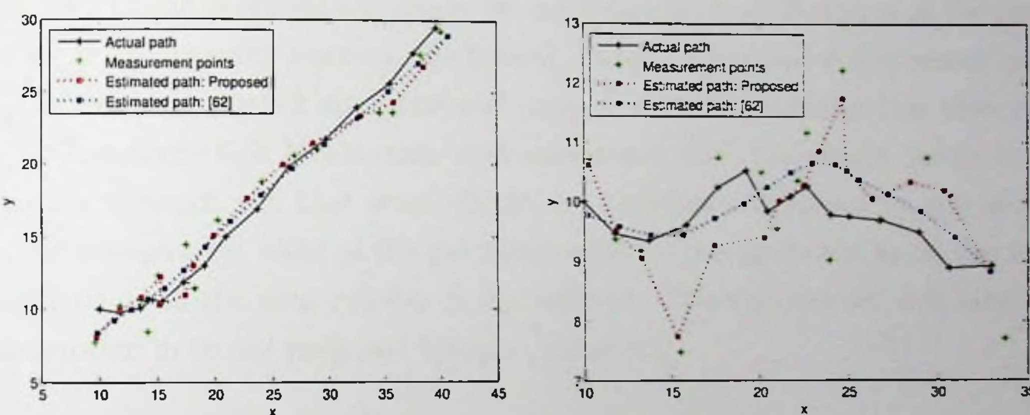
Figure 5.11: (a)RMSE with different number of anchors (b)RMSE with different total number of nodes (c)RMSE with different radio ranges

Through simulations it is quite evident that the proposed method of combining the DV-Hop algorithm with Kalman filter has improved the localization accuracy of target the considerably.

5.3.4 Performance Comparison

In this Section we compare the performance of proposed tracking algorithm with the work done in [61] with the modifications mentioned in Section 5.2. A scenario of sensors having a radio range of $22m$ in a $100m \times 100m$ area is used. The comparison of performances is done by varying the total number of nodes in the network. Anchor Ratio of the proposed algorithm is kept at a constant of 10%. The position error of the moving target is an average of over 100 simulations with different node configurations.

Figure 5.12 compares the estimated paths of a target trajectory using the proposed algorithm and the algorithm reported in [61]. It is evident that for the given realization, [61] tracks the target more closely than the proposed algorithm.



(a) Trajectory 1 (b) Trajectory 2

Figure 5.12: Comparison of target tracking (Nodes = 200, Radio Range = 22m)

Figure 5.13 compares the average error of position between the proposed algorithm and work done in [61] with the modifications mentioned in Section 5.2 while varying the total number of nodes in the network. Though these two algorithms take two different approaches for tracking a moving target in a WSN, we can compare the final outcome which is the average positioning error in these two cases.

Through simulations it is clear that the algorithm in [61] out performs the proposed algorithm in the best part of the graph. This is mainly due to several key differences between the two algorithms as listed below.

- [61] assumes that the nodes in the existing network have been localized

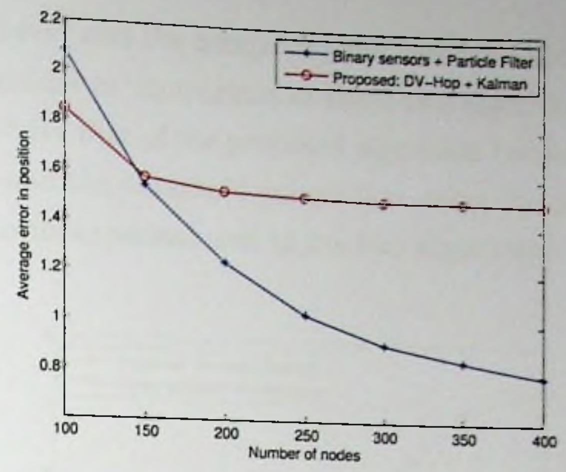


Figure 5.13: Average positioning error with different number of nodes (Nodes = 200. Radio Range = 22m)

with zero error. Therefore fusion center knows the exact locations of the sensor nodes in the network, which are then used to estimate the target's trajectory. However in the proposed algorithm we don't assume that locations of the sensor nodes in the existing network are known, instead locations of the sensor nodes in the existing network are calculated using the DV-Hop localization algorithm. Therefore there is a localization error associated with the sensor nodes in the existing network and that error effects the estimation accuracy of the moving target's trajectory. Most of the practical sensor networks do not know the exact locations of all the sensor nodes in the network. Therefore we did not take that assumption in to our proposed tracking algorithm.

∘ [61] uses RSSI enabled sensor nodes to measure the respective power levels. That means the algorithm in [61] relies on a ranging technique to detect the moving target. However the proposed tracking algorithm does this initial measuring using the DV-Hop algorithm which is a range free technique. Ranging techniques always give better estimations than range free techniques in terms of accuracy though they are more expensive.

• [61] uses the Particle filter for recursive update of the state vector of the moving target, whereas the proposed algorithm uses Kalman filter. Particle filter is a more complex algorithm which suits when the measurement equations are non-linear. Kalman filter can be used when the target dynamic and measurement equations are linear. The proposed algorithm produces a set of liner relationships after the measurement conversation process in pre-localization step.

Communication cost and the computational cost are another two parameters that we need to consider in comparison of these two algorithms. Computational cost of [61] is well above that of the proposed algorithm because of the more complex Particle filter and the threshold calculation using the power levels. Figure 5.14 compares the computational cost of the two algorithms.

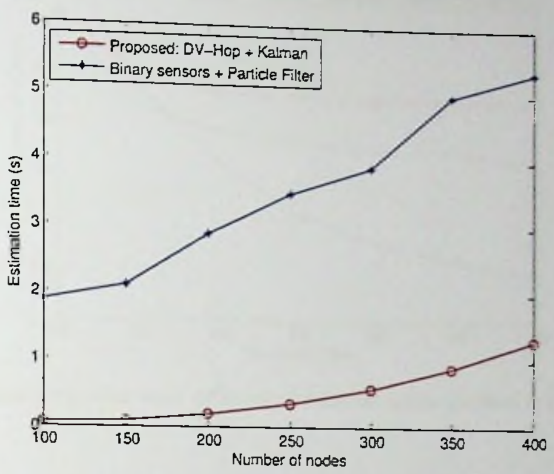


Figure 5.14: Computational cost comparison (Anchor Ratio = 10%, Radio Range = 22m)

Communication cost of the proposed algorithm is very much lower than the algorithm in [61]. In algorithm [61], all the sensors that detect the target at a certain time, should transmit a packet to the fusion center whereas in the proposed algorithm there's no additional packet transmission is taking place to estimate the target states. This is because the proposed algorithm uses the already available Hop Count Table in the pre-localization step.

