

Implementation of Smart Antenna Based Secure Localization Systems for Wireless Sensor Networks

Thesis submitted by

Rathindra Nath Biswas

Doctor of Philosophy (Engineering)

Department of Electronics and Telecommunication Engineering
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India

2023

**JADAVPUR UNIVERSITY
KOLKATA, INDIA**

Index No. 216/19/E

1. Title of the thesis

Implementation of Smart Antenna Based Secure Localization Systems for Wireless Sensor Networks

2. Name, Designation & Institution of the Supervisor/s

- (i) Prof. (Dr.) Mrinal Kanti Naskar
Professor
Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata - 700032

- (ii) Prof. (Dr.) Swarup Kumar Mitra
Professor
Department of Electronics and Communication Engineering
MCKV Institute of Engineering
Howrah - 711204

3. List of Publications

International Conferences

- (i) **R. N. Biswas**, A. Saha, S. K. Mitra, M. K. Naskar, “Realization of Adaptive Beamforming in Smart Antennas on a Reconfigurable Architecture”, *In Proceedings of the IEEE International Conference on Emerging Trends in Electronic Devices and Computational Techniques (EDCT 2018)*, pp. 1-7. Kolkata, March 2018.

National Conferences

Nil

International Journals

- (i) **R. N. Biswas**, S. K. Mitra, M. K. Naskar, “A Robust Mobile Anchor Based Localization Technique for Wireless Sensor Networks using Smart Antenna”, *International Journal of Ad-Hoc and Ubiquitous Computing (IAHUC)*, Vol. 15, No. 1/2/3, pp. 23-37, Published by Inderscience Enterprises Ltd., Switzerland, 2014 (**Impact Factor: 0.654, SCI Indexed**).
- (ii) **R. N. Biswas**, S. K. Mitra, M. K. Naskar, “Wireless Node Localization under Hostile Radio Environment using Smart Antenna”, *Wireless Personal Communications (WIRE)*, Vol. 116, pp. 1815-1836, Published by Springer Nature, Switzerland, 2021 (**Impact Factor: 1.671, SCI Indexed**).
- (iii) **R. N. Biswas**, S. K. Mitra, M. K. Naskar, “Localization under Node Capture Attacks using Fuzzy Based Anchor Mobility Control”, *Journal of Ambient Intelligence and Humanized Computing (AIHC)*, <https://doi.org/10.1007/s12652-021-03619-6>, Published by Springer Nature, Switzerland, 2022 (**Impact Factor: 7.104, SCI Indexed**).
- (iv) **R. N. Biswas**, A. Saha, S. K. Mitra, M. K. Naskar, “FPGA Implementation of Adaptive Beamforming in Smart Antenna for Secure Localization Applications”, *Wireless Personal Communications (WIRE)*, Springer (communicated).
- (v) **R. N. Biswas**, A. Saha, S. K. Mitra, M. K. Naskar, “Design and Implementation of Anchor Coprocessor Architecture for Wireless Node Localization Applications”, *Peer-to-Peer Networking and Applications (P2P)*, Springer (communicated).

Book Chapters

- (i) **R. N. Biswas**, S. K. Mitra, M. K. Naskar, “Preserving Security of Mobile Anchors Against Physical Layer Attacks: A Resilient Scheme for Wireless Node Localization” In M. T. Banday (Ed.), *Cryptographic Security Solutions for the Internet of Things*, pp. 211-243, Published by IGI Global, Hershey, PA, 2019.
- (ii) **R. N. Biswas**, S. K. Mitra, M. K. Naskar, “Energy Efficient and Secure Localization in Wireless Sensor Networks: An Approach Through Anchor Mobility Control”, In S. Shandilya, S. K. Shandilya, T. Thakur, A. K. Nagar (Eds.), *Novel Advancements in Electrical Power Planning and Performance*, pp. 250-282, Published by IGI Global, Hershey, PA, 2020.

4. List of Patents

Nil

5. List of Presentations in National/International/Conferences/Workshops

- (i) **R. N. Biswas**, A. Saha, S. K. Mitra, M. K. Naskar, “Realization of Adaptive Beamforming in Smart Antennas on a Reconfigurable Architecture”, In *Proceedings of the IEEE International Conference on Emerging Trends in Electronic Devices and Computational Techniques (EDCT 2018)*, pp. 1-7. Kolkata, March 2018.

Statement of Originality

I Sri **Rathindra Nath Biswas** registered on **16th July, 2019** do hereby declare that this thesis entitled “**Implementation of Smart Antenna Based Secure Localization Systems for Wireless Sensor Networks**” contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work.

I also declare that I have checked this thesis as per the “Policy on Anti Plagiarism, Jadavpur University, 2019”, and the level of similarity checked by iThenticate software is **3%**.

Signature of Candidate:


Rathindra Nath Biswas

Date:

Certified by Supervisor (s):

(1) Prof. (Dr.) Mrinal Kanti Naskar
Professor
Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata - 700032


15.06.23


M. K. Naskar
Professor
Department of Electronics & Telecommunication Engg.
Jadavpur University, Kolkata-700 032

(2) Prof. (Dr.) Swarup Kumar Mitra
Professor
Department of Electronics and Communication Engineering
MCKV Institute of Engineering
Howrah - 711204

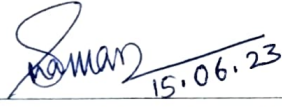

15.06.2023


Dr. Swarup Kumar Mitra
Professor, Department of ECE
MCKV Institute of Engineering, Howrah

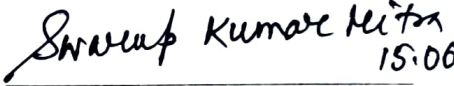
Certificate from the Supervisor/s

This is to certify that the thesis entitled “**Implementation of Smart Antenna Based Secure Localization Systems for Wireless Sensor Networks**” submitted by **Shri Rathindra Nath Biswas**, who got his name registered on **16th July, 2019** for the award of Ph. D. (Engg.) degree of Jadavpur University is absolutely based upon his own work under the supervision of **Prof. Mrinal Kanti Naskar** and **Prof. Swarup Kumar Mitra** and neither his thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

Signature of the Supervisor (s):

1. 
15.06.23
Prof. (Dr.) Mrinal Kanti Naskar
Professor
Department of ETCE
Jadavpur University
Kolkata - 700032


Professor
Deptt. of Electronics & Telecommunication Engg.
Jadavpur University, Kolkata-700 032

2. 
15.06.2023
Prof. (Dr.) Swarup Kumar Mitra
Professor
Department of ECE
MCKV Institute of Engineering
Howrah - 711204


Professor, Department of ECE
MCKV Institute of Engineering, Howrah

Acknowledgements

I would like to express my sincere gratitude and thanks to my supervisor, Prof. Mrinal Kanti Naskar, for his invaluable advice, continuous support and encouragement during my research work. I appreciate all of his immense knowledge, contributions of time, suggestions, motivations and patience for my research. He provided me with all facilities to carry out the research in his Advanced Digital and Embedded Systems Laboratory at Jadavpur University. I am also extremely grateful to my co-supervisor, Prof. Swarup Kumar Mitra for his tremendous support throughout my research. He always motivated me and made himself available to clarify my doubts. Without his help, it would not be possible to conduct this research. I acknowledge it as a great opportunity to do my PhD programme under their guidance and learn from their research expertise.

I gratefully acknowledge the contributions of UG student from Jadavpur University, Mr. Anurup Saha (now PhD Scholar in Department of Electrical and Computer Engineering, Georgia Institute of Technology, USA), for his assistance in simulating the proposed algorithms at the hardware level.

I would like to thank Prof. Manotosh Biswas, Prof. Ananda Sankar Chowdhury and Prof. Sheli Sinha Chowdhury, who were associated as Head of Electronics and Telecommunication Engineering Department at Jadavpur University in different phases of my PhD course, for supporting me with valuable suggestions towards the progress of my research. I would like to thank all the members of the doctoral research committees for their insightful comments which inspired me to broaden my research from various angles.

I would like to express my whole-hearted thanks to Mr. Debaprasad De, Mr. Pulak Majumdar, Mrs. Chandrani Sadhukhan, Mr. Sekhar Rana, Dr. S. K. Sahil Babu, Dr. Rajarshi Middy, Dr. Arpita Chakraborty and Mr. Asim Maity for sharing their valuable knowledge, insights and useful discussions whenever necessary.

I warmly thank Mr. Jayanta Ganguli, Project Manager, Techno Electric & Engineering Company Ltd. for his support, advice and encouragement throughout my research work. I would also like to appreciate the help and support received from all my colleagues from Acharya Jagadish Chandra Bose Polytechnic, Berachampa and Central Calcutta Polytechnic, Kolkata.

I am deeply indebted to my parents, Late Rajendra Nath Biswas and Mrs. Kanak Prova Biswas, my uncle and aunt Mr. Natobar Adhikary and Mrs. Mili Adhikary for their unconditional love, support, patience, tolerance and encouragement for my research. I gratefully acknowledge the contributions of my friends Mr. Gour Chandra Saha, Mr. Suvas Kumar Sakar, Dr. Supriya Dhabal and Dr. Sk Intiaj, who have continuously motivated me even in hard times.

Finally, I would like to thank my wife Mrs. Piyali Biswas and my beloved son Mr. Praneel Biswas for their invaluable patience, endless support and encouragement, without which it would not be possible to conduct this research.

Jadavpur University, Kolkata


Rathindra Nath Biswas 15/06/2023

Dedicated to two holy souls of late
Ratan Ganguly and late Asima Ganguly
in the divine

Abstract

Wireless Sensor Networks (WSNs) consist of low cost, tiny sensors (nodes) with limited processing and communication capabilities. It does not require huge infrastructure and a central supervision to set up such a network. Therefore, a large number of nodes are often randomly deployed over the region of interest on an ad-hoc basis to achieve this. For these unique features, WSNs have now been the best choice for many applications, especially in a remote and hostile area. For example, a variety of applications such as data gathering, monitoring of the disaster prone areas (flood, forest fire and volcanic eruption, etc.), wildlife monitoring, industrial control, healthcare, surveillance and security of military operations, and smart agriculture, etc. are very common. However, the nodes do not know their locations in the network and act cooperatively with each other. For the realization of an event occurring, such applications require location data to be relayed over the network from the source nodes to a sink or base station. The incorporation of a localization system is often desirable, enabling each node to autonomously determine its precise location within the network. Most of the state-of-the-art localization systems work with three main components, such as

- *distance/angle estimation* based on Received Signal Strength (RSS), Time of Arrival (ToA), Time Difference of Arrival (TDoA) and Angle of Arrival (AoA),
- *position computation* based on estimated distance/angle data and known anchor references (anchors, also called beacons, are a few location aware nodes with GPS receivers), and a
- *localization algorithm* propagates the information depending on its operating principle (centralized/distributed), to estimate the positions of the remaining nodes.

All nodes and anchors relay data packets over a common wireless medium. Thus, they are vulnerable to malicious attacks. In the literature, several attacks like Sybil, replay, and wormhole, etc. are very common. Adversaries can compromise a few nodes to measure localization data such as, RSS, ToA, TDoA, and AoA from received data packets. Moreover, they can modify the radio environment in a particular region of the network. This would make the distance/angle estimation erroneous to nodes located there. Eventually, the proliferation of such erroneous information from these compromised/affected nodes may mislead the plan/decision to be made at the base station.

Over the past decade, several pioneering research works have reported many algorithms to mitigate such attacks. Either they use cryptographic principles or they detect and eliminate misbehaviors in malicious nodes. Cryptographic approaches may not be suitable for nodes with limited resources (nodes are battery powered and battery replacement or recharging is not possible). These are also fragile at compromised nodes and secret keys/passwords may be accessible to attackers. On the other hand, non-cryptographic methods seem to be more reliable in a hostile environment. These are robust enough to keep data packets secure in the application layer or other inner layers. But, do not perform well in physical layer as nodes are exposed to adversaries while sending data packets over radio. Therefore, implementing resilient localization systems that preserve data packet security in the physical (PHY) layer still poses a great challenge to researchers.

In this context, the direction finding capability of a smart antenna can be more effective in estimating RSS/AoA with high precision. A smart antenna exploits spatial diversity to counteract multipath fading under a changeable and unpredictable wireless channel. Its adaptive beamforming attribute can preserve data integrity and improve quality of services over point-to-point communication links. Integrating smart antennas with anchors does not impose additional overhead on nodes and proves to be a new idea for secure localization applications in WSNs. It is beneficial to isolate malicious nodes in the network via antenna pattern control, making the localization process as secure and robust as possible. This is an open research area and so far no research efforts have been reported in this direction. For this reason, we are undertaking our research work based on smart antennas to develop secure localization systems which should have a lot of potential for use in the near future.

In this thesis, we have claimed the benefits of secure localization systems based on smart antennas over conventional methods, in terms of energy efficiency and localization accuracy. In this context, we discussed the different types of possible attacks and their effects, the principles of preserving security against them, and the challenges of secure localization in WSNs. We briefly explained the two important features of smart antennas, namely adaptive beamforming and direction finding capabilities, and emphasized its use for secure localization applications.

In *Chapter 1*, we discussed the overview of the problem, the objective and the motivation of this research work.

In *Chapter 2*, we briefly reviewed some common security attacks in WSNs and performed an extensive literature survey of existing secure localization algorithms for application layer or other inner layers. This chapter also contains the study of a few existing secure localization methods for the physical layer.

In *Chapter 3*, we introduced two anchor mobility control strategies (centroid and fuzzy) for the realization of resilient localization systems in WSNs. We presented a robust localization system based on smart antennas called Row Matching Algorithm and evaluated its performance using different anchor mobility models (e.g., random, centroid, and fuzzy).

In *Chapter 4*, we presented a secure localization system using a fuzzy based anchor mobility control strategy to mitigate node capture attacks in WSNs.

In *Chapter 5*, we also presented a secure localization system using a consistent variant assortment strategy to be effective in a tampered radio environment in WSNs.

In *Chapter 6*, we presented a security framework to protect anchor data using the strategy of antenna pattern control and pseudo-reference generation.

In *Chapter 7*, we presented the methodology for implementing a reconfigurable beamforming architecture on an FPGA (Field Programmable Gate Array) board to embed smart antennas with anchors.