The proposed target tracking algorithm can be implemented as a fully distributed manner, where as [61] is a centralized algorithm. This is one of the key advantages of the proposed algorithm over the [61]. Further in [61] at least three sensors in the network should detect the moving target at once. Otherwise the algorithm fails to estimate the target states. However in the proposed tracking algorithm, it is enough to detect the target only by a one sensor in the network. This is due to the pre-localization step of the algorithm. Therefore the proposed algorithm is more suitable to apply in emergency environments where nodes can be destroyed, due to its distributive nature, simplicity and demand of low sensor nodes.

To improve the performances of the proposed tracking algorithm, we applied the proposed improved DV-Hop algorithm introduced in Section 4.2, in the pre-localization step. Through this modification we could improve the estimation accuracy of the proposed target tracking algorithm further as shown in Figure 5.15.

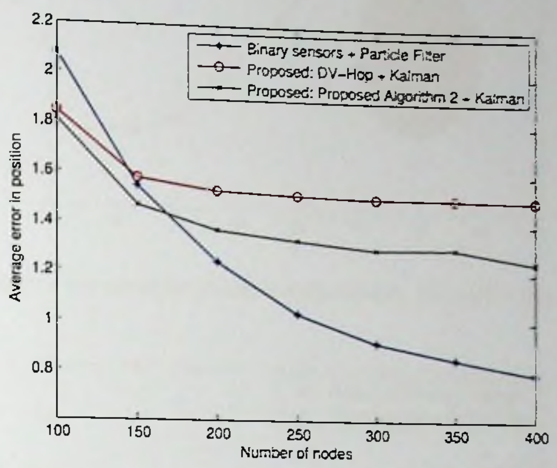


Figure 5.15: Average positioning error with different number of nodes (Nodes = 200, Radio Range = 22m)

5.3.5 Proposed Tracking Algorithm in an Emergency Environment

In this Section we apply the proposed algorithm in an emergency situation where nodes are getting destroyed due to a spreading fire. A scenario of 200 sensors having a radio range of $22m$ in a $100m \times 100m$ area is used as the initial network configuration. Radius of the fire gradually increases so that the nodes in the network get destroyed. We try to track a moving target with different radius of fire and compare the average position error of the target tracking. The position error of the moving target is an average of over 100 simulations with different node configurations.

Figure 5.16 shows the simulation environment of the tracking algorithm in a fire emergency.

Figure 5.17 shows the RMSE in position of the proposed target tracking algorithm with different radius of fire in the network.

Through simulations it is very clear that the RMSE of position increases when the radius of fire increases. Both anchors and the other nodes in the network are destroyed due to the fire and when fire increases with time more and more nodes in the network will be destroyed. Until there is at least a single node that detects

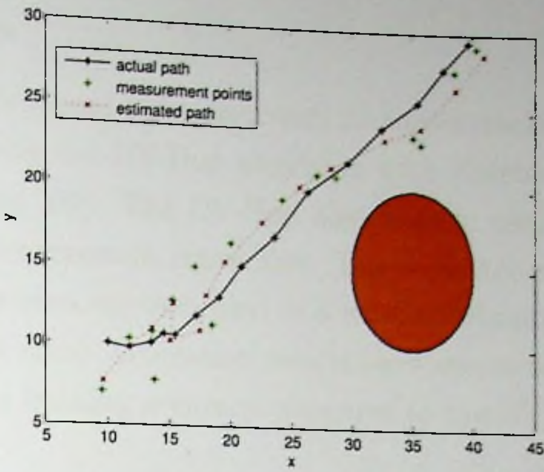


Figure 5.16: Target tracking in a fire situation (Initial configuration: Anchors = 20, Nodes = 200, Radio Range = 22m)

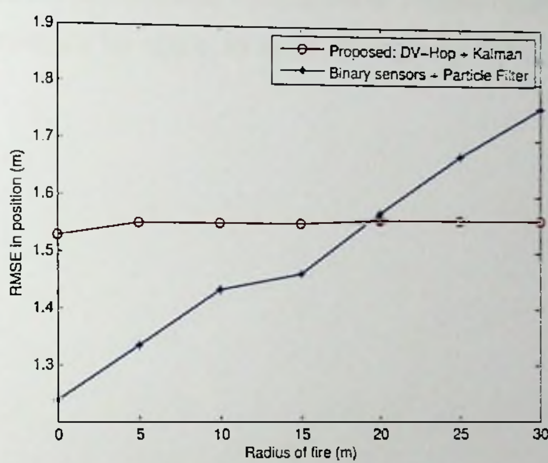


Figure 5.17: RMSE in position with different radius of fire (Initial configuration: Anchors = 20, Nodes = 200, Radio Range = 22m)

the moving target at a given time, the algorithm can estimate the target states. However due to the destruction of anchors which could have been selected as the cluster head, the measurement error comes out of pre-localization step can increase resulting an increase in RMSE in position of the moving target. That's why when the radius of fire increases sometimes the RMSE in position of the moving target increases.

In algorithm [61], when the sensor nodes in the existing network are gradually dying, estimation accuracy of the moving target reduces. However in the proposed algorithm, we don't have this problem severely. Therefore the proposed tracking algorithm is very much suitable for target tracking purposes in emergency environments.

5.3.6 Conclusion

In this Section, we proposed a new approach for target tracking in wireless sensor network by combining the DV-Hop algorithm with Kalman filtering, based on the work reported in [60]. The DV-Hop algorithm is used for pre-localization of the target and measurement conversion. The converted measurement and its associated noise statistics are then used in a standard Kalman filter for recursive update of the target state. Simulation results have shown that the proposed approach improves the tracking accuracy compared to the DV-Hop localization.

Then we applied the proposed tracking algorithm in a fire emergency environment and compared the results. Simulation results show that the proposed algorithm in good enough to apply in such emergency environments.

Chapter 6

NAVIGATION SUPPORT

The main objective of this Chapter is to a present a novel navigation support algorithm for fire fighters in an emergency situation. In previous two Chapters we applied localization and tracking functionalities in an emergency environment and here we extend that by incorporating navigation support to rescue operations.

6.1 Importance of Localization and Navigation

Localization and navigation support is very useful in many day to day applications, but essential in emergency rescue operations. Teams must be able to reach incident locations safely and quickly and incident commanders must be able to keep track of their locations [7]. The simple task of getting out of a building becomes a challenge with little or no visibility due to smoke and power failure. High levels of mental and physical stress add to the difficulty: getting lost in a burning or collapsing building can have fatal consequences for both the rescue personnel and the building's occupants as oxygen supplies run out and medical attention is delayed [7]. It has been identified that 'lost inside' as a major cause of injuries to fire fighters. It is also reported that disorientation and failure to locate victims are contributing factors to fire fighter deaths [7].

The following paragraph from [7], describes the importance of proper navigation support for fire fighters.

In some instances, fire fighters might have only a few seconds to reach safety. They must find the exit as quickly as possible and might not be able to retreat along the same path they used to enter the building owing to a collapsed ceiling or floor. Alternative exits might be available but will not be clearly visible. When a fire fighter radios a distress call, the rescue team must be able to find that person. Even when situations are not immediately life threatening, precious time can be wasted by searching the same room twice or failing to search another.

The incident commander also needs to know elements of the building layout, team members' locations, and the parts of the building that have already been searched.

6.1.1 Current Practices

Fire fighters have developed navigation practices for use in poor visibility. Details vary, but overall, they use the same ideas worldwide. These methods tend to be simple and practical, and the equipment is seemingly low-tech and very robust [7]. The following paragraph which was extracted form [7] describes such a technique.

Following a hose is a simple method for finding the exit in a dark or smoky building. If no hose is available, fire fighters can use dedicated ropes called lifelines that connect them to a point outside the dangerous area. The other end can remain attached if a new team comes in to continue the search, or fire fighters can attach additional lines to the main lifeline and branch off in different directions while remaining physically linked to the rest of the team. A series of knots on the main lifeline helps fire fighters determine the direction and distance to the exit and can serve as reference points when radioing positions to a commander. Likewise, a flashlight left in a room's doorway helps locate the exit and indicates to colleagues that the room is being searched; a chalk mark on the door indicates that a room has already been searched. Teams returning from a search mission can sketch the building's layout to assist the commander and any further teams.

Figure 6.1 shows a sketch of a lifeline and Figure 6.2 shows how the fire fighters follow the lifeline for navigation purpose.

6.2 Related Work

In the literature there are quite a few work addressing the problem of navigation support in emergency situations. Few of them are listed below.

• Sha *et al.* [3] presented WSNs as a promising technology for fire rescue applications. The paper specified the requirements of a fire rescue system and proposed FireNet, the WSN based architecture.

• Klann *et al.* [4] proposed a concept called LifeNet for using an ad-hoc sensor network providing relative positioning and a wearable system to support firefight-

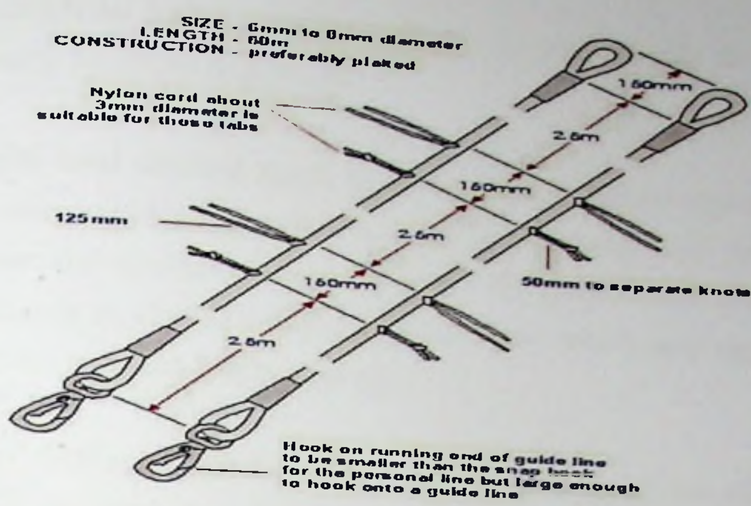


Figure 6.1: Lifeline for fire fighters (taken form Shropshire Fire and Rescue Service - Brigade Order 7)

Figure 6.2: Incoming team follows the lifeline (taken form Shropshire Fire and Rescue Service - Brigade Order 7)

ers at indoor-navigation under impaired visibility.

- Fischer *et al.* [7] presented a survey on location and navigation support for emergency responders. The paper describes the current localization and navigation techniques and challenges on high-tech location systems. At the end it provides a solid review on currently available location and navigation support systems for emergency responders.

- Tseng *et al.* [65] proposed a navigation algorithm for safely guiding people to quickly escape from a hazardous area. The design allows multiple exits and multiple emergency events in the sensing field. During a non-emergency situation, sensors are responsible for monitoring the environment. When emergency events are detected, the protocol can quickly identify hazardous areas and sensors can establish escape paths that are as safe as possible leading to exits. In particular, when surrounded by hazards, sensors will try to guide people as farther away from emergency locations as possible.

- Li *et al.* [66] used directed graph to model the emergency regions. Human's movements were regarded as network flows on the graph. By calculating the maximum flow and minimum cut on the graph, the system could provide firemen rescue commands to eliminate key dangerous areas, which may significantly reduce congestion and save trapped people.

- Tseng *et al.* [67] combined a distributed navigation algorithm with WSNs to help safely guide people to a building exit while helping them avoid hazardous areas. At normal time, sensors monitor the environment. When the sensors detect emergency events, the protocol quickly separates hazardous areas from safe areas, and the sensors establish escape paths.

6.3 Proposed Method to Emulate Lifeline Technique using WSNs

In this Section, we propose a WSN based mechanism to emulate the technique of lifeline used by the fire fighters. First we assume that the monitoring area is equipped with a grid based WSN. Grid based sensor network is not essential and even a randomly deployed sensor network is quite enough to run the algorithm. However, in this case we use a deterministically deployed sensor network as the back bone to run the algorithm to lay the lifeline (rope). Instead of physically laying a rope form a starting point to the destination point through the fire field, new sensor nodes will be thrown along the path so that other fire fighters can be followed the path to go to the destination. Figure 6.3 shows the deployment of the new sensor nodes alone the safest path through the fire field from the starting point to the destination.