In *Chapter 8*, we also presented the methodology for implementing a coprocessor architecture on an FPGA board for the anchors to compute the location of the nodes centrally.

In *Chapter 9*, we concluded the thesis by summarizing the overall findings. We have also included suggestions for the future direction of research in this area.

We have listed the main contributions of the thesis, in brief, as follows.

- Studied in detail existing secure localization algorithms and explained the requirement for smart antenna based security solutions in WSNs.
- Proposal of two anchor mobility control strategies (centroid and fuzzy) for the realization of resilient localization systems in WSNs.
- Proposal of two secure localization algorithms (via anchor mobility control, and consistent variant assortment) against two different attack scenarios in WSNs.
- Proposed security strategy for anchor data against PHY layer attacks in WSNs via antenna pattern controls.
- Proposed methodologies for implementing smart antenna beamformer and anchor coprocessor architectures on an FPGA platform.
- Verification of the operational feasibility of the proposed secure localization models based on smart antennas compared to conventional algorithms, in terms of energy efficiency and localization accuracy using simulations and performance analyses.

Contents

<i>Acknowledgments</i>	v
<i>Abstract</i>	vii
<i>List of Figures</i>	xiv
<i>List of Tables</i>	xvi
<i>List of Algorithms</i>	xvii
<i>List of Abbreviations</i>	xviii
<i>List of Symbols</i>	xx
1 Introduction	1
1.1 Wireless sensor networks	1
1.1.1 Advantages of WSNs	1
1.1.2 Challenges of WSNs	2
1.1.3 Types of WSNs	2
1.1.4 Applications of WSNs	3
1.1.5 Basics of MICAz motes	3
1.2 Overview of the problem	4
1.3 Motivation and objective of the thesis	5
1.3.1 Importance of localization systems	6
1.3.2 Features of localization systems	7
1.3.3 Basics of localization systems	7
1.3.4 Components of localization systems	7
1.3.5 Types of localization systems	8
1.3.6 Performance of localization systems	9
1.3.7 Security issues in localization systems	9
1.3.8 Importance of secure localization systems	10
1.3.9 Strategies for secure localization systems	10
1.3.10 Challenges for secure localization systems	11
1.3.11 Concept of smart antennas in localization systems	11
1.3.12 Secure localization systems based on smart antennas	13
1.4 Justification of this work	15
1.5 Contribution of the thesis	15
1.6 Organization of the thesis	16
2 Literature survey	18
2.1 Introduction	18
2.2 Distance/angle estimation techniques	18
2.2.1 Received signal strength	18
2.2.2 Time of arrival	19

2.2.3	Time difference of arrival	20
2.2.4	Angle of arrival	20
2.2.5	Communication range	20
2.3	Position computation techniques	21
2.3.1	Trilateration	21
2.3.2	Multilateration	22
2.3.3	Triangulation	23
2.3.4	Bounding box	24
2.3.5	Probabilistic approaches	25
2.4	Overview of localization systems	25
2.4.1	Anchor-free	25
2.4.2	Anchor-based	26
2.4.3	Range-free	27
2.4.4	Range-based	27
2.4.5	Centralized	27
2.4.6	Distributed	28
2.4.7	Single-hop	28
2.4.8	Multi-hop	29
2.4.9	Directional/smart antenna-based	29
2.5	Fragility of localization systems	29
2.5.1	Attacks on distance/angle estimation	29
2.5.2	Attacks on position computation	30
2.5.3	Attacks on localization algorithm	30
2.6	Overview of secure localization systems	30
2.6.1	Cryptography	31
2.6.2	Anomaly detection	31
2.6.3	Location verification	32
2.6.4	Robust computation	32
2.6.5	Simple systems with extra hardware	32
2.7	Summary	33
3	Resilient localization and anchor mobility control strategies	34
3.1	Introduction	34
3.2	System design preliminaries	35
3.2.1	Network architectures and protocols	35
3.2.2	Radio propagation characteristics	35
3.3	Row matching algorithm	37
3.3.1	Cluster formation	37
3.3.2	Beacon message generation/transmission	37
3.3.3	Position computation	38
3.4	Mobility control strategies	42
3.4.1	Centroid-based strategy	43
3.4.2	Fuzzy-based strategy	45
3.5	Performance evaluation	45
3.5.1	Performance metrics	49
3.5.2	Simulation environments	50
3.5.3	Simulation results	50
3.5.4	Performance analysis	50
3.6	Summary	53
4	Secure localization under node capture attacks via anchor mobility control	54
4.1	Introduction	54

4.2	Attack scenarios	55
4.3	Security frameworks	56
4.3.1	Suspicious node identification	57
4.3.2	Malicious node elimination	59
4.4	Localization system	60
4.4.1	Mobility controls	61
4.4.2	Position computation	61
4.4.3	Design methodology	63
4.5	Performance evaluation	65
4.5.1	Performance metrics	67
4.5.2	Simulation environments	68
4.5.3	Simulation results	68
4.5.4	Performance analysis	69
4.6	Summary	71
5	Secure localization under hostile radio environments via a consistency check	73
5.1	Introduction	73
5.2	Attack scenarios	74
5.3	Consistent variant assortment	75
5.4	Localization system	77
5.4.1	Beacon message generation/transmission	77
5.4.2	Position computation	78
5.4.3	Design methodology	79
5.5	Performance evaluation	81
5.5.1	Performance metrics	82
5.5.2	Simulation environments	82
5.5.3	Simulation results	83
5.6	Summary	87
6	Anchor data security under PHY layer attacks via antenna pattern control	88
6.1	Introduction	88
6.2	Attack scenarios	89
6.2.1	Direct attacks	89
6.2.2	Indirect attacks	89
6.3	Security criterion	89
6.4	Security framework	90
6.4.1	Antenna pattern controls	91
6.4.2	Pseudo reference generation	95
6.5	Performance evaluation	95
6.5.1	Performance metrics	95
6.5.2	Simulation environments	97
6.5.3	Simulation results	97
6.5.4	Performance analysis	98
6.6	Summary	100
7	Implementation of a smart antenna beamformer architecture on FPGA	101
7.1	Introduction	101
7.2	Design preliminaries	102
7.2.1	Field programmable gate array	102
7.2.2	Coordinate rotation digital computer	102
7.3	Design methodology	105
7.3.1	Reference template	106
7.3.2	Array steering vector	106

7.3.3	Optimal pattern	108
7.4	System implementation	110
7.4.1	Finite state machine with datapath	110
7.4.2	Proposed architecture	114
7.4.3	FPGA design flow	115
7.5	Performance evaluation	118
7.5.1	Performance metrics	118
7.5.2	Simulation environments	119
7.5.3	Simulation results	120
7.5.4	Performance analysis	124
7.6	Summary	124
8	Implementation of an anchor coprocessor architecture on FPGA	125
8.1	Introduction	125
8.2	Coprocessor features	126
8.3	Design methodology	127
8.3.1	Cluster formation	127
8.3.2	Distance and angle (image) estimation	127
8.3.3	Position computation	128
8.4	System implementation	129
8.4.1	Finite state machine with datapath	132
8.4.2	Proposed architecture	134
8.5	Performance evaluation	137
8.5.1	Performance metrics	137
8.5.2	Simulation environments	139
8.5.3	Simulation results	139
8.5.4	Performance analysis	140
8.6	Summary	144
9	Conclusion	145
9.1	Thesis outcomes	145
9.2	Future scope	147
9.3	Summary	147
	Bibliography	148

List of Figures

1.1	WSN architecture for a specific task	2
1.2	MICAz mote and its operations	4
1.3	Localization system architecture	8
1.4	Block diagram of an adaptive array antennas	12
2.1	Signal strength variation between source and sink nodes	19
2.2	Signal propagation time between source and sink nodes	19
2.3	Difference in signal propagation time between source and sink nodes	20
2.4	Direction of arrival of signal between source and sink nodes	21
2.5	Communication range between nodes	21
2.6	Position computation with trilateration	22
2.7	Position computation with multilateration	23
2.8	Position computation with triangulation	24
2.9	Position computation with bounding box	25
2.10	Position computation with probabilistic method	26
3.1	Functional blocks of the MICAz mote	36
3.2	Format of the beacon message	38
3.3	An adaptive beam pattern of smart antenna	39
3.4	Original and imaginary node positions at two anchor points	40
3.5	Position computation of nodes in RMA	41
3.6	Synchronizing pulse for mode switching	42
3.7	Data flow among various components of anchor	46
3.8	Input/output membership functions	47
3.9	Control surfaces of the output variables	47
3.10	Error, energy consumption and time for localization in each node	51
3.11	Average error, energy consumption and time for localization in WSNs	52
3.12	Number of localized and clustering nodes at each anchor point	52
4.1	Anchor trajectories in WSN	55
4.2	Attack scenarios	56
4.3	Segregation of benevolent nodes	59
4.4	Removal of malicious nodes	60
4.5	Cluster formation in two successive anchor points	61
4.6	Anchor mobility control strategies for secure localization	62
4.7	Format of the beacon message	63
4.8	Triangulation process for position computation	63
4.9	Trigger pulses for mode switching in anchor/node	65
4.10	State diagrams illustrating the node/anchor behaviors	66
4.11	Localization error for each node	69
4.12	Average localization error	69
4.13	Energy consumption for each node	70
4.14	Average energy consumption	70
4.15	Localization time for each node	70

4.16	Average localization time	70
4.17	Malicious node discovery rate	70
4.18	Benevolent node discovery rate	70
4.19	Average localization error	71
4.20	Malicious node discovery rate	71
5.1	Network architecture with attack scenarios	74
5.2	RSS/AoA estimation in corrupted radio	75
5.3	CVA for position estimation in corrupted radio	77
5.4	Format of a beacon message	77
5.5	Triangulation method for position computation	80
5.6	Trigger pulses for mode switching	81
5.7	FSM for anchor and node	81
5.8	Localization error for each node	83
5.9	Average localization error in WSN	84
5.10	Energy consumption for each node	85
5.11	Average energy consumption in WSN	85
5.12	Localization time for each node	86
5.13	Average localization time in WSN	86
6.1	A linear and symmetric array	92
6.2	A reference template	94
6.3	PSO-based pattern controls	98
6.4	Efficiency in pattern generation	99
6.5	Success rate to secure anchor data	100
7.1	FPGA Virtex-5 board and its functionality	103
7.2	Rotating a vector in a 2-D plane	103
7.3	Smart antenna beamforming system	106
7.4	State diagram of reference template module	111
7.5	State diagram of CORDIC module	111
7.6	State diagram of the PSO module	112
7.7	RTL schematic of the Template module	115
7.8	RTL schematic of the CORDIC module	116
7.9	RTL schematic of the PSO module	117
7.10	Block schematic of FPGA design flow	118
7.11	Optimal patterns for a cluster	121
7.12	Convergence in the beamforming process	122
7.13	Efficiency of the beamforming process	122
8.1	Functional diagram of the anchor coprocessor	126
8.2	Generation of original and imaginary position for nodes	128
8.3	Computation process in position estimation for nodes	129
8.4	State diagram of cluster formation module	132
8.5	State diagram of distance and angle (image) estimation module	132
8.6	State diagram of the CORDIC module	133
8.7	State diagram of position computation module	133
8.8	RTL schematic of cluster formation module	134
8.9	RTL schematic of distance/angle (image) calculation module	136
8.10	RTL schematic of CORDIC evaluation module	137
8.11	RTL schematic of position computation module	138
8.12	Average localization accuracy	140
8.13	Number of anchor points	141

List of Tables

3.1	Clock time allotted for various modes	42
3.2	Control signal input for various modes	42
3.3	Rule base for anchor displacement (d_a)	46
3.4	Rule base for anchor direction (ϕ)	46
3.5	Simulation parameters	50
3.6	Comparison of error and energy consumption in localization	51
4.1	Anchor behaviors in the FSM	66
4.2	Node behaviors in the FSM	67
4.3	Simulation parameters	68
4.4	Comparison of localization error and simulation time	71
4.5	Comparison of detection rate and time complexity	71
5.1	Anchor behaviors in FSM	81
5.2	Node behaviors in FSM	82
5.3	Simulation parameters	83
5.4	Results obtained with different beacons	84
5.5	Comparison of results	87
6.1	Simulation parameters	97
6.2	Antenna design parameters and pattern attributes	98
6.3	Optimized element excitation coefficients ($W_{g_{best}}$)	99
7.1	Beamforming processor behaviors in the FSM	112
7.1	Beamforming processor behaviors in the FSM	113
7.1	Beamforming processor behaviors in the FSM	114
7.2	Simulation parameters	119
7.3	Optimal values of the weight vector (W_n)	120
7.4	Accuracy and convergence of the beamforming process	120
7.5	FPGA resource utilization	122
7.6	Computation time	123
7.7	Power consumption	123
7.8	Comparison of FPGA resource utilization	123
7.9	Comparison of computation time and power consumption	124
8.1	Anchor coprocessor behaviors in the FSM	135
8.2	Simulation parameters	139
8.3	Accuracy and anchor points in the localization process	141
8.4	FPGA resource utilization	142
8.5	Computation time	142
8.6	Power consumption	142
8.7	Comparison of hardware resource utilization and localization accuracy	143
8.8	Comparison of computation time and power consumption	143

List of Algorithms

1	Pseudo-code for forming clusters and keeping their databases	38
2	Pseudo-code for generating and transmitting beacon messages	39
3	Pseudo code for accumulating beacon messages and computing position in the node	41
4	Pseudo-code for centroid based anchor mobility control	44
5	Pseudo-code for fuzzy based anchor mobility control	48
6	Pseudo code for discriminating suspicious and benevolent nodes	58
7	Pseudo-code for generating and transmitting beacon messages	64
8	Pseudo code to accumulate beacon messages and compute position in the node . .	65
9	Pseudo-code for consistent variant assortment process in node position estimation	76
10	Pseudo-code for generating and transmitting beacon messages	78
11	Pseudo code for accumulating beacon messages in each node	79
12	Pseudo-code for estimating positions in each node	80
13	Pseudo-code for particle swarm optimization	93
14	Pseudo-code for generating the optimal pattern	96
15	Pseudo-code for realizing the sine/cosine function in CORDIC	105
16	Pseudo-code to generate the reference template	107
17	Pseudo-code for constructing an array steering vector	108
18	Pseudo-code for generating the optimal pattern	109
19	Pseudo-code for forming clusters and keeping their databases	127
20	Pseudo-code for estimating distance and angle (image) of clustering nodes	128
21	Pseudo-code for computing the position of clustering nodes	130
22	Pseudo-code to realize the square root function using the Babylonian method . . .	131
23	Pseudo-code to realize the Modulus function	131