6.3.1 Algorithm

Let's assume that each sensor node in the grid of the monitoring area provides a severity index $s_i(k)$, where $s_i(k)$ is the severity index [68] of i^{th} sensor at time t_k. Severity index is a cumulative result of the conditions of the environment defending on factors such as temperature, humidity, smoke and wind speed. We assume that severity index is fully capable of representing the state of the fire in the field. Higher severity index of sensor means that the fire at that area is

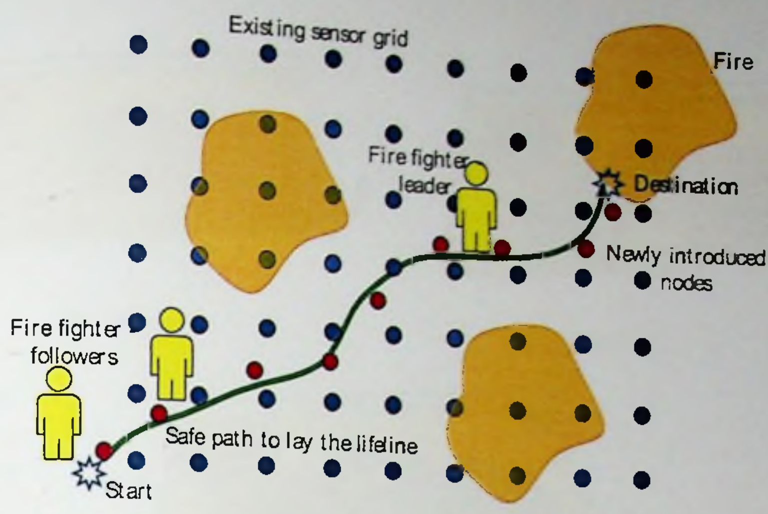


Figure 6.3: Localizing a newly added node

higher in magnitude. We assume that each and every sensor node is capable of computing the severity index of the present moment and few time steps ahead as $[s_i(k) \quad s_i(k+1) \quad \cdots \quad s_i(k+a)]$.

Let the severity matrix of the network at time t_k as, $S_k = \{s_1(k) \quad s_2(k) \cdots s_n(k)\}$, where n is the number of sensors in the network.

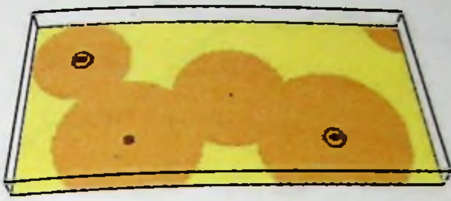
Then the fire fighter has to use the knowledge of severity matrix of time t_k to t_{k+a} to make the next safest movement to go to the required destination. Here the parameter a can be either 1 or 2 or 10 or 100 depending on the nature of the situation and the computational power of the system.

6.3.2 Simulation Results

Figure 6.4 shows a scenario of a spreading fire in an indoor environment which is simulated using Fire Dynamics Simulator (FDS). The objective is to find the safest path between two given points in the fire field to throw new sensor nodes to emulate the lifeline.

Figure 6.5 shows the temperature distribution at each point with time.

For this simulation we assume that the severity index of each sensor node is proportional to the temperature distribution of the field. Now let's imagine that the fire fighters need to go from point 2 to point 45 of the fire field given in Figure

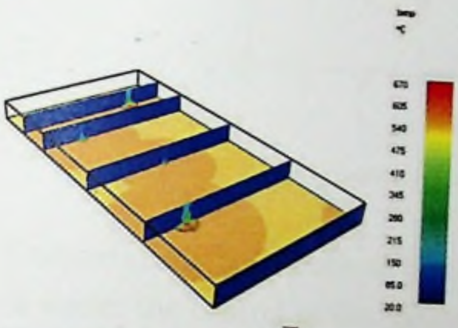


(a) At $t = 5$

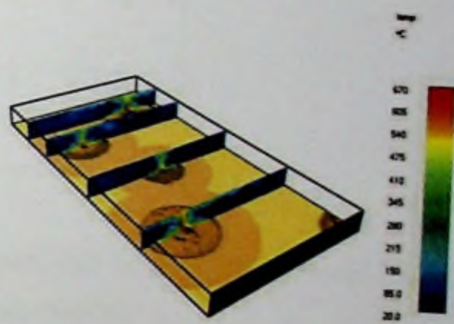
(b) At $t = 15$

(c) At $t = 30$

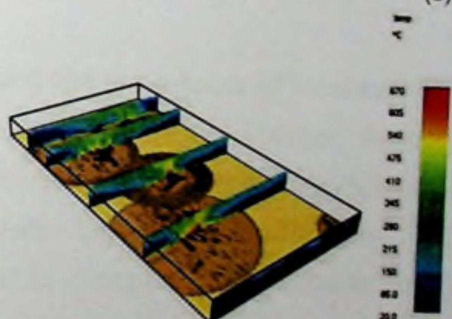
Figure 6.4: Spreading fire in an indoor environment



(a) At $t = 5$



(b) At $t = 15$

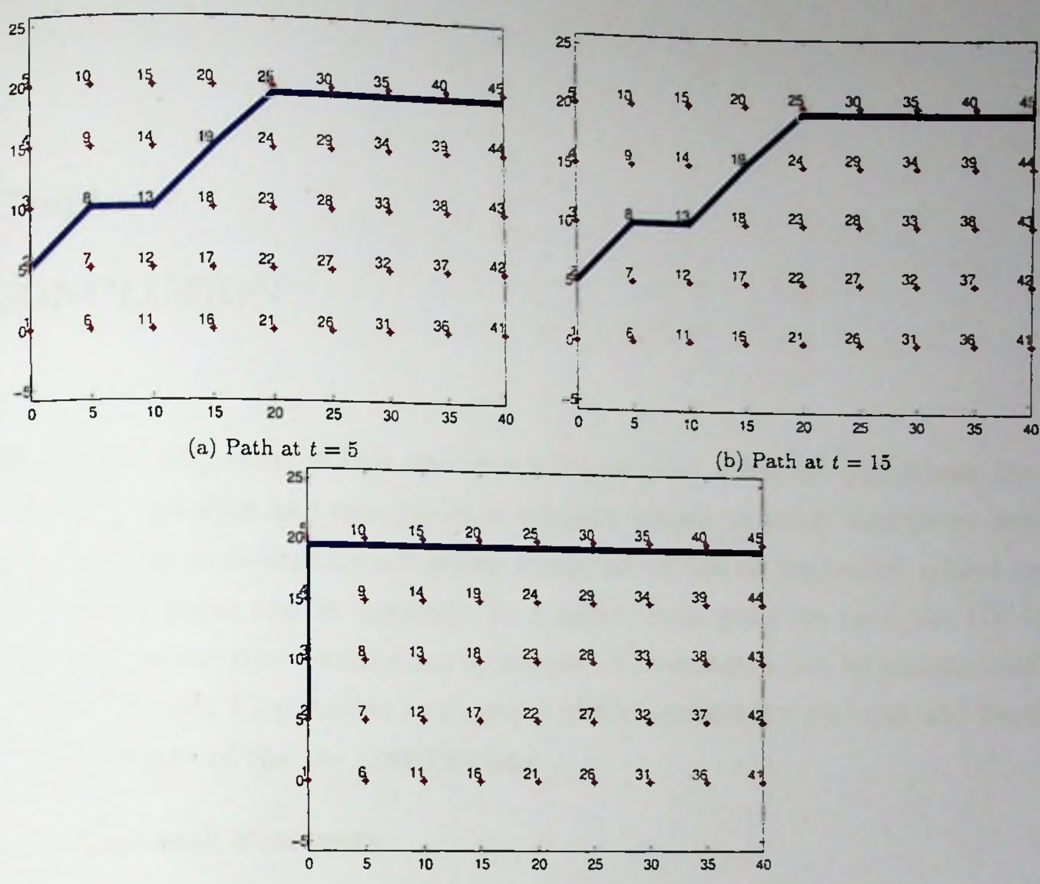


(c) At $t = 30$

Figure 6.5: Temperature Distribution

6.6. Based on the severity values this path will be changed as shown in Figure 6.6.

According to the Figure 6.6, it is very clear that depending on the severity index distribution, the safest path changes and fire fighters have to follow that



(a) Path at $t = 5$

(b) Path at $t = 15$

(c) Path at $t = 30$

Figure 6.6: Laying new sensors to emulate the lifeline

path. Computations to be done to find the path can be done at a central location which is under the control of the incident commander.

6.3.3 Conclusion

In this Chapter we studied the importance of localization and navigation support for emergency situations such as a fire. Further, we propose a new method to emulate the laying of lifeline using WSNs as a navigation support for fire fighters in emergency environments. Simulation results show that the proposed method can be practically applied in emergency situations as navigation guide for the fire fighters.

Chapter 7

CONCLUSION

The overall objective of this research is to develop a suit of algorithms for localization, tracking and navigation of wireless sensor nodes in multistory indoor environments in emergency situations where nodes can be destroyed, added anew and mobile nodes can be present. To achieve these goals we used the DV-Hop algorithm as our core localization technique. The research can be summarized as follows. Section 7.1 provides a summary of the research carried out and Section 7.2 a summary of the key contributions.

7.1 Research Summary

7.1.1 Introduction to Localization

Localization is the task of determining the physical coordinates of sensor nodes or spatial relationship among them. Work reported in this research describes what localization means and classifies the localization algorithms as centralized vs distributed, range based vs range free and event driven localization algorithms.

7.1.2 Study of the DV-Hop Algorithm

The work reported in this thesis uses the DV-Hop algorithm as the core. Even though the DV-Hop algorithm is an attractive option for the localization of nodes in a wireless sensor network due to its simplicity, it suffers from poor accuracy. This was due to several fundamental reasons that we investigated in depth during the research. Further, we evaluated the performances of the DV-Hop algorithm for its accuracy, communication cost and computational cost while varying in different network parameters, configurations and node positioning. Ultimately, we extended the 2D DV-Hop algorithm to 3D environments, and evaluated the performances.

Many scholars have paid their attention to improve the DV-Hop algorithm. In this thesis we have reported a critical literature survey and a comparative performance survey on a selected set of improved DV-Hop algorithms.

7.1.3 Proposed Improved DV-Hop Algorithms for Improved Localization

In this research we proposed three novel improved DV-Hop algorithms. The first two are based on using anchor position re-estimation in order to obtain a better estimate of the Hop Size. Simulation results show that both the proposed algorithms provide improvements over the DV-Hop algorithm. A performance comparison was provided. Localization error variance also is lower in both the proposed algorithms, and thus they ensure more steady performance.

The third proposed algorithm was a DV-Hop based localization scheme that can be used for localizing grid-based sensor networks. We assumed that the sensor network was deployed in a controlled manner, where the sensors are fixed randomly on a regular grid. Through simulations it was evident that the proposed algorithm can localize more nodes with zero localization error.

7.1.4 Proposed Target Tracking Algorithm

Target tracking is the function of tracking a moving object in a WSN. In this research, we proposed a new approach for target tracking in a wireless sensor network by combining the DV-Hop algorithm with Kalman filtering. The DV-Hop algorithm was used for pre-localization of the target and measurement conversion. The converted measurement and its associated noise statistics were then used in a standard Kalman filter for recursive update of the target state. Simulation results have shown that the proposed approach improves the tracking accuracy compared to the DV-Hop localization, while retaining the distributed nature of the algorithm.

7.1.5 Extending the DV-Hop Algorithm for Emergency Environments

Though localization is studied widely in literature, few such algorithms are deployed in emergency situations such as fire situations due to multiple challenges. We studied these key challenges for localization in an emergency environment where nodes can be destroyed, added a new and mobile nodes cab be presented.

Introducing new sensor nodes to the existing network is a common requirement in WSN related applications. Especially nodes with unknown locations can be newly added randomly for additional information gathering during an emergency situation such as a fire. Therefore these newly added nodes should be localized as quickly as possible with the best possible accuracy. In this research, we proposed a novel method to localize a newly introduced node with the help of the DV-Hop algorithm.

7.1.6 Proposed Navigation Support Technique for Fire Fighters

Here, we proposed a WSN based mechanism to emulate the technique of the lifeline, a physically laid rope used by the fire fighters in an emergency situation. Simulation results showed that the proposed method can be practically applied in emergency situations as a navigation guide for fire fighters.

7.2 Key Contributions

The key contributions of this research are:

- An in-depth analysis of the DV-Hop algorithm and its improvements
- Novel improvements to the DV-Hop algorithm
- A novel algorithm to localize newly introduced node to an existing network
- A novel tracking algorithm with the DV-Hop as the core
- A navigation support technique for mobile nodes

Numerical results of those findings can be summarized as follows.

Table 7.1 summarizes the performance comparison of the proposed novel DV-Hop based algorithms relative to the DV-Hop algorithm.