List of Abbreviations

Acronym	Description	Acronym	Description
2-D	Two Dimension	3-D	Three Dimension
A/D	Analog to Digital	AoA	Angle of Arrival
APIT	Approximate Position In Triangulation	ASIC	Application Specific Integrated Circuits
AWGN	Additive White Gaussian Noise	BLE	Basic Logic Elements
BRAM	Block Random Access Memory	BS	Base Station
CAD	Computer Aided Design	CG	Conjugate Gradient
CLB	Configurable Logic Blocks	CMA	Constant Modulus Algorithm
CMOS	Complementary Metal Oxide Semiconductor	CoG	Center of Gravity
CORDIC	Coordinate Rotation Digital Computer	CPU	Central Processing Unit
CVA	Consistent Variant Assortment	DGL	Distributed Grid-based Localization
DoA	Direction of Arrival	DRL	Distributed Range-free Localization
DSSS	Direct Sequence Spread Spectrum	DV	Distance Vector
ECDF	Empirical Cumulative Distribution Function	EDIF	Electronic Design Interchange Format
EEPROM	Electrically Erasable Programmable Read-Only Memory	EIRP	Effective Isotropic Radiated Power
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Technique	FCFS	First Come First Served
FF	Flip-flops	FM	Frequency Modulation
FNBW	First Null Beamwidth	FPGA	Field Programmable Gate Arrays
FSM	Finite State Machine	FSMD	Finite State Machine with Datapath
GPS	Global Positioning System	HDL	Hardware Description Language
HiRLoc	High-resolution Robust Localization	I/O	Input/Output
ID	Identity	IG	Inertial Guidance
IOB	Input-Output Block	ISM	Industrial, Scientific and Medical
LBSA	Simulated Annealing based Localization	LEACH	Low Energy Adaptive Clustering Hierarchy
LFSR	Linear Feedback Shift Register	LMS	Least Mean Squares
LoS	Line of Sight	LUT	Lookup Table
MAC	Multiplier-Accumulator	MAC	Media Access Control
MAP	Mobile Assisted Programming	MDS	Multi-Dimensional Scaling

MIMO	Multiple Input-Multiple Output	MMSE	Minimum Mean Square Error
MSE	Mean Square Error	MUSIC	Multiple Signal Classification
MUX	Multiplexers	NLoS	Non-Line of Sight
PC	Personal Computer	PHY	Physical Layer
PSO	Particle Swarm Optimization	QoS	Quality of Service
RF	Radio Frequency	RLS	Recursive Least Squares
RM	Rotation Mode	RMA	Row Matching Algorithm
RSS	Received Signal Strength	RSSI	Received Signal Strength Indicator
RTL	Register Transfer Level	SDMA	Spatial Division Multiple Access
SDN	Software-Defined Networking	SDP	Semi Definite Programming
SeRLoc	Secure Range-Independent Localization	SINR	Signal-to-Interference-plus- Noise Ratio
SIR	Signal-to-Interference Ratio	SLL	Sidelobe Level
SMI	Sample Matrix Inversion	SNR	Signal-to-Noise Ratio
SoC	System-on-Chip	SR	Success Rate
SRAM	Static Random Access Memory	TDoA	Time Difference of Arrival
TLS	Total Least Squares	ToA	Time of Arrival
ToF	Time of Flight	TPR	True Positive Rate
UWB	Ultra Wideband	VM	Vectoring Mode
WPAN	Wireless Personal Area Network	WSN	Wireless Sensor Network

List of Symbols

Notation	Description	Notation	Description
$r(t)$	Reference signal	$e(t)$	Error signal
$x(t)$	Input signal	$y(t)$	Output signal
$s(t)$	Desired signal	$i(t)$	Interfering signal
$\eta(t)$	Noise signal	$x_s(t), x_i(t)$	Desired and interference parts of input signals
θ_d	Direction of the desired signal	θ_n	Direction of interfering signals
θ	Angle of arrival of intercepted signals	θ'	Angle (image) of the intercepted signals
θ_p	Direction of highest priority node	φ	Angle estimated from received signals
d_a	Displacement of new anchor point	ϕ	Direction of new anchor point
D	Number of interfering signals	D'	Number of signal sources/received
W	Array weight vector	$\bar{a}(\theta)$	Array steering vector
d	Distance between two nodes	d_0	Reference distance from source
d_{max}	Maximum communication range	d_p	Distance from priority node
d'_i	Actual distance between the i -th anchor and a node	d_i	Estimated distance between the i -th anchor and a node
P_t	Transmit antenna power	P_r	Receiving antenna power
G_t	Transmit antenna gain	G_r	Receiving antenna gain
λ	Signal wavelength	f	Operating frequency
L_p	Path loss component	α	Path loss exponent
β	Wave number	Δ	Inter-element spacing
A	Vandermonde matrix of steering vector	Φ	Diagonal unitary matrix
A_0, B_0	Coefficient matrix of linear equations	b_0, c_0	Constant matrix of linear equations
A'	Gain factor	ζ	Scaling factor
R_{xx}	Array correlation matrix	$()^H$	Hermitian operator
σ_η^2	Noise variance	E_x	Signal subspace
Ψ, Γ	Non-singular transformation matrix	μ_k	Eigenvalue of k -th signal
μ_0	Unified parameter	n	Number of beacon message/anchors/sample size/bits in data packet
j	Imaginary unit	$diag$	Diagonal elements
arg	Argument of a complex number	$()^{-1}$	Inverse operator
τ_1, τ_2	Signal transmission/reception time	τ_r, τ_s	Arrival times of the radio/ultrasound signals
τ_p	Signal propagation time	τ_c	Signal processing time

τ_d	Link set up time	τ_b	Computation time in beamformer
τ'_b	Computation time in coprocessor	τ'	Localization time
τ'_{av}	Average localization time	v_r, v_s	Speed of the radio/ultrasound signals
(x_k, y_k)	Actual position of the k -th node	(x'_k, y'_k)	Estimated position of the k -th node
(x_{ai}, y_{ai})	Position of the i -th anchor	(x_{an}, y_{an})	New position of the anchor
X	Estimated node position	X_{total}	Total number of estimated positions
(x, y)	Coordinates of rotation vector	X_{mean}	Final estimated position
X_{pbest}, X_{gbest}	Personal/global best position	x	Unknown variable of linear equations
(x_0, y_0)	Initial coordinates	(x_n, y_n)	Final coordinates
x_{pk}, y_{pk}	Pseudo reference for k -th node	X_0	Particle position
$f(X_0)$	Fitness function	X_{min}, X_{max}	Search space
$X_{\sigma\eta}$	Random variable of zero-mean Gaussian distributions	σ_η	Standard deviation
σ	Direction of the micro-rotation		
ϵ_r	Permittivity	ϵ_b	Beamforming error
ϵ_d	Distance estimation error	ϵ_θ	Angle estimation error
ϵ	Localization error	ϵ_{av}	Average localization error
E_{elec}	Energy dissipation per bit in the electronic circuits	E_{amp}	Energy consumption in the power amplifier circuit
E_b	Power dissipated to broadcast a message	E_r	Power dissipated to receive data packets
E_{total}	Total energy consumption of a node	t'_k	Number of anchor points required to localize k -th node
ξ	Energy consumption for a node	ξ_{av}	Average energy consumption
T	Time period	SNR_{th}	Threshold level in Signal-to-Noise Ratio
t_{max}	Maximum permissible anchor points	N_s	Number of nodes in active mode
$2N$	Number of array elements	N	Number of dimensions
N_{fb}, N_{fm}	Number of benevolent/malicious nodes wrongly detected	N_{tb}, N_{tm}	Total number of benevolent/malicious nodes
N_b, N_m	Number of benevolent/malicious nodes	N_c	Number of clock cycles per search
SR	Success rate	TPR	True positive rate
AF_d	Beam pattern desired	AF_p	Beam pattern produced
M	Number of particles	$FNBW$	Beamwidth between the first nulls
SLL	Sidelobes level	K	Depth of nulls
η_0, γ_0	multiplication factor	c_1, c_2	constants
γ', ψ'	Latitude, Longitude	r	Radius of curvature of the earth
r_0	Initial rotating vector	r_n	Final rotating vector
a, b	Semi-major/Semi-minor axis of the ellipse	e	Eccentricity
CS	Cluster size	nCS	Updated cluster size
P_{tr}	Pointer value	w_k	Priority index
U, V	Data table in anchor	U', V'	Data table in node
B_m	Generated beacon message	ms	Mode state
P, Q	Data tables in anchor	P', Q'	Data tables in node
D_0, D'_0	Deviation in coprocessor/node	t_0, \dots, t_4	Allotted clock time
t_b, t_r	Time in broadcasting/receiving signal	t_d, t_a	Time in distributing/accumulating beacon message

t_g, t_m	Time in generating beacon message/anchor movement	t_{con}, t_c	Convergence time/total number of anchor points/clock period
V_0	Particle velocity	V_{min}, V_{max}	Velocity range
CoG	Center of gravity	$\rho(n)$	Degree of membership function
ρ_t	Power consumption	ρ_s, ρ_d	Static and dynamic power
FIS	Fuzzy inference system	$Index$	Index of vector elements
$rand()$	random number in the range of $\{0,1\}$	i_{max}, it_{max}	Maximum permissible iterations
δ	Variant/deviations	δ_{max}	Maximum limit for variant
nID	Updated clustering node ID	$s'ID, n'ID$	Node IDs in sleep/active mode
bID	Benevolent node ID	sID	Suspicious node ID
Λ_l	Localization accuracy	Λ_b	Beamforming accuracy
Λ_s	Number of successful attempts	Λ	Total number of attempts
R'	Rotation matrix	R	Register
W_{pbest}, W_{gbest}	Personal/global best weight vector	$f(W_{pbest})$	Personal best value of weight vector
$f(W_{gbest})$	Global best value of weight vector		
p	Number of estimated positions	q	Number of estimated variants
κ	Inertia weight	κ_0	Inertia weight step
f_0, s_0	Any number	Y_0, w_0	Modulus value/square root of number
$EIRP, P_0$	Effective isotropic radiated power	Ω_c	Consistency factor
Ω	Number of sample points	z	Accumulator
v	Hardware resource utilization	n_u, n_l	Number of resources used/the total resources available
F	Fitness value	χ	Beamforming efficiency
p'	Total number of micro-rotations	K_0	Total number of samples
s_j	Received signals	ψ	Progressive phase shift
t, i, j, n	Dummy variables	l, k, m, r'	Dummy variables
$it, d', count$	Dummy variables	sum, idx	Dummy variables
fCS, fID	Dummy variables	$Repeat$	Dummy variables
$D_{min}, Flag$	Dummy variables		

Chapter 1

Introduction

1.1 Wireless sensor networks

A large number of small, inexpensive, battery-powered, self-configuring sensor devices (nodes), most often deployed randomly over the region of interest (indoor/outdoor, or even in remote and hostile areas) on an ad-hoc basis, constitute Wireless Sensor Networks (WSNs) to accomplish specific tasks [1]. Such networks operate efficiently without the need for centralized supervision and have thus become very popular in a variety of applications [2]. Nodes, also known as motes, sense raw data from the environment, convert it to digital form, and then transmit it to a sink or base station in a collaborative manner. At the base station, the data is processed and analyzed to recognize any event occurring in the network. The base station is usually placed at a convenient location and has enormous power, long transmission range, and higher processing capacity. Finally, a task manager or user collects the processed data directly from the base station. The base station connects the user externally via a wireless radio link (satellite or internet) [3]. Fig. 1.1 illustrates a typical WSN architecture and its operations.

1.1.1 Advantages of WSNs

A dense network composed of adequate wireless sensing devices distributed in space provides several advantages as follows [4].

- *Signal-to-noise ratio* (improves SNR level by reducing average distances from sensor to sensor and from sensor to signal source or target).
- *Energy efficiency* (increases efficiency by reducing power consumption for communications through a multi-hop network topology).
- *Flexibility* (suitable for deployment via simple infrastructures and self-organizing capabilities).
- *Robustness, fault tolerance and scalability* (allows new nodes to replace faulty nodes in data routing through network redundancy).
- *Ad-hoc features* (easy and fast network configuration in a remote and hostile environment).
- *Data fusion* (aggregates additional data from other sensors during multi-hop transmission in the network).
- *Cost* (simple and cost-effective network architecture).

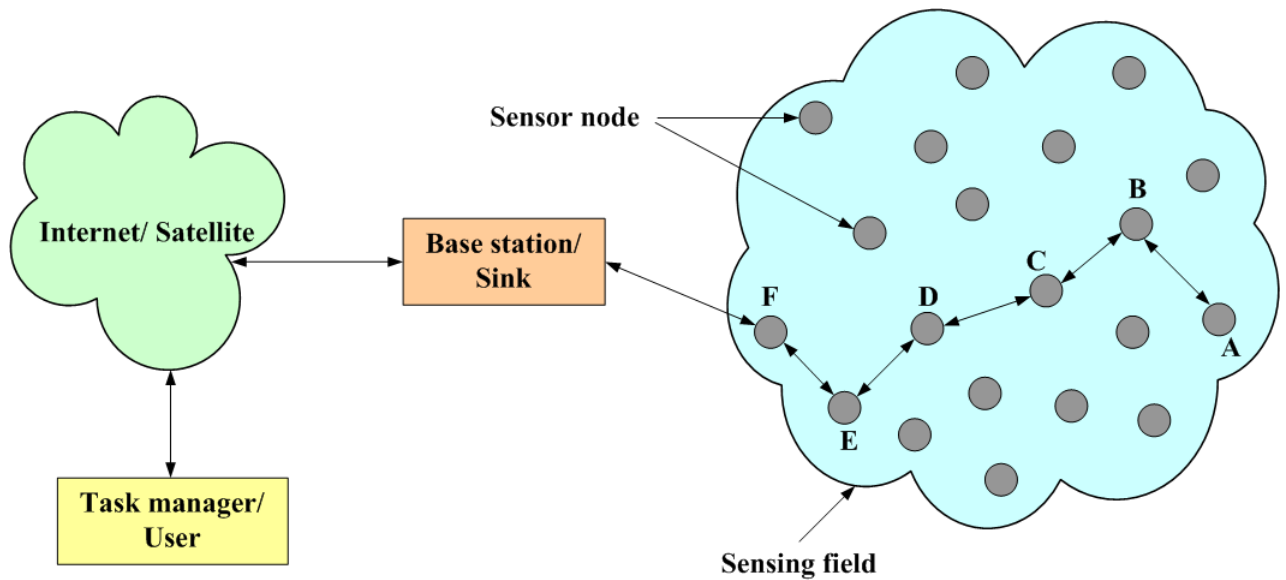


Fig. 1.1. WSN architecture for a specific task

1.1.2 Challenges of WSNs

The realization of a WSN still faces several challenges depending on the deployment conditions and the physical environment. Therefore, they must be overcome for its versatile applications. These are mostly inherent limitations of tiny sensors, as listed below [5].

- *Device heterogeneity* (very small sensing areas of the order of a few meters and low communication range of the order of a few tens of meters).
- *Manufacturing quality* (possesses limited memory, processor with less computational capability and limited energy from battery).
- *Limited power capacity* (very difficult to replace battery or to employ energy harvesting in remote/hostile places).
- *Network support* (provides limited support for networking because no universal routing protocols or central registry services exist).
- *Dynamic topology* (changes network topology frequently due to multipath-fading and node failures because each node acts as a router as well as an application host).
- *Application support* (provides limited support in developing application software because it involves dynamic collaboration among nodes as a distributed system, to handle multiple competing events in real-time).
- *Quality of service* (relays unreliable data packets through wireless connectivity from source node to sink node).
- *Ecological issues* (is prone to security attacks and physical damages when deployed at outdoor in hostile environment).

1.1.3 Types of WSNs

Based on deployment areas and application purposes, WSNs are classified into five categories [6], such as

- *Terrestrial* (consists of hundreds to thousands of nodes, deployed randomly or pre-planned over the area of interest and communicates efficiently with base stations).

- *Underground* (consists of several nodes hidden in the ground to monitor underground conditions and additional sink nodes are placed above ground to relay information to the base station. It is more expensive than terrestrial WSNs in terms of deployment, maintenance, and equipment cost).
- *Underwater* (consists of several nodes and autonomous vehicles deployed underwater and establishes communications through acoustic waves).
- *Multimedia* (consists of low-cost nodes with microphones and cameras to collect multimedia data, such as images, video, and audio. It offers new challenges such as higher energy and bandwidth requirements as well as efficient systems for data retrieval, compression and correlation).
- *Mobile* (consists of nodes that can move and sense data from the environment. It improves coverage, energy efficiency, and channel capacity).

1.1.4 Applications of WSNs

Currently, WSNs are used to perform monitoring and tracking tasks in various applications [5], as listed below.

- *Environmental* (forest fire detection, animal tracking, flood detection, weather forecasting, and seismic prediction).
- *Military* (enemy tracking, security surveillance and battlefield surveillance).
- *Health* (tracking and monitoring of patients).
- *Transport* (monitoring of traffic, dynamic routing management, and monitoring of parking lots).
- *Industry* (process monitoring).

1.1.5 Basics of MICAz motes

The MICAz is a popular mote module manufactured by Crossbow to enable low-power WSN operations on the ISM bands (2.4 to 2.48 GHz). It is supported by MoteWorks™ which is based on the open-source operating system, TinyOS. MoteWorks™ provides the WSN platform for developing a reliable, ad-hoc mesh network over live programming capabilities, multi-layered development tools, server middleware, and client user interface [7]. The MICAz mote can also function as a base station when connected to a standard PC interface or a gateway board. It has four basic components [8], such as

- *Sensing unit* (consists of multiple sensors and A/D converters).
- *Processor and storage unit* (consists of low-power microcontroller, Atmel ATmega128L and internal flash memory).
- *Transceiver unit* (consists of wireless transmitter, receiver and an omnidirectional antenna).
- *Power supply unit* (battery).

Fig. 1.2 illustrates a typical MICAz mote module. It has a 51-pin expansion connector for plugging in a variety of sensors and data acquisition boards to collect physical data (e.g., light, temperature, RH, pressure, acceleration, acoustic, and magnetic, etc.) from the environment. It simultaneously runs data sensing, data processing, and data communication applications on its single processor board. The RF transceiver is compatible with IEEE 802.15.4 protocols allowing a data transmission rate of 250 kbps over the radio.

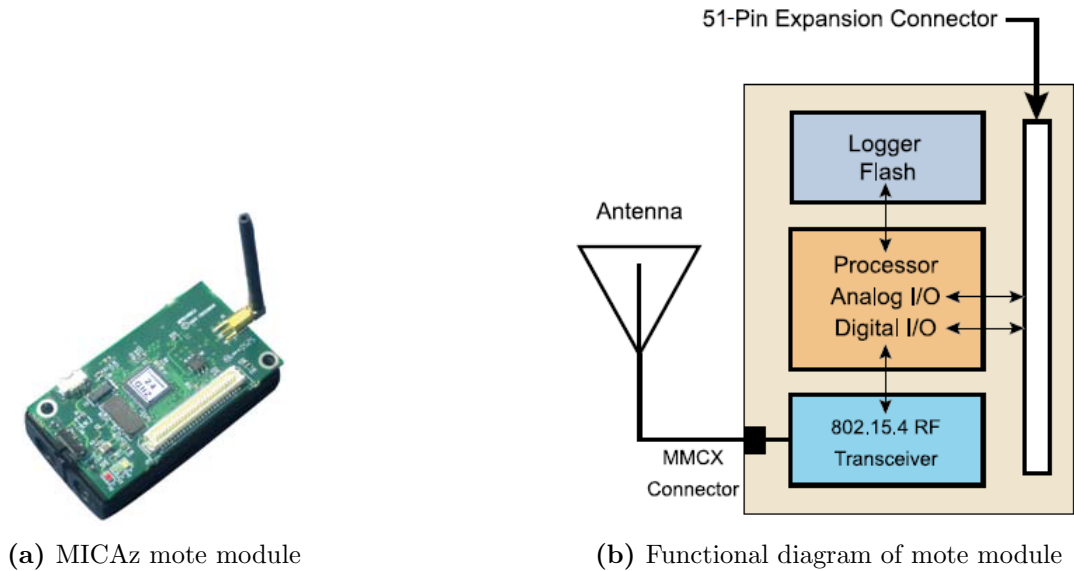


Fig. 1.2. MICAz mote and its operations

It ensures data security by resisting RF interference using DSSS (Direct Sequence Spread Spectrum) technique. It has a communication range (line of sight) of 75 to 100 meters for outdoor applications. It is powered by an external 2.7 to 3.3 V battery which draws a current of 8 mA (active), 19.7 mA (receive), 20 μ A (idle, voltage regular on), and 1 μ A (sleep, voltage regulator off) for operation in its different modes.

1.2 Overview of the problem

Recent advances in digital and wireless communication technology have enabled the development of low-cost, low-power, multifunctional micro-sensors with built-in sensing, processing, and short-range communication capabilities. It is possible to establish a reliable wireless network architecture with less infrastructure to collect information from an indoor/outdoor environment in real time. A network of wireless sensors is able to identify an event by analyzing the raw data accumulated at the sink. Thus, it has now been a good choice for deployment in a hostile environment in an emergency [9].

Due to several advantages, WSNs have found their widespread use in the civilian and defense sectors over the past two decades. Rapid technological growth always expedites the addition of new features and thus explores new areas of research. Secure localization based on smart antennas is one such new research area that is very much relevant to the current WSN scenario. Adversarial attacks are very common in WSNs and cause problems in localization systems [10]. Security issues become more critical for resource constrained nodes in the physical layer as compared to the application or other inner layers. Malicious attacks typically affect localization systems by compromising a few benevolent nodes (spoofing, replication, and forgery) or by compromising the network environment in particular regions (signal strength reduction, packet drop, packet replay and packet delay) [11]. This causes drastic drops in network throughput and deteriorates Quality of Service (QoS) in data transmission. Moreover, it disrupts the entire network operations to produce lots of dead nodes and causes coverage problem. Nodes operate in a cooperative manner to relay data packets in the network. They become vulnerable and thus data packets also remain fragile for attackers. However, the base station needs to have knowledge about data sources to recognize any event. Localization systems play an important role in estimating the location of corresponding data and are now becoming the target of malicious attacks.

Location estimation involves measurement of Received Signal Strength (RSS), Time of Arrival (ToA), Time Difference of Arrival (TDoA), and Angle of Arrival (AoA) on relayed data packets. It also uses the known coordinates of some nodes as a reference [12]. These nodes, called anchors/beacons, are aware of their location with Global Positioning System (GPS) receivers. Therefore, attacks can interfere with the localization process, causing incorrect measurement of data (distance/angle) and/or introducing a GPS error. An attack can happen by flooding fake IDs, jamming intermittently, retransmitting a message several times, and fabricating a fake node ID. Although a number of pioneering research works on secure localization in WSNs are available in the literature, most of them do not consider the security issues in the physical layer. This thesis addresses the security problems encountered in the physical layer, protecting localization systems against malicious attacks aimed at capturing nodes/anchors and altering the radio environment.

1.3 Motivation and objective of the thesis

Cryptographic approaches are susceptible to internal attacks in WSNs. Moreover, they are more complex, imposing additional overhead on resource-constrained nodes and hence often avoided [13]. Instead, non-cryptographic schemes are more reliable in a hostile environment and are now preferred in most cases. Existing non-cryptographic methods are robust enough to provide data packet security in the application or other inner layers. But most of them fail to perform well against attacks in the physical (PHY) layer. Nodes and anchors will be exposed to adversaries whenever they send data packets over the radio. In the literature, only a few schemes currently exist to mitigate PHY layer attacks [14, 15]. They mainly use transmit power control, noise and jamming technique. Despite this, they remain fragile to protect localization data in a corrupted radio environment. Therefore, it is desirable to implement more efficient systems to withstand more complex attacks. At the same time, achieving resilient localization systems that preserve the security of data packets in the physical layer is still a significant challenge for researchers. However, the introduction of adaptive array antennas can provide desirable outcomes. An adaptive array antenna exploits spatial diversity to counteract multipath fading under a changeable and unpredictable wireless channel [16]. Thus, it improves throughput and quality of services over stable point-to-point communication links. Its direction-finding feature estimates localization data (e.g., RSS, ToA, TDoA, AoA, etc.) with high precision. Also, its adaptive beamforming attribute ensures secure data communications between nodes and anchors by keeping malicious nodes out of radio coverage.

The main objective of this thesis is to integrate smart antennas to enhance the integrity and confidentiality of relayed data packets towards the implementation of secure localization systems. Since the integration of smart antennas always incurs huge overhead, it is very difficult to maintain the lifetime and efficiency of resource-constrained nodes in WSNs. Securing anchor reference data from attack by jamming the recovery of GPS signals to preserve the accuracy of localization systems also remains a challenging task. Therefore, this work suggests the integration of adaptive array antennas with only a few mobile anchors to meet the resource constraints of tiny nodes and to keep deployment costs within acceptable limits. Introducing mobility into the anchor always reduces the risk of attacks on its references. It is a new concept and very little research work is reported in this direction. The objective of this research study is to analyze the compliance and merits of secure localization systems based on smart antenna compared to existing systems in WSNs. In this context, we have proposed a few works as follows.

- Design of a robust localization system using smart antennas.
- Design of a secure localization system, detecting malicious nodes using an intelligent anchor mobility control strategy.

- Design of a secure localization system, protecting anchor points by antenna pattern control and generating pseudo-references.
- Design of a secure localization system based on a consistent variant assortment, which improves network performance by preventing malicious nodes from being dead in a hostile environment.
- Design and implementation of a smart antenna beamformer for a secure localization application.
- Design and implementation of an anchor coprocessor for localization application.
- Performance analysis and comparative study of secure localization systems based on smart antennas with existing systems.

The following sections include brief discussions on the concept of secure localization and smart antennas that are related to this research topic.

1.3.1 Importance of localization systems

The purpose of localization systems is to identify the actual location (e.g., latitude, longitude, and altitude) of any node in the network. Network operations such as geographic routing, geographic key distribution, and location-based authentication must have information about the locations of all allied nodes [17, 18]. From this perspective, possible approaches to get the location information directly could be either of the two

- equipping all nodes with GPS receivers, or
- manual installation of each node at known coordinates.

But, it is an unrealistic and cumbersome process as it puts several constraints on tiny nodes like energy consumption, memory capacity, implementation cost, etc. Hence, a network with self-localization capability is always desirable. The most viable solution is to incorporate suitable localization systems through which nodes can autonomously estimate their positions and relay this information as part of data packets.

A wireless sensor network infrastructure is mainly deployed to monitor an area of interest in a harsh environment. By using a localization system with the network, prior configuration of nodes is not necessary to know their locations. This often facilitates the random deployment of nodes from aircraft into inaccessible terrain or disaster-prone areas for relief operations. Thus, a localization system is necessary to provide position information to nodes. Node location information is very important for recognizing the event and making decisions accordingly. Several factors that signify the importance of the localization system in WSNs are listed below [5].

- *Identification of gathered data* (the occurrence of an event can be described with the mapping of the collected data. It is thus important to know the corresponding data locations to identify the region where the event occurred).
- *Correlation of gathered data* (knowledge of the locations of collected data can enable the process of data fusion at any intermediate node by correlating data on the same region as it is transmitted through the network).
- *Addressing of nodes* (localization allows nodes configured with unique IDs to use their physical location in the network).
- *Network management* (localization allows to manage and query all the nodes located in a particular region, to evaluate their coverage and to generate their energy maps).

- *Geographic algorithms* (node location information is also important for developing some algorithms focused on routing, density control, and object tracking to optimize the use of network resources).