The two novel algorithms proposed in this research are based on using anchor position re-estimation in order to obtain a better estimate of the Hop Size. Through simulations it is evident that both the algorithms show improvement in accuracy over the DV-Hop algorithm. On average Proposed Algorithm 1 shows an accuracy improvement of 10% and Proposed Algorithm 2 shows an accuracy improvement of 7% over the DV-Hop algorithm.

The optimum Hop Size computation requires an iterative procedure. Thus, both algorithms need more computational power and time than the DV-Hop algorithm. Also, the computational cost of Algorithm 1 is around 10% higher than

Table 7.1: Performance Comparison of novel algorithms relative to the DV-Hop algorithm

Algorithm	Average Localization Error Improvement			Comp. Time	Com. Cost
	with anchors Figure 4.5(a)	with nodes Figure 4.6(a)	with range Figure 4.7(a)	with anchors Figure 4.8	
Algorithm 1 (Section 4.1)	11%	10%	11%	−15%	higher
Algorithm 2 (Section 4.2)	8%	6%	6%	−5%	higher

Algorithm 2. Algorithm 1 introduces a centralized processing component into the DV-Hop algorithm, while Algorithm 2 retains the original distributed nature. Thus, Algorithm 1 incurs a higher communications cost than Algorithm 2.

Table 7.2 summarizes the comparison of RMSE in position of the proposed tracking algorithm and [61] and Table 7.3 summarizes the compariosn of computational cost of the two algorithms in terms of the estimation time.

Table 7.2: Estimation error comparison of novel tracking algorithm (Radio Range = $22m$)

Nodes	100	150	200	250	300	350	400
RMSE (m) [61]	2.0756	1.5447	1.2377	1.0312	0.9135	0.8489	0.78888
RMSE (m) Proposed (Section 5.3)	1.8399	1.5776	1.5292	1.5161	1.5036	1.5038	1.5009

Table 7.3: Estimation time comparison of novel tracking algorithm (Radio Range = $22m$)

Nodes	100	150	200	250	300	350	400
Time (s) [61]	1.8711	2.0952	2.8768	3.4887	3.8933	4.9768	5.3460
Time (s) Proposed (Section 5.3)	0.0541	0.0839	0.1818	0.3482	0.5713	0.8941	1.3117

Through simulations it is evident that the estimation accuracy of the target trajectory of the proposed algorithm is lower than that of the [61]. However [61] needs around 15 times more computational cost due to its high complexity and also higher communication cost than our algorithm.

The proposed target tracking algorithm can be implemented as a fully distributed manner, where as [61] is a centralized algorithm. This is one of the key advantages of the proposed algorithm over the [61]. Further in [61] at least three sensors in the network should detect the moving target at once. Otherwise the algorithm fails to estimate the target states. However in the proposed tracking algorithm, it is enough to detect the target only by a one sensor in the network. Therefore the proposed algorithm is more suitable to apply in emergency environments where nodes can be destroyed, due to its distributive nature, simplicity and demand for fewer sensor nodes.

Both anchors and the other nodes in the network are destroyed due to the fire and when fire increases with time more and more nodes in the network will be destroyed. Until there is at least a single node that detects the moving target at a given time, the proposed algorithm can estimate the target states. In algorithm [61], when the sensor nodes in the existing network are gradually dying, estimation accuracy of the moving target reduces. However in the proposed algorithm, we do not have this problem severely as shown in Table 7.4. Therefore the proposed tracking algorithm is very much suitable for target tracking purposes in emergency environments.

Table 7.4: Estimation error comparison of novel tracking algorithm in an emergency environment (Radio Range = $22m$)

Radius of fire (m)	0	5	10	15	20	25	30
RMSE (m) [61]	1.2377	1.3347	1.4321	1.4603	1.5688	1.6721	1.7569
RMSE (m) Proposed (Section 5.3)	1.5292	1.5510	1.5510	1.5510	1.5569	1.5569	1.5569

7.3 Future Work

Some interesting areas of future research have become apparent during the course of this research. In this Section we briefly outline them.

Improving the target tracking accuracy further: In this research, for the first time in literature we combined the DV-Hop algorithm with Kalman filtering to produce a novel tracking algorithm. In future, researchers can find new ways of improving the estimation accuracy this proposed algorithm. Combining different filtering techniques such as Extended Kalman filtering and Particle filtering with the DV-Hop algorithm can be an interesting research area.

Navigation support for fire fighters: Navigation support for fire fighters in a fire field is a must. Fire fighters have developed navigation practices for use in poor visibility. Details vary, but overall, they use the same ideas worldwide. These methods tend to be simple and practical, and the equipment is seemingly low-tech and very robust. In this research we emulated the lifeline navigation technique used by fire fighters to find their way in a fire filed. Likewise in future, researches can combine the WSNs with navigation practices used by fire fighters for better navigation solutions.

Practical implementation of the proposed algorithms: Results of the algorithms presented in this thesis are based on Matlab simulations due to the lack of hardware facilities. However in practical environments, there can be changes in performances of the algorithms. Therefore in future, these proposed algorithms can be implemented in a real WSN for performance evaluation.

References

[1] Xirong, Bao, F., Zhang, S., Liu, L.: An improved dv-hop localization algorithm for wireless sensor networks. In: Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on. (Sept 2010) 1–4

[2] Waltenegus Dargie, C.P.: Fundamentals of Wireless Sensor Networks. John Wiley & Sons Ltd (2010)

[3] Sha, K., Shi, W., Watkins, O.: Using wireless sensor networks for fire rescue applications: Requirements and challenges. In: Electro/information Technology, 2006 IEEE International Conference on. (May 2006) 239–244

[4] Klann, M., Riedel, T., Gellersen, H., Fischer, C., Pirkl, G., Kunze, K., Beuster, M., Beigl, M., Visser, O., Gerling, M.: Lifenet: an ad-hoc sensor network and wearable system to provide firefighters with navigation support. `http://eis.comp.lancs.ac.uk/fileadmin/eis/publication/2007-LifeNet.pdf`

[5] Guoqiang Mao, B.F.: Localization Algorithms and Strategies for Wireless Sensor Networks. Information Science Reference - Hershey, New York (2009)

[6] Chauhan, P., Ahlawat, P.: Target tracking in wireless sensor network. International Journal of Information & Computation Technology **4**(6) (2014) 643–648

[7] Fischer, C., Gellersen, H.: Location and navigation support for emergency responders: A survey. IEEE Pervasive Computing **9**(1) (January 2010) 38–47

[8] Shang, Y., Ruml, W.: Improved mds-based localization. In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies. Volume 4. (March 2004) 2640–2651 vol.4