1.3.2 Features of localization systems

Localization systems are considered a key technology for the development and operation of WSNs. Thus, a localization system must include attributes that are often required for WSN applications [5], as listed below .

- *Auto-organization* (must be capable of working independently of any infrastructure).
- *Scalability* (must be suitable for use in a large-scale and/or dense sensor networks).
- *Robustness* (must have a high tolerance for communication problems and computing positions with incorrect distance and angle information).
- *Efficiency* (must comply with resource constraints to improve network lifetime).

1.3.3 Basics of localization systems

In WSNs, nodes remain scattered over the field and communication between two nodes occurs over symmetric links (bidirectional and having equal signal strength) within a specific range. Thus, the Euclidean graph in 2-D (Two-Dimensional) space can represent them, where each node has a unique coordinate specifying its position. Several categories of nodes (according to their current states) can exist in the network [5], as listed below.

- *Unknown, free, or dumb nodes* (refers to nodes that do not know their location information after deployment).
- *Settled, or localized nodes* (refers to nodes that were initially unknown categories, but then successfully estimated their positions using a localization system).
- *Beacon, landmark, or anchor nodes* (refers to nodes that do not need a localization system to estimate their physical positions. They can obtain their position information from manual placement or from external devices such as GPS receivers).
- *Reference nodes* (refers to nodes whose location information is used by an unknown node to estimate its location).

The main objective of localization systems is to allow unknown, free, or dumb nodes to estimate their positions in the network. Thus, localization systems in a multi-hop network use a set of beacon nodes with their known positions to find the position of all unknown nodes, transforming unknown nodes into settled nodes. The beacon nodes form the basis of most localization systems for WSN. The number of nodes settled and the error obtained in their position estimation are the two main parameters to define the quality of a localization system. A beacon or settled node can serve as a reference node in the localization system.

1.3.4 Components of localization systems

Localization systems have three distinct components on which they perform the entire localization process in a sequence [19], such as

- *Distance/angle estimation* (estimate distance/angle information between two nodes by measuring the value of RSS, ToA, TDoA, and AoA of signals received from them),

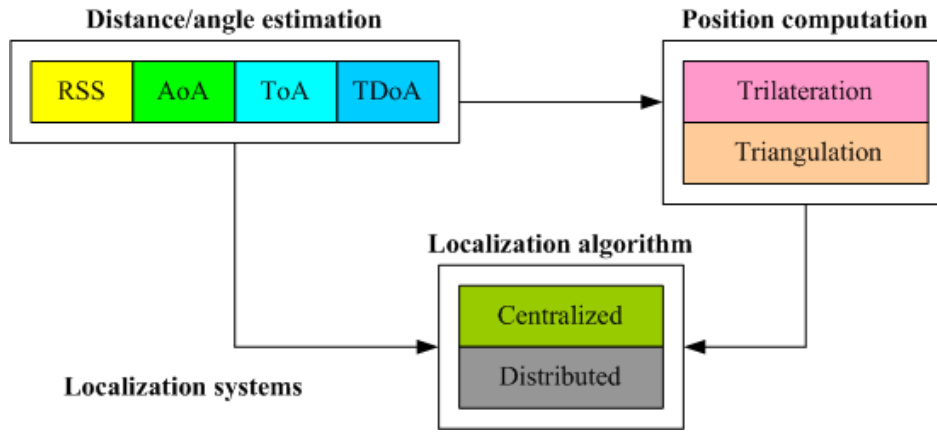


Fig. 1.3. Localization system architecture

- *Position computation* (compute positions of nodes on available distance/angle information and anchor references using trilateration, multilateration, triangulation, bounding box, or probabilistic approaches), and
- *Localization algorithm* (promulgate the information of distances/angles and positions through the network to estimate the positions of all nodes).

Therefore, the performance of such localization systems depends mainly on the proper functioning of these components. Fig. 1.3 shows the various parts of a localization system.

1.3.5 Types of localization systems

There are several categories of localization systems in the literature, mainly based on the implementation strategies of localization algorithms [20], such as

- *centralized, or distributed* (according to position computation strategy like self-positioning/remote positioning).
- *range-based, or range-free* (depending on range measurements).
- *anchor-based, or anchor-free* (depending on the need for additional infrastructure).
- *single-hop, or multi-hop* (depending on node connectivity and topology).
- *indoor, or outdoor* (according to node deployment scenarios).

In a centralized system, data is accumulated in a single central node (sink) for processing. Although it often provides higher accuracy but it needs a central coordination with all nodes for its operation. On the other hand, nodes can localize themselves autonomously obtaining data from their neighbors in a distributed system. However, it increases the error cumulatively for its multi-hop attributes. A range-based system uses the range information (distance/angle) of nodes to estimate their positions. It depends on measurements of several signal parameters such as Received Signal Strength Indicator (RSSI), Time of Flight (ToF), Time Difference of Arrival (TDoA), and Direction of Arrival (DoA), etc. It can produce a more precise and accurate position estimation, but it often requires additional hardware, which increases system costs. Instead, a range-free system uses the hop counts between a pair of nodes as a distance metric without requiring any hardware for distance/angle estimation. Thus, it is more cost-effective for its simple structure and can only provide abrasive position estimation. Anchor-based systems employ a few location-aware nodes (static or mobile) either with manual placement or GPS devices as a reference, called anchors, to alleviate position computations in unknown nodes and improve localization accuracy.

Thus, all nodes are localized in a geodetic coordinate system (latitude, longitude). Anchor-free systems, in contrast, are employed in a network where no nodes have pre-configured positions and use local distance information from a node/point to determine coordinates of unknown nodes. However, the basic limitation lies on the fact that such a coordinate system cannot be unique and is often embedded with a global coordinate through translation, rotation, and flipping. A single-hop system employs a sufficient number of beacon nodes to establish one hop (direct) communication with unknown nodes. It is lighter and simpler, but has severe scalability problems. On the other hand, a multi-hop system is more scalable due to its distributed nature and is popular for a wide area network. For greater accuracy and compatibility with wider coverage, most state-of-the-art localization systems prefer to distributed, multi-hop, range-based categories, often including a few beacons for an outdoor deployment environment.

1.3.6 Performance of localization systems

Performance of the localization systems refer to the degree of accuracy they provide in computing position of unknown nodes. Several metrics are used to evaluate it [5], as listed below.

- *Mean error and consistency* (indicates an acceptable amount of the mean error of the estimated positions and its repeatability under varying network conditions that always defines the use of localization systems to specific applications).
- *Communication cost* (signifies the system complexity for exchanging localization data among nodes and also overhead imposed to the resource-constrained nodes).
- *Number of settled nodes* (defines the speed and the success rate of localization systems within the stipulated time).
- *Number of beacon nodes* (indicates the efficiency of localization systems to use minimum beacons and making it more cost-effective).

However, the performance of localization systems also depends on some network features [11], as listed below.

- *Network density* (a highly dense network always ensures more neighbors for an unknown node, yielding lower distance between nodes and thus the mean error becomes lower).
- *Network size* (more nodes is deployed to keep network density unchanged in an increased sensor field area, which often yields more hops in position computations and increases the mean error).
- *Number of beacon nodes* (a deployment of more beacon nodes always increases the settled nodes in the network, which in turn, reduces the mean error).
- *GPS accuracy* (beacon nodes generally use the GPS receivers to provide reference positions in the localization and thus, the GPS error will cause an erroneous position estimation).

1.3.7 Security issues in localization systems

In WSNs, all nodes and anchors share a common wireless medium to relay data packets. They become vulnerable to malicious attacks in a hostile environment, and the data packets also become fragile. Usually, the attacks can happen in two ways [11], such as

- inject deceptive information via some malicious nodes located outside the network (known as direct/external attacks), or
- compromising some benevolent nodes with malicious code by implanting external nodes (adversaries) in the network (known as indirect/internal attacks).

However, internal attacks such as Sybil, Replay, Wormhole, etc., become emerging issues as they bypass any authentication/authorization process. They control many fraudulent nodes by re-programming the captured nodes and thus make the localization process very problematic. By exploiting captured nodes, adversaries can make estimated positions incorrect in benevolent nodes generally in two ways [17], such as

- interfere with the localization process by deceiving where a malicious node invalidates a benevolent node by silently repeating its relayed data packets or stealing all of its authentication credentials (e.g., cryptographic keys, passwords, etc.) and constantly updating them.
- tamper with localization information (e.g., RSS, ToA, TDoA, AoA, etc.) where a malicious node sends outdated information, causing congestion in the data transmission. Otherwise, it modifies the radio environment (placing obstacles/magnets) on a particular network region to alter information exchanged (e.g., reference coordinates, time stamps, transmission power, hop counts, etc.).

It would surely make the distance or angle estimation erroneous using this outdated, modified, or disturbed information and produce inaccurate position estimations for benevolent nodes. Eventually, the proliferation of wrong information from these compromised or affected nodes can degrade overall network performance.

1.3.8 Importance of secure localization systems

Localization systems often play a crucial role in decision-making regarding any event occurring on the network. As a result, they are now tempting targets for adversaries. An adversarial attack can easily mislead any plan or decision made at the base station by disrupting the functionality of any part of the localization systems [11]. Therefore, they need to be made as robust and secure as possible to improve their performance in a hostile environment. For this, localization systems always need to be developed in such a way that would perform well in presence of

- *compromised nodes* (malicious nodes or a few benevolent nodes that have been corrupted by a malicious code), and
- *compromised environment* (adversaries can change the propagation characteristics over network environment and may also access the nodes physically).

It is important to utilize the network resources (e.g., energy, media access, processor, and memory, etc.) properly to implement and maintain the security of localization systems. Considering the resource constraints in WSNs, it is also essential to define the level of the security and number of resources to be used for a particular application. Localization systems often become more vulnerable due to requirement of resource saving solutions. Therefore, the security solution must always comply with the trade-off issues in a WSN. Moreover, each component of the localization systems needs to be made secure enough. Otherwise, a small error arising in any component will greatly affect the functioning of entire localization system, deteriorating the overall performance of WSN.

1.3.9 Strategies for secure localization systems

A localization system always works keeping strong coordination among its components. Therefore, all of its components need to be protected from malicious attacks to implement a secure localization system. Several approaches can be followed to achieve the security solutions for localization problems [11], as listed below.

- *Cryptographic security solutions* (use of authentication and/or message integrity checks can easily defend attacks by implanting malicious nodes or changing the values in data packets).

- *Detection and blocking of misbehaviors* (observation on the behavior of nodes over time can protect the position computation simply by ignoring information gathered from untrusted nodes).
- *Robust position computations* (use of statistical decisions and outlier filtering can ensure position computations in the presence of bogus information, if benevolent nodes become more than the malicious nodes).
- *Verification of estimated locations* (use of redundant information available on nodes and in the network can check the reliability of computed positions).
- *Secure and simple algorithms* (use of simple, GPS-free, range-free, and/or single-hop algorithms can ensure the reduction of dependable components as well as their vulnerabilities).

1.3.10 Challenges for secure localization systems

Cryptography is very useful to preserve position confidentiality preventing external malicious nodes from gathering network information. But, it fails in the presence of compromised nodes/compromised environment as the attackers can access to locally stored keys and passwords. Also, it is avoided due to the limited resources (e.g., processor and memory) in sensor nodes. For this reason, most secure localization systems use non-cryptographic security solutions and cryptography is used to provide a second layer of protection [11]. However, detecting and blocking information from compromised nodes can become effective only for position computation component. Likewise, making statistical decisions is worthy to protect the position computation and distance/angle estimation components. Instead, filtering the estimated positions prior to use in the final computations can be a viable security solution for all three components, verifying only the result of the overall localization system. On the other hand, a simple and secure algorithm can be used to protect only the localization algorithm component and it often requires an increased computational and communication complexity. In this context, localization system with a mobile anchor can provide the best result. By using a single-hop communication with neighboring nodes to exchange messages, it can be worthwhile against a number of distributed attacks.

1.3.11 Concept of smart antennas in localization systems

In a wireless sensor network, most of the radiated power is wasted as the nodes establish communication links with omnidirectional patterns from their dipole antennas. Such unused power creates interference with neighboring nodes and deteriorates the overall Quality of Service (QoS) in the network. Integrating smart antennas with wireless sensor network infrastructures improves the performance of propagation characteristics of signals transmitted from nodes under variable radio conditions. A smart antenna consists of an antenna array in a particular geometry (e.g., linear, circular, or planar) followed by a digital signal processor, as shown in Fig. 1.4. It exploits spatial diversity to mitigate multipath fading and improves the Signal-to-Interference-plus-Noise Ratio (SINR) in a wireless channel [21]. This always makes the links stable and secure to guarantee a higher throughput and an improved QoS in data transmission over an erratic channel [16]. A smart antenna operates in two successive phases [22], such as

- estimate the directions of intercepted signals, called the *Angle of Arrival* (AoA) estimation, and
- generate an optimal pattern based on the obtained AoA data, called *adaptive beamforming*.

Adaptive beamforming remarkably improves link capacity by providing higher bandwidth per channel. Angle of arrival estimation, on the other hand, ensures several location-based services in WSNs.

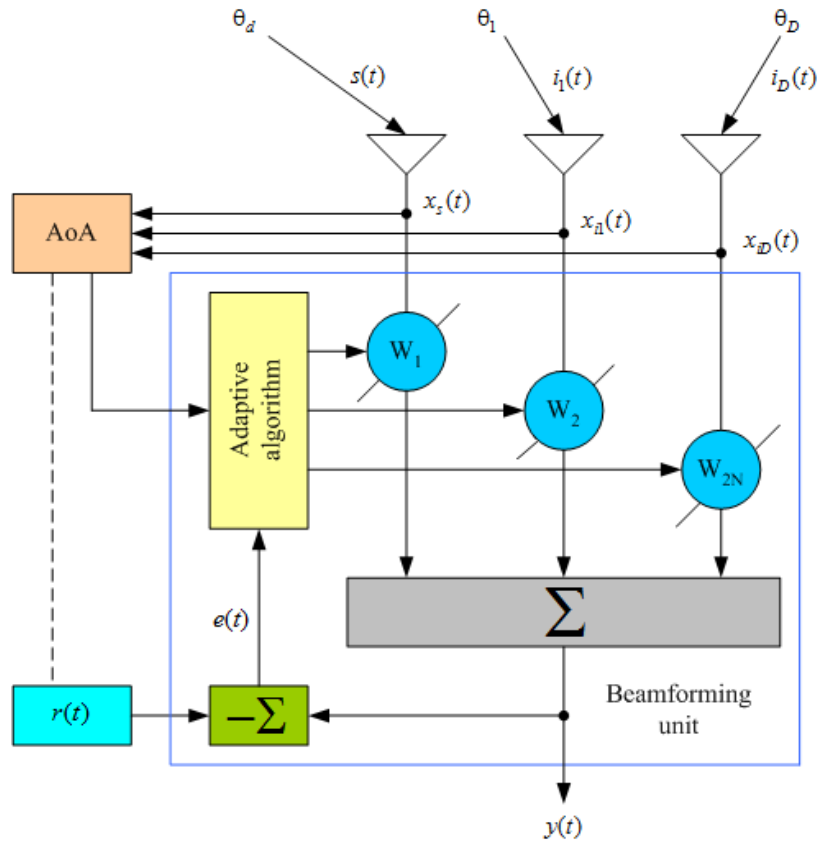


Fig. 1.4. Block diagram of an adaptive array antennas

Such attributes often become not only effective at relaying data packets with higher throughput, but can also preserve their integrity and confidentiality by sending them over more stable and private links [23]. Therefore, smart antennas can now be the best choice for precise localization of nodes in WSNs. To send data packets securely to target nodes, it is possible to establish radio links by producing antenna patterns with a narrower beamwidth. Such patterns are also able to cancel interference from other nearby nodes by steering deep nulls in their directions and keeping the Sidelobe Level (SLL) lower. Hence, smart antennas can ensure the lowest probability of errors in the distance/angle estimation process, which leads to the precise localization of nodes in WSNs. A smart antenna processes intercepted signals iteratively to produce an optimal pattern with higher directive gain in a chaotic environment. Thus, it significantly increases the coverage area across the entire network.

Its principle of operation is more likely to be a Multiple-Input Multiple-Output (MIMO) system that uses the spatial diversity effect of antenna arrays, normally referred to as a Spatial Division Multiple Access (SDMA). Therefore, its channel capacity or data transmission rate can be further improved by installing a large number of array elements and transmitting the space-time block coded waveform. Smart antennas are classified into two categories [24], such as

- Switch-beam antennas, and
- Adaptive array antennas.

An adaptive array antenna is more popular due to its ability to handle signal in fluctuating radio environment. It uses the AoA information to produce an optimal pattern under varying signal conditions by computing the array weight vector. Such a pattern steers the main beam towards the target node and also steers nulls in the interfering directions. This can ensure more stable links with such patterns that can effectively reduce the effect of multipath fading and co-channel interference.

Estimation of AoA is mainly based on time series analysis, spectrum analysis, eigenstructure methods, parametric methods, and linear prediction methods [22]. For example, Bartlett, Capon, Multiple Signal Classification (MUSIC), Root-MUSIC, and Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT) are very common. Beamforming algorithms, on the other hand, typically work on maximizing Signal-to-Interference Ratio (SIR), minimizing variance, and minimizing Mean Square Error (MSE). Some common examples are Least Mean Squares (LMS), Recursive Least Squares (RLS), Constant Modulus Algorithm (CMA), Sample Matrix Inversion (SMI), and Conjugate Gradient (CG) methods.

Beamforming in adaptive array antennas is an iterative process of updating the array weight vector (W^T) to produce an output signal $y(t)$ in accordance with the reference signal $r(t)$. The process continues until the error signal $e(t) = r(t) - y(t)$ reaches an allowable limit and can be expressed as follows.

$$y(t) = W^T x(t) \quad (1.1)$$

where, the intercepted signals, $x(t)$ is a composite signal of the desired signal, $x_s(t)$ arriving at the angle θ_d , D number of interfering signals, $x_i(t)$ arriving at the angles from θ_1 to θ_D and a zero mean Additive White Gaussian Noise (AWGN), $\eta(t)$. This can also be expressed as follows.

$$x(t) = \bar{a}(\theta_d)s(t) + [\bar{a}(\theta_1) \quad \bar{a}(\theta_2) \quad \dots \quad \bar{a}(\theta_D)] \begin{bmatrix} i_1(t) \\ i_2(t) \\ \cdot \\ \cdot \\ i_D(t) \end{bmatrix} + \eta(t) = x_s(t) + x_i(t) + \eta(t) \quad (1.2)$$

Here, $s(t)$ and $i(t)$ are the wavefronts of the desired and interfering signal, respectively. Also, $\bar{a}(\theta)$ is the steering vector of the array.

However, integrating smart antennas with resource constrained nodes would impose huge overheads. Therefore, we considered its integration with a few mobile anchors in the WSNs.

1.3.12 Secure localization systems based on smart antennas

Secure localization based on smart antennas is the new research area, where adaptive beamforming concept and direction finding attribute are used to solve localization problems in WSNs. An adaptive beam pattern often keeps malicious nodes outside radio links in an erratic channel. Moreover, precise estimation of directions always becomes useful for securely relaying data packets to target nodes over a private link. Therefore, malicious nodes cannot interfere with the localization process by altering data packets or impersonating. Blocking malicious nodes guarantees a significant improvement in localization accuracy in a hostile environment. This work keeps the localization process secure against malicious attacks by using the following strategic ways, such as

- separate compromised nodes from the network during a node capture attack,
- filter out erroneous distance or angle estimates in a corrupted radio environment, and
- keep the anchor reference intact during malicious attacks.

To achieve this, we proposed hybrid localization systems that compute the positions of the nodes based on RSS and AoA information. For all cases, almost the same network architecture and the same data communication protocols are considered. These are mentioned below.

- Nodes are assumed to broadcast signals periodically and consistently with their respective MAC (Media Access Control) IDs.

- The anchors are configured with a prescribed threshold for the signal-to-noise ratio (SNR_{th}) to ensure reliable data collection from the received signal.
- At each anchor point, a unique cluster is formed only with neighboring nodes whose RSS exceeds this threshold and the distance and angle information of the received signals are estimated.
- Anchors are supposed to generate and transmit beacon messages (containing estimated distance, angle, and current anchor references) to the corresponding nodes by establishing secure and stable point-to-point communication links.

For each cluster, the distance (d) between an anchor and the neighboring nodes is estimated using the path loss model. Thus, considering radio wave propagation through free space, Friis' transmission equation is written as follows [16].

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2} = \frac{P_t G_t G_r}{L_p} \quad (1.3)$$

It can also be expressed (in dB) as $P_r = P_t + G_t + G_r - L_p$

where, P_t , P_r and G_t , G_r is the power and gain of the transmitting and receiving antenna, respectively. λ is the wavelength of the transmitted signal and L_p denotes the path loss component.

Assuming log-normal shadowing conditions, path loss component is modified and the distance information of signals arriving from k -th neighboring node is estimated as follows.

$$L_p = \left(\frac{4\pi d_0}{\lambda} \right)^2 \left(\frac{d_k}{d_0} \right)^\alpha X_{\sigma\eta} \quad (1.4)$$

It can also be expressed (in dB) as $L_p = 32.45 + 20 \log_{10} d_0 + 20 \log_{10} f + 10\alpha \log_{10} \left(\frac{d_k}{d_0} \right) + X_{\sigma\eta}$

where, α is called the path loss exponent and d_0 is a small reference distance from the nodes (in KM). f is the frequency of operation (in MHz) and $X_{\sigma\eta}$ is a random variable of zero-mean Gaussian distributions with standard deviation σ_η .

Likewise, the angle (θ) between an anchor and the neighboring nodes is estimated using the TLS-ESPRIT (Total Least Squares-Estimation of Signal Parameters via Rotational Invariance Technique) algorithm [25]. The ESPRIT exploits the rotational invariance in the signal subspace, assuming a translational invariance structure of the array which consists of two identical sub-arrays with a finite separation (Δ), called doublets. Considering number of signal sources (D') is less than that of array elements ($2N$), the signals induced on each of the sub-arrays is written as follows.

$$x_1(t) = A_1 s(t) + \eta_1(t) \quad (1.5)$$

and

$$x_2(t) = A_2 s(t) + \eta_2(t) = A_1 \Phi s(t) + \eta_2(t) \quad (1.6)$$

where, $\Phi = \text{diag} \left\{ e^{j\beta\Delta \sin \theta_1}, e^{j\beta\Delta \sin \theta_2}, \dots, e^{j\beta\Delta \sin \theta_{D'}} \right\}$ is a diagonal unitary matrix and β is the wave number. Also, A_1 and A_2 is called Vandermonde matrix of steering vectors for two sub-arrays.

Again, considering the contributions of both sub-arrays, total received signal is expressed as follows.

$$x(t) = \begin{bmatrix} A_1 \\ A_1 \Phi \end{bmatrix} s(t) + \begin{bmatrix} \eta_1(t) \\ \eta_2(t) \end{bmatrix} \quad (1.7)$$

Thus, correlation matrix for the complete array is expressed as follows.

$$R_{xx} = E [xx^H] = AR_{ss}A^H + \sigma_\eta^2 I \quad (1.8)$$

whereas correlation matrices for two sub-arrays are represented as follows.

$$R_{11} = E [x_1x_1^H] = A_1R_{ss}A_1^H + \sigma_{\eta_1}^2 I \quad (1.9)$$

and

$$R_{22} = E [x_2x_2^H] = A_1\Phi R_{ss}\Phi^H A_1^H + \sigma_{\eta_2}^2 I \quad (1.10)$$

Due to the invariance array structure, signal subspace (E_x) can be decomposed into two subspaces: E_1 and E_2 whose columns include the D' eigenvectors corresponding to the largest eigenvalues of R_{11} and R_{22} . Since these arrays are related with translational invariance features, E_1 and E_2 would be related by a unique non-singular transformation matrix Ψ such that $E_1\Psi = E_2$. Similarly, there must also be a unique non-singular transformation matrix Γ such that $E_1 = A_1\Gamma$ and $E_2 = A_1\Phi\Gamma$.

From the above relationships, it is derived that $\Gamma\Psi\Gamma^{-1} = \Phi$.

Therefore, the eigenvalues of Ψ must be equal to the diagonal elements of Φ such that $\mu_1 = e^{j\beta\Delta \sin\theta_1}$, $\mu_2 = e^{j\beta\Delta \sin\theta_2}$, \dots , $\mu_{D'} = e^{j\beta\Delta \sin\theta_{D'}}$ and the columns of Γ must be the eigenvectors of Ψ .

Now, the angle information of signals arriving from k -th neighboring node is estimated as follows.

$$\theta_k = \sin^{-1} \left(\frac{\arg(\mu_k)}{\beta\Delta} \right) \quad (1.11)$$

where, $k = 1, 2, \dots, D'$.

Keeping compatibility with MICAz motes, we chose $P_t = 30$ dBm, $G_t = 0$ dBi, $G_r = 10$ dBi, $d_0 = 1$ m, $\beta = \frac{2\pi}{\lambda}$, $\Delta = \frac{\lambda}{2}$, $X_{\sigma\eta} = 0$ dB and $\alpha = 2$ for this work.

1.4 Justification of this work

Locating resource constrained nodes in WSNs against physical layer attacks is an important research area at present. Since all nodes relay data packets over a common wireless medium, there is always a risk of malicious attacks to capture them and possibly degraded network performance. In this context, the integration of smart antennas with localization systems could provide desirable solutions to the discussed problem. From this point of view, study and research are important for the development of secure localization systems based on smart antennas. It is still an open area of research and therefore requires rigorous study on this subject. The present thesis aims to design and implement new secure localization systems based on smart antennas for WSNs and to compare their performance with other existing systems in terms of localization accuracy and energy consumption.

1.5 Contribution of the thesis

The contributions of the thesis are as follows.

- A brief survey of security challenges for existing localization systems in WSNs.
- A review of adversarial attacks in the physical layer and the applicability of the smart antenna concept in WSNs.

- Literature review of existing secure range-based localization systems and their effectiveness against physical layer attacks.
- Proposal of two new mobility control schemes (centroid and fuzzy) for anchors in order to develop secure range-based localization systems.
- Proposal of a resilient range-based localization system using a smart antenna to estimate RSS and AoA from nodes in WSNs.
- Proposal of a new secure range-based localization system using a deterministic mobility control scheme (centroid/fuzzy) against node capture attacks in the physical layer.
- Proposal of a new secure range-based localization system using a consistent variation assortment method against attacks aimed at corrupting the radio environment in specific areas of WSNs.
- Proposal of a new secure range-based localization system using the strategy of antenna pattern control and pseudo-reference generation to protect anchor data against PHY layer attacks in WSNs.
- Proposal of a beamformer architecture for smart antennas and its implementation on an FPGA (Field Programmable Gate Array) platform for secure localization applications in WSNs.
- Proposal of a coprocessor architecture for anchors and its implementation on an FPGA platform for the localization of nodes in WSNs.
- A comparative study of all proposed systems with existing ones showing their merits and their feasibility in terms of performance.

1.6 Organization of the thesis

This chapter briefly describes the typical architecture of WSNs and the challenges associated with deploying them in a harsh environment. It also clearly explains security issues and the impact of attacks causing problems to the process of locating nodes in WSNs. It briefly introduces the concept of smart antennas and explains the needs of secure localization systems based on smart antennas to solve these problems. This chapter also includes the motivation and justification of the present research and the contributions of the thesis. The rest of the thesis is organized as follows.

Chapter 2 presents a brief overview of common security attacks in the literature, different methods for locating nodes in WSNs, and an overview of range-based localization systems. It also includes an extensive review of existing secure range-based localization systems.

Chapter 3 presents two new mobility control schemes (centroid and fuzzy) for anchors. These schemes work on the principles that should keep all nodes encountered first in a cluster to its next cluster. The centroid method evaluates the next anchor point using a weighted aggregation of the estimated distance and angle from all clustering nodes. On the other hand, the fuzzy method uses fuzzy logic to evaluate the next anchor point. These schemes are useful for locating clustering nodes in two successive anchor points and thus require the fewest anchor points for localization. This chapter also presents a robust localization system called Row Matching Algorithm (RMA) and its performance against three different anchor mobility control schemes (random, centroid, and fuzzy). It shows that both centroid and fuzzy schemes have advantages to use in realizing a secure localization system.