[9] Niculescu, D., Nath, B.: Ad hoc positioning system (aps). In: Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE. Volume 5. (2001) 2926–2931 vol.5

[10] Bulusu, N., Heidemann, J., Estrin, D.: Gps-less low-cost outdoor localization for very small devices. Personal Communications, IEEE **7**(5) (Oct 2000) 28–34

[11] Doherty, L., pister, K., El Ghaoui, L.: Convex position estimation in wireless sensor networks. In: INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Volume 3. (2001) 1655–1663 vol.3

[12] zeng Wang, J., Jin, H.: Improvement on apit localization algorithms for wireless sensor networks. In: Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09. International Conference on. Volume 1. (April 2009) 719–723

[13] Römer, K.: The lighthouse location system for smart dust. In: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services. MobiSys '03, New York, NY, USA, ACM (2003) 15–30

[14] Zhong, Z., He, T.: Msp: Multi-sequence positioning of wireless sensor nodes. In: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems. SenSys '07, New York, NY, USA, ACM (2007) 15–28

[15] Stoleru, R., He, T., Mathiharan, S., George, S., Stankovic, J.: Asymmetric event-driven node localization in wireless sensor networks. Parallel and Distributed Systems, IEEE Transactions on **23**(4) (April 2012) 634–642

[16] Ning Wang, X., Xu, X.: Asymmetric event-driven localization algorithm in constrained space. In: International Journal of Distributed Sensor Networks. (2013)

[17] Will, H., Hillebrandt, T., Yuan, Y., Yubin, Z., Kyas, M.: The membership degree min-max localization algorithm. In: Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), 2012. (Oct 2012) 1–10

[18] WANG, L., ZHANG, J., CAO, D.: An improved dv-hop localization algorithm for wireless sensor networks. In: Journal of Computational Information Systems Vol. 8 (6). (2012) 24632475, keywords=Dv–Hop Algo-

rithm;Mobile Agent; Coordinate Translation Method (CTM);Draw Circle Method (DCM);Total Least Squares (TLS) ,

[19] Chen, H., Sezaki, K., Deng, P., So, H.C.: An improved dv-hop localization algorithm for wireless sensor networks. In: Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on. (June 2008) 1557–1561

[20] Li, J., Zhang, J., Xiande, L.: A weighted dv-hop localization scheme for wireless sensor networks. In: Scalable Computing and Communications; Eighth International Conference on Embedded Computing, 2009. SCALCOM-EMBEDDEDCOM'09. International Conference on. (Sept 2009) 269–272

[21] Zhang, J., Li, W., Cui, D., Sun, X., Zhou, F.: Study on improved dv-hop node localization algorithm in wireless sensor network. In: Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on. (June 2010) 1855–1858

[22] Lee, J., Chung, W., Kim, E., Hong, I.W.: Robust dv-hop algorithm for localization in wireless sensor network. In: Control Automation and Systems (ICCAS), 2010 International Conference on. (Oct 2010) 2506–2509

[23] Hu, Y., Li, X.: An improvement of dv-hop localization algorithm for wireless sensor networks. In: Telecommunication Systems Volume 53, Issue 1. (May 2013) 13–18

[24] Shu XIANG, Yunzhou ZHANG, Z.X.L.X.W.F.: An improved dv-hop localization algorithm using residual weight in wireless sensor network. In: Journal of Computational Information Systems 8. (August 2012)

[25] Lin, J.Z., Chen, X.B., Liu, H.B.: Iterative algorithm for locating nodes in wsn based on modifying average hopping distances. In: Journal on Communications, vol. 30 No. 10. (October 2009) 107–113

[26] Chen, X., Zhang, B.: Improved dv-hop node localization algorithm in wireless sensor networks. In: International Journal of Distributed Sensor Networks, vol. 2012, Article ID 213980, 7 pages. (July 2012)

[27] Yin, M., Shu, J., Liu, L., Zhang, H.: The influence of beacon on dv-hop in wireless sensor networks. In: Grid and Cooperative Computing Workshops, 2006. GCCW '06. Fifth International Conference on. (Oct 2006) 459–462

[28] Zheng, Y., Wan, L., Sun, Z., Mei, S.: A long range dv-hop localization algorithm with placement strategy in wireless sensor networks. In: Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on. (Oct 2008) 1–5

[29] Fang, W., Yang, G.: Improvement based on dv-hop localization algorithm of wireless sensor network. In: Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference on. (Aug 2011) 2421–2424

[30] Ding, J., Zhang, L., Cheng, G., Ling, Z., Zhang, Z., Lei, Y.: Study on dv-hop algorithm based on modifying hop count for wireless sensor networks. In: International Journal of Computer Science Engineering and Technology (IJCSET). (Oct 2012)

[31] Yi, T., Fang, Z., Li, R.: Rmadv-hop: An improved dv-hop localization algorithm. In: Information Technology: New Generations (ITNG), 2010 Seventh International Conference on. (April 2010) 939–943

[32] Tian, S., Zhang, X., Liu, P., Sun, P., Wang, X.: A rssi-based dv-hop algorithm for wireless sensor networks. In: Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on. (Sept 2007) 2555–2558

[33] Wenjuan WANG, Yuwang YANG, Z.L.: An improved dv-hop algorithm for wireless sensor networks. In: Journal of Computational Information Systems 10. (March 2014)

[34] Kumar, S., Lobiyal, D.: An advanced dv-hop localization algorithm for wireless sensor networks. In: Wireless Personal Communications Volume 71, Issue 2. (2013, month=July, pages=1365-1385, keywords=Wireless sensor network;Localization;Weighted least square;DV-Hop algorithm;Correction factor;Random distribution ,)

[35] Wang, Y., Tian, W., Pang, Y., Liu, Y.: A new improved dv-hop localization algorithm. In: Communication Technology and Application (ICCTA 2011), IET International Conference on. (Oct 2011) 772–776

[36] Chen, K., hua Wang, Z., Lin, M., Yu, M.: An improved dv-hop localization algorithm for wireless sensor networks. In: Wireless Sensor Network, 2010. IET-WSN. IET International Conference on. (Nov 2010) 255–259

[37] Xu, D., Liu, C.: An improved dv-hop localization algorithm based-on local area in nonhomogeneous density wsn. In: Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on. (Dec 2010) 1–3

[38] Zhou, Z., Xiao, M., Liu, L., Chen, Y., Lv, J.: An improved dv-hop localization algorithm. In: Information Science and Engineering (ISISE), 2009 Second International Symposium on. (Dec 2009) 598–602

[39] Quanrui, W., Jiuqiang, H., Dexing, Z., Ruiling, L.: An improved multihop distance estimation for dv-hop localization algorithm in wireless sensor networks. In: Vehicular Technology Conference (VTC Fall), 2012 IEEE. (Sept 2012) 1–5

[40] Gao, G.Q., Lei, L.: An improved node localization algorithm based on dv-hop in wsn. In: Advanced Computer Control (ICACC), 2010 2nd International Conference on. Volume 4. (March 2010) 321–324

[41] Liu, J., Wang, W., Shang, Y.: An improving localization algorithm for wireless sensor networks based on dv-hop. In: Measurement, Information and Control (MIC), 2012 International Conference on. Volume 1. (May 2012) 511–515

[42] Gui, L., Val, T., Wei, A.: Improving localization accuracy using selective 3-anchor dv-hop algorithm. In: Vehicular Technology Conference (VTC Fall), 2011 IEEE. (Sept 2011) 1–5

[43] Agashe, A.A., Patil, R.S.: An optimum dv hop localization algorithm for variety of topologies in wireless sensor networks. In: International Journal on Computer Science & Engineering;Jun2012, Vol. 4 Issue 6. (June 2012) 957

[44] Xiaoyong YAN, Huanyan QIAN, J.Y.: A localization method based on principal component analysis. In: Journal of Computational Information Systems 8. (November 2012)

[45] Jiangwen LI, Liling ZHANG, Y.W.P.X.: A novel range-free localization algorithm based on the grid scanning method. In: Journal of Computational Information Systems 9. (September 2013)

[46] Keji MAO, Guanglin DAI, L.C.Q.S.L.W.Q.C.: Semi-auto dv-hop localization algorithm based area division in wsn. In: Journal of Computational Information Systems 9. (June 2013)

[47] Tsai, H.W., Chu, C.P., Chen, T.S.: Mobile object tracking in wireless sensor networks. Comput. Commun. **30**(8) (June 2007) 1811–1825

[48] Zhao, W., Han, Y., Wu, H., Zhang, D.: Weighted distance based sensor selection for target tracking in wireless sensor networks. Signal Processing Letters, IEEE **16**(8) (Aug 2009) 647–650

[49] Oh, S.: A scalable multi-target tracking algorithm for wireless sensor networks. International Journal of Distributed Sensor Networks (2012) 16

[50] Yan Zhou, Dongli Wang, T.P., Lan, Y.: Energy-efficient target tracking in wireless sensor networks: A quantized measurement fusion framework. International Journal of Distributed Sensor Networks (2014) 10

[51] Yeow, W.L., Tham, C.K., Wong, W.C.: Energy efficient multiple target tracking in wireless sensor networks. Vehicular Technology, IEEE Transactions on **56**(2) (March 2007) 918–928

[52] Teng, J., Snoussi, H., Richard, C., Zhou, R.: Distributed variational filtering for simultaneous sensor localization and target tracking in wireless sensor networks. Vehicular Technology, IEEE Transactions on **61**(5) (Jun 2012) 2305–2318

[53] Wang, G., Alam Bhuiyan, M., Cao, J., Wu, J.: Detecting movements of a target using face tracking in wireless sensor networks. Parallel and Distributed Systems, IEEE Transactions on **25**(4) (April 2014) 939–949

[54] Lin, J., Xiao, W., Lewis, F., Xie, L.: Energy-efficient distributed adaptive multisensor scheduling for target tracking in wireless sensor networks. Instrumentation and Measurement, IEEE Transactions on **58**(6) (June 2009) 1886–1896

[55] Xu, E., Ding, Z., Dasgupta, S.: Target tracking and mobile sensor navigation in wireless sensor networks. Mobile Computing, IEEE Transactions on **12**(1) (Jan 2013) 177–186

[56] Wang, X., Ma, J., Wang, S., Bi, D.: Distributed energy optimization for target tracking in wireless sensor networks. Mobile Computing, IEEE Transactions on **9**(1) (Jan 2010) 73–86

[57] Masazade, E., Fardad, M., Varshney, P.: Sparsity-promoting extended kalman filtering for target tracking in wireless sensor networks. Signal Processing Letters, IEEE **19**(12) (Dec 2012) 845–848

[58] Oka, A., Lampe, L.: Distributed target tracking using signal strength measurements by a wireless sensor network. Selected Areas in Communications, IEEE Journal on **28**(7) (September 2010) 1006–1015

[59] Wang, Y., Wang, D.: Energy-efficient node selection for target tracking in wireless sensor networks. International Journal of Distributed Sensor Networks (2013) 6

[60] Wang, X., Fu, M., Zhang, H.: Target tracking in wireless sensor networks based on the combination of kf and mle using distance measurements. Mobile Computing, IEEE Transactions on **11**(4) (April 2012) 567–576

[61] Djuric, P., Vemula, M., Bugallo, M.: Target tracking by particle filtering in binary sensor networks. Signal Processing, IEEE Transactions on **56**(6) (June 2008) 2229–2238

[62] Zhou, G., He, T., Krishnamurthy, S., Stankovic, J.A.: Impact of radio irregularity on wireless sensor networks. In: Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services. MobiSys '04, New York, NY, USA, ACM (2004) 125–138

[63] J. Z. Lin, X.B.C., Liu, H.B.: Iterative algorithm for locating nodes in wsn based on modifying average hopping distances. Journal on Communications **30**(10) (2009) 107–113

[64] Yaakov Bar-Shalom, X.-Rong Li, T.K.: Estimation with Applications to Tracking and Navigation. John Wiley & Sons Ltd (2001)

[65] Tseng, Y.C., Pan, M.S., Tsai, Y.Y.: A distributed emergency navigation algorithm for wireless sensor networks. IEEE Computers **39**(7) (2006) 55–62

[66] Li, S., Zhan, A., Wu, X., Yang, P., Chen, G.: Efficient emergency rescue navigation with wireless sensor networks. `http://www.iis.sinica.edu.tw/page/jise/2011/201101_04.pdf`

[67] Tseng, Y.C., Pan, M.S., Tsai, Y.Y.: Wireless sensor networks for emergency navigation. Computer **39**(7) (July 2006) 55–62

[68] Weeraddana, D., Walgama, K., Kulasekere, E. International Journal of Electrical, Computer, Electronics and Communication Engineering **7**(7) (2013) 434 – 440