Chapter 4 presents a new secure localization system using a centroid/fuzzy based anchor mobility control scheme to counter node capture attacks in WSNs. By detecting any discrepancy in node IDs between two consecutive clusters, this scheme can easily identify misbehaving nodes in the network. It iteratively blocks malicious nodes from participating in the localization process. The simulation results for the success rate in detecting malicious nodes justify the merits of using such a system over existing systems.

Chapter 5 also includes a new secure localization system that is very effective at locating nodes in a corrupt radio environment. By adjusting the threshold value of the signal-to-noise ratio and the permissible variation iteratively, it estimates the mean of all positions which produces the consistent variations for any node. Simulation results for localization accuracy validate its robustness against existing systems.

Chapter 6 presents a security framework for anchor data against PHY layer attacks in WSNs using the strategy of antenna pattern control and pseudo-reference generation. This can be effective in securing anchor data for the localization process. In a cluster, data transmission occurs over point-to-point private links, generating an antenna pattern with a prescribed side-lobe level and steering deep nulls to the remaining nodes. Also, relayed beacon messages contain a pseudo-reference for anchors. Thus, it improves security by keeping eavesdroppers out of radio coverage and misleading them whenever they access beacon messages. Simulation results achieving a higher success rate corroborates its effectiveness over existing systems.

Chapter 7 presents a beamforming architecture for smart antennas on an FPGA board. By integrating such an architecture with anchors, it is possible to send data packets securely to nodes, canceling interference from other nearby nodes in an erratic channel. It is implemented using the Finite State Machine with Datapath (FSMD) design technology and appropriate COordinate Rotation DIgital Computer (CORDIC) blocks. The results of the simulations on the Xilinx Virtex-5 FPGA board validate its realization with a minimum of hardware resources.

Chapter 8 presents a coprocessor architecture for anchors on an FPGA board. Using such an architecture, an anchor computes the location of nodes centrally based on their RSS and AoA information. It improves network lifetime by avoiding computational overhead and reducing power consumption at resource-constrained nodes. The simulation results on the Xilinx Virtex-5 FPGA board justify its realization with acceptable hardware resources and less computation time.

Chapter 9 concludes the thesis by summarizing the overall findings and also suggesting the future direction of research in this area.

Chapter 2

Literature survey

2.1 Introduction

As mentioned earlier, localization systems play an important role in decision-making regarding any event in WSNs and they are now an easy target for attacks. A localization system operates on a strong relationship between its three components and thus the effect of attacks disrupting one of them propagates through the entire system, eventually degrading network operations. For example, malicious attacks causing an erroneous distance/angle estimate in any node, lead to an incorrect position computation. A localization algorithm can propagate this error through multi-hop network connectivity, which also causes a localization error in all allied nodes. Thus, localization systems are usually very fragile and difficult to secure [11]. In this chapter, we have studied existing localization systems and discussed their vulnerabilities to malicious attacks. We have also included an intensive study of existing secure localization systems and explained their limitations against physical layer attacks.

The rest of this chapter is organized as follows. An overview of commonly used distance/angle estimation techniques is presented in Section 2.2. Section 2.3 also discussed various position computation methods. Existing localization systems are described in Section 2.4. Section 2.5 explains the fragility of existing localization systems to malicious attacks. Secure localization systems available in the literature are discussed in Section 2.6 and Section 2.7 summarizes the chapter finally.

2.2 Distance/angle estimation techniques

These methods aim to obtain/identify distance or angle information between two nodes. These are used in the first component of any localization system and the estimated data is used in the next two components. Several methods are common to exist in the literature [5]. However, the level of accuracy and some other cost factors such as limitation of CPU (Central Processing Unit), energy, and memory resources often restrict their applications in many cases. Some popular methods for distance/angle estimation in localization systems are as follows.

2.2.1 Received signal strength

The distance between two nodes can be derived by measuring the parameter called Received Signal Strength (RSS) at one of the nodes. It is obvious that the signal transmitted by a node with a prescribed strength would be received with reduced strength at another node due to multipath fading during propagation. Received signal strength (measured in dBm or watts) at a node (Rx), is a function of distance to source nodes (Tx), and always varies inversely to the square of distance (d), for a free space LoS (Line of Sight) radio propagation model as shown in Fig. 2.1.

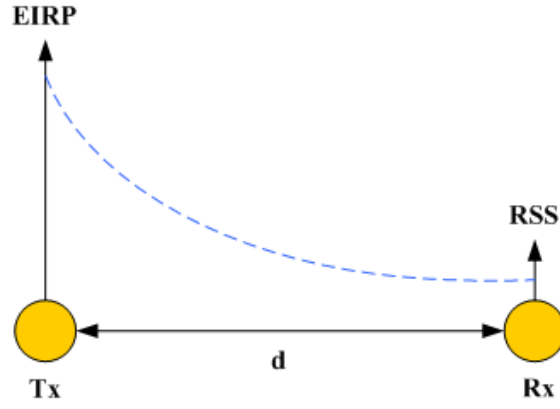


Fig. 2.1. Signal strength variation between source and sink nodes

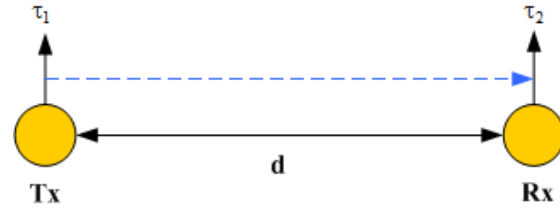


Fig. 2.2. Signal propagation time between source and sink nodes

However, considering the effect of noise and interference in real-time node deployment environments, a path loss exponent is also introduced to estimate the distance using the received signal strength. The path loss (L_p) is expressed in equation (1.3) and equation (1.4) which is re-written as follows.

$$L_p = \left(\frac{EIRP}{RSS} \right) G_r = \left(\frac{4\pi d_0}{\lambda} \right)^2 \left(\frac{d}{d_0} \right)^\alpha X_{\sigma\eta} \quad (2.1)$$

Here, $EIRP = P_t G_t$ is the effective isotropic radiated power of nodes. Also, P_t and G_t denote the transmit power and the gain of the dipole antenna for source nodes, respectively. At the sink node, RSS implies the intercepted power, and G_r is the gain of the dipole antenna. The nodes use an omnidirectional antenna for broadcasting and receiving signals, hence $G_t = G_r = 1$. λ is the wavelength of radio waves, d is the distance between the nodes and d_0 denotes a small reference distance from the nodes. In addition, α is called the path loss exponent, and $X_{\sigma\eta}$ is a random variable of zero-mean Gaussian distributions with standard deviation σ_η . This is a simple method with low overhead for sink nodes. However, its level of accuracy still depends on proper calibration of the systems in a controlled environment with an appropriate path loss exponent.

2.2.2 Time of arrival

The distance between two nodes can be estimated by measuring the signal propagation time, also called Time of Flight (ToF) or Time of Arrival (ToA). It is a simple method to consider that the distance between two nodes is directly proportional to the propagation time of the signal between them as shown in Fig. 2.2. Thus, the distance is estimated by using the equation as follows.

$$d = (\tau_2 - \tau_1) v_r \quad (2.2)$$

Here, v_r is the speed of the radio signal in free space (speed of light). Also, τ_1 and τ_2 are the instants at which signal transmission and reception occur in the nodes. However, such estimation still requires precise synchronization between the source and sink nodes. Additionally, the source node must include the transmission time with its relayed data packet.

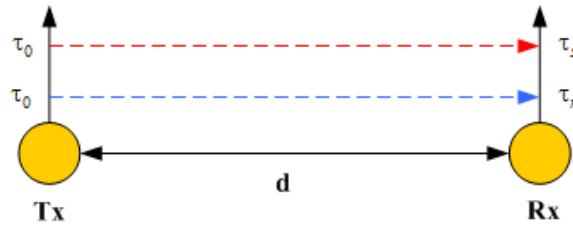


Fig. 2.3. Difference in signal propagation time between source and sink nodes

2.2.3 Time difference of arrival

The distance between two nodes can also be estimated by measuring the difference of signal propagation time, called Time Difference of Arrival (TDoA). It is based on estimating the difference in arrival times at a sink node for two dissimilar signals, such as radio/ultrasound or radio/acoustic sent simultaneously from a source node as shown in Fig. 2.3. The nodes are equipped with extra hardware to send two types of signals having different propagation speeds. To keep a significant difference, the first signal is usually a radio (relaying the data packet) which travels fast enough (at the speed of light) and the second signal is some kind of sound much slower than the radio. Thus, the sink nodes can now estimate the distance by evaluating the difference in arrival times of the two signals as follows.

$$d = (\tau_s - \tau_r) (v_r - v_s) \quad (2.3)$$

Here, v_r and v_s are the speed of radio and ultrasound signals, respectively. Also, τ_r and τ_s are the arrival times of the radio and ultrasound signals, respectively. This method often results in less error in distance estimates. However, it increases the cost for the nodes due to the use of additional hardware to send the second signal still having a lower range (3 m to 10 m).

2.2.4 Angle of arrival

The angle between two nodes can also be used in localization systems by measuring the parameter called Angle of Arrival (AoA) or Direction of Arrival (DoA) of the signal transmitted by the source nodes as shown in Fig. 2.4. This angle is often estimated at the sink node using an electronic compass, or relative to another received signal. By using a directional antenna or using an array of antennas (usually a linear array with uniform spacing between elements), the angle estimation is based on the time of arrival of the signal received at each array element. Thus, the angle of arrival for a signal transmitted by a source node is estimated using equation (1.11) which is re-written as follows.

$$\theta = \sin^{-1} \left(\frac{\arg(\mu)}{\beta \Delta} \right) \quad (2.4)$$

Here, θ is the angle of arrival of the signal received. Also, β and Δ respectively indicate wave number and inter-element spacing. Assuming the translational invariance characteristics of the array, μ is called the eigenvalue of a unique non-singular transformation matrix corresponding to the signals received. It is possible to have a very low angle estimation error (on the order of a few degrees) from this method. However, the need for additional hardware and the need for minimum distance between array elements results in higher cost and larger size for tiny nodes.

2.2.5 Communication range

The distance between two nodes can be roughly calculated using the communication range of the nodes (d_{max}) as shown in Fig. 2.5. Usually, such methods can be useful for WSNs with manual placement of nodes keeping a uniform distance between them. Thus, by counting the number of hops for relayed data packets received at a sink node, the distance from the source nodes can be estimated.

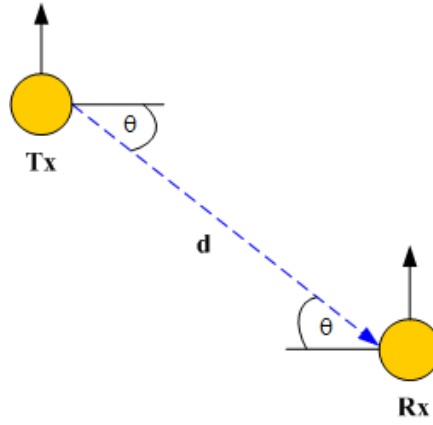


Fig. 2.4. Direction of arrival of signal between source and sink nodes

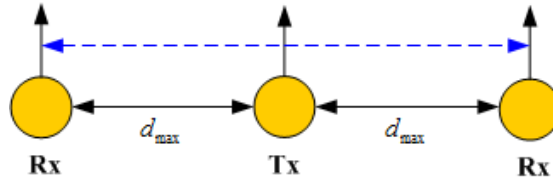


Fig. 2.5. Communication range between nodes

Likewise, the distance between a source and a sink node can be assumed to be between zero and the maximum communication range for WSNs with random node deployment. Considering a midpoint approximation to find the distance of a node, the maximum error will be only half of the communication range in this case. It is a simple method with the lowest cost as it does not require any additional hardware and no additional computations are needed for distance estimation. However, a distance estimation error on the order of half the communication range can often be inappropriate for most localization systems.

2.3 Position computation techniques

These methods aim to compute the position of a node on the basis of distance and/or angle information and reference positions. These are used in the second component of a localization system. Several methods are commonly used to compute the position of nodes in the literature [19]. However, the performance of a localization system often depends on the choice to use a particular method. Available information and processor limitations are always the basis for such a choice. Some popular position computation methods are as follows.

2.3.1 Trilateration

It is a simple method that uses distance information to compute the position of an unknown node in space. In this method, the position of such a node is computed as the point of intersection of three circles as shown in Fig. 2.6. Each circle is formed to have a center at the position of a reference node and has a radius of the distance to the node. Thus, trilateration methods involve three reference nodes to estimate the position of the k -th node in WSNs, using the equation of circles, as follows.

$$(x_k - x_{ai})^2 + (y_k - y_{ai})^2 = d_i^2 \quad (2.5)$$

where, $i = 1, 2, 3$. Also, (x_k, y_k) is the position of the k -th node, (x_{ai}, y_{ai}) is the position of the i -th reference node, and d_i is the estimated distance between the i -th reference node and the k -th node. Thus, it is now possible to easily compute the position of the unknown node by solving these three quadratic equations for two unknown variables, x_k and y_k .

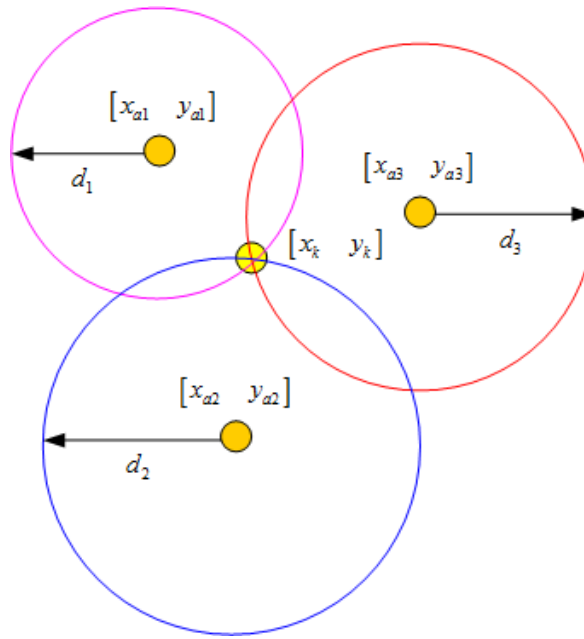


Fig. 2.6. Position computation with trilateration

However, position computation can be problematic for inaccurate distance estimation and/or retrieval of inaccurate position information from reference nodes in real-time applications. In such cases, three circles can intersect at multiple points, resulting in an infinite set of possible solutions.

2.3.2 Multilateration

It is a very useful method that considers the error of the distance estimations as shown in Fig. 2.7. This method can estimate the position of unknown nodes by solving a set of equations as in an overdetermined system. Usually, an overdetermined system always involves more equations than unknown variables and often yields a unique solution for a system of linear equations. Thus, the position of unknown nodes can be computed by considering n number of reference nodes ($n > 3$) and formulating a set of linear equations. Assuming a distance estimation error (ε_d), an overdetermined system of equations takes the form as follows.

$$(x_k - x_{ai})^2 + (y_k - y_{ai})^2 = d_i^2 \quad (2.6)$$

where, $i = 1, 2, \dots, n$. Also, $d_i = d'_i - \varepsilon_d$, and d'_i is the actual distance between the i -th reference node and the k -th node. Also, considering ε_d is a random variable with zero mean and subtracting all equations by the last one, they become a system of linear equations, which can be represented in the matrix form as follows.

$$A_0 x = b_0 \quad (2.7)$$

where,

$$A_0 = \begin{bmatrix} 2(x_{a1} - x_{an}) & 2(y_{a1} - y_{an}) \\ \vdots & \vdots \\ 2(x_{a(n-1)} - x_{an}) & 2(y_{a(n-1)} - y_{an}) \end{bmatrix}$$

$$x = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

$$b_0 = \begin{bmatrix} x_{a1}^2 - x_{an}^2 + y_{a1}^2 - y_{an}^2 + d_1^2 - d_n^2 \\ \vdots \\ x_{a(n-1)}^2 - x_{an}^2 + y_{a(n-1)}^2 - y_{an}^2 + d_{n-1}^2 - d_n^2 \end{bmatrix}$$

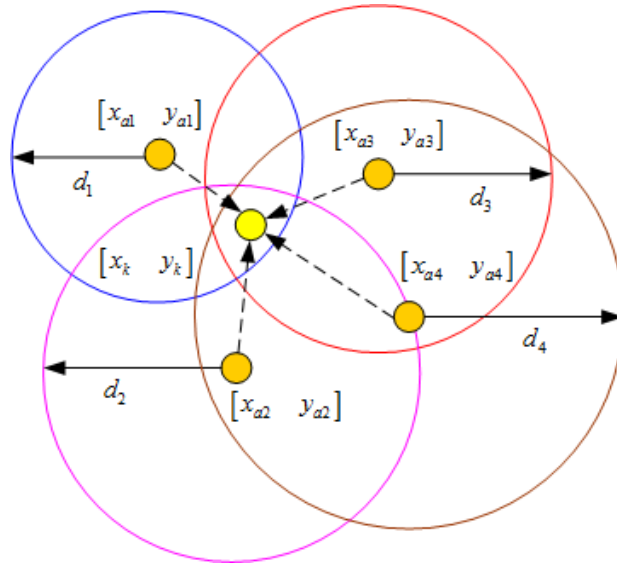


Fig. 2.7. Position computation with multilateration

An approximate solution to the overdetermined systems can be easily found by evaluating standard least squares formula obtained from the problem as follows.

$$\min_x \|A_0 x - b_0\| \quad (2.8)$$

The solution to such problem can be represented as follows.

$$x = (A_0^T A_0)^{-1} (A_0^T b_0) \quad (2.9)$$

Thus, it is an iterative process that aims to minimize the distance estimation error by summing the squares of all the differences obtained in the distance estimates. Such a difference usually arises due to the use of two different distance estimation methods, such as

- estimate distance of the unknown node from a reference node using the RSSI, and
- estimate Euclidean distance using the estimated position and reference node position.

Therefore, the position computation is achieved by using the following equation recursively.

$$x = \min \left\{ \sum_{i=1}^n \left(\sqrt{(x_k - x_{ai})^2 + (y_k - y_{ai})^2} - d_i^2 \right)^2 \right\} \quad (2.10)$$

However, this method has a more complexity of $O(n^3)$ for position computation. Here, n is the number of equations or number of reference nodes.

2.3.3 Triangulation

It is also a simple method that uses angle information instead of distance to compute the position of an unknown node. In this method, the position of such a node is computed as the point of intersection of two straight lines as shown in Fig. 2.8. Each straight line is formulated by keeping the position of a reference node on it and also forming a slope ($\tan \theta_i$) with the angle of arrival of the signal received from the node. Thus, triangulation methods involve two reference nodes to estimate the position of the k -th node in the WSNs, using the straight lines equation, as follows.

$$y_k - y_{ai} = \tan \theta_i (x_k - x_{ai}) \quad (2.11)$$

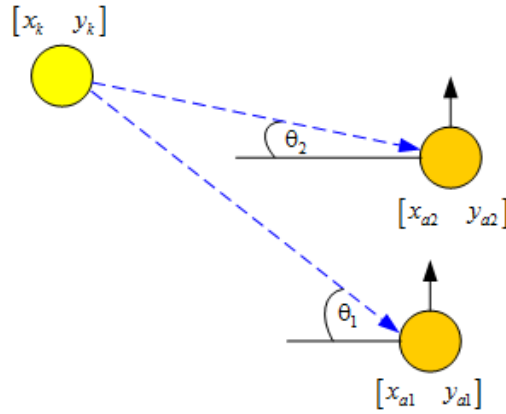


Fig. 2.8. Position computation with triangulation

where, $i = 1, 2$. Likewise, this method may also be suitable for computing position using both angle and distance information. In this case, it uses simple trigonometric formulas of sine and cosine functions, as follows.

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} x_{ai} + d_i \cos \theta_i \\ y_{ai} + d_i \sin \theta_i \end{bmatrix} \quad (2.12)$$

where, $i = 1, 2$. Due to the symmetry of the geometry, a reference node at the center of a circle having a radius of distance to a node placed on the circumference, always estimates the same angle for two distinct positions. Thus, it also takes two reference nodes to find the correct position of an unknown node in WSNs.

2.3.4 Bounding box

It uses squares instead of circles as in trilateration keeping the possible positions of an unknown node within the bounds of a box as shown in Fig. 2.9. This method defines a unique bounding box for each reference node i , keeping such a node in the central position of a square having sides of $2d_i$ and vertex coordinates as follows.

$$\left\{ (x_{ai} \mp d_i), (y_{ai} \mp d_i) \right\} \quad (2.13)$$

It is easier to find the intersection of all bounding boxes, by creating a small rectangular box to confine the unknown node, as it does not require floating point operations. Thus, it is computed by taking the maximum and minimum of the low coordinates and high coordinates of all the bounding boxes, respectively, as follows.

$$\max \left\{ (x_{ai} - d_i), (y_{ai} - d_i) \right\} \quad (2.14)$$

and

$$\min \left\{ (x_{ai} + d_i), (y_{ai} + d_i) \right\} \quad (2.15)$$

Finally, the position of an unknown node is computed as the center of the intersection of all bounding boxes, as follows.

$$x = \left\{ \frac{\max(x_{ai} - d_i) + \min(x_{ai} + d_i)}{2}, \frac{\max(y_{ai} - d_i) + \min(y_{ai} + d_i)}{2} \right\} \quad (2.16)$$

Although it results in more localization error than trilateration, it requires less CPU resources to compute the intersection of squares compared to computing the intersection of circles in trilateration.

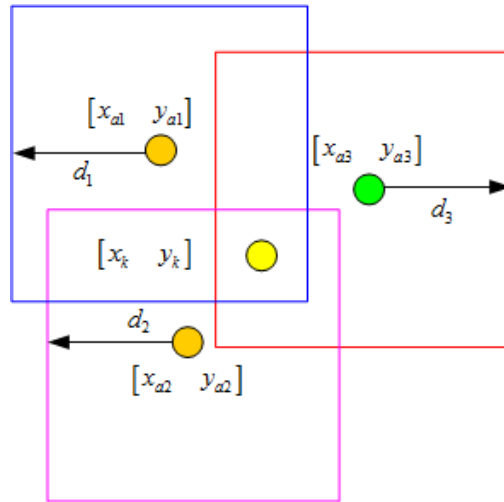


Fig. 2.9. Position computation with bounding box

2.3.5 Probabilistic approaches

It is a very useful method for computing the position of an unknown node while accounting for uncertainty in distance estimates. In this method, the position computation is based on considering a set of points with certain probabilities to be the actual position of the unknown node as shown in Fig. 2.10. It generally models the error of the distance estimate as a random variable. An approximation of the position of the unknown node is computed over several samples, as follows.

- if an unknown node only receives a packet from a single reference node, it has an equal probability of being placed around the reference node.
- if two packets are received from two reference nodes, it has a greater probability of being placed at a pair of intersection points.
- if more packets are received from multiple reference nodes, it has the greatest probability of being placed at a unique point.

Thus, it is possible to identify the probable location of the unknown node by considering the point with a greater probability. However, it has a higher computational overhead and requires more memory space to store information. The complexity of this method is $O(n^2)$. Here, n is the sample size, assumed to be a grid of $d \times d$. It is suitable for centralized localization systems having a more powerful central node to compute positions on gathered information.

2.4 Overview of localization systems

Due to the variety of applications, solutions to the localization problem take a different form. However, the primary requirement for the development of a localization system always remains the same, minimizing errors and maximizing energy efficiency in the computation. Thus, several aspects such as number of beacons, number of nodes, deployment scenario, and use of GPS receivers, etc. are considered for its implementation. Also, the choice to use a suitable system often depends on various attributes such as available resources, network topology and its protocols, and acceptable mean error for a particular application. Some important localization proposals are discussed to show their operational feasibility and limitations in different application domains [20].

2.4.1 Anchor-free

These systems are simple and able to compute a relative estimate of the positions of any unknown node using local distance information.

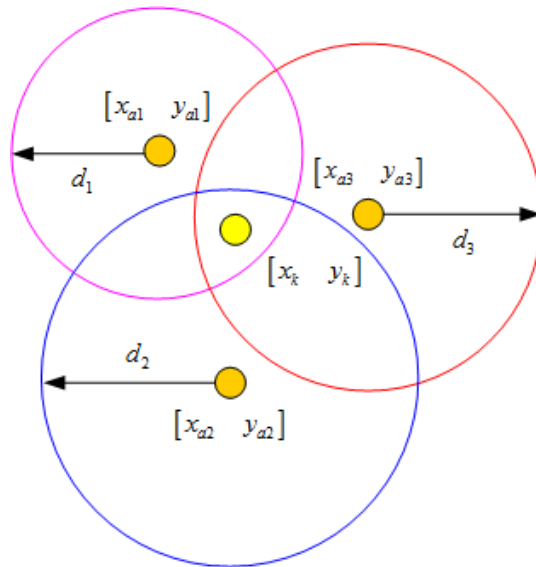


Fig. 2.10. Position computation with probabilistic method

However, they have an inherent limitation that the coordinate system is not unique and must be integrated into the global coordinate by translation, rotation, and flipping. They also have the disadvantage that the positions of the stationary nodes must be recomputed if the reference node moves. Thus, a node usually starts computing its position independently with a random assignment of initial coordinates and converges to a consistent solution using only local node interactions [26]. Such systems can be of two types such as, incremental (initially starts with a core of three or four nodes, then iteratively grows to include appropriate nodes based on measured distances) and concurrent (all nodes compute and refine their positions simultaneously). However, an incremental system leads to poor assignments of global coordinates since it propagates computation errors. In some works, the difference between measured and computed distances is reduced by using an iterative optimization algorithm. Thus, it creates a local map dividing the network into small overlapping sub-regions and these local maps are then stitched together to form a single global map [27]. Also, connectivity information is used to approximate the distance between each pair of nodes in some other works [28].

2.4.2 Anchor-based

These systems involve some nodes with known positions (by manual placement or using GPS receivers) in computing the positions of unknown nodes. Thus, they have the advantage that nodes can locate themselves directly relative to a global coordinate system. However, the accuracy still depends on the number of anchors and their distributions in the sensor field. In some works, nodes receive messages from a mobile beacon and transmit proximity information to build and maintain position estimates [29]. It achieves localization accuracy on the order of a few meters without the need for additional hardware. In other works, a mobile anchor is used to periodically broadcast messages with its current position [30]. Nodes within communication range receive and record the first and last anchor location to estimate their positions. Also, some proposals consider wireless channel variation due to long-distance propagation, fading and irregular coverage [31]. It shows that in a noisy environment, the probability of correctly receiving packets from a mobile anchor is proportional to the Signal-to-Noise Ratio (SNR). It is more consistent and very accurate but not energy-efficient as packets received only for the maximum RSSI are useful and the rest is useless. Moreover, other works have adopted a mobile beacon to estimate the positions of unknown nodes in WSNs [32, 33, 34, 35].

2.4.3 Range-free

These systems are often used to estimate the relative positions of nodes. These are very simple and cost-effective localization systems in WSNs. In some works, the positions of unknown nodes are computed by determining the centroid of all anchor positions near them [36]. It is simple but the localization error is high. Likewise, other work, called DV-Hop, is based on a distance vector routing mechanism in which anchors initiate broadcast messages with current locations by initializing a hop-count parameter to one [37]. The hop-count value is incremented at each intermediate hop by flooding the messages. It estimates the average distance per hop by obtaining all the information from other anchors, and propagates this information to nearby nodes. However, it needs more anchors to reduce distance estimation errors. Some works also use an area-based test, called APIT (Approximate Point-In-Triangulation), which estimates the positions of unknown nodes by constraining them to triangular regions formed with different combinations of anchors [38]. The position of the node is computed as the Center of Gravity (CoG) of the intersection of all triangles. In another work, called Amorphous, hop distance is transmitted to nodes by beacon message propagation [39]. Nodes compute the average of all hop distances collected from their neighborhood. However, it requires knowing the density of the network a priori. The error due to low resolution is compensated by deducting half the radio range from the average estimate.

2.4.4 Range-based

These systems rely on distance or angle information between nodes to estimate their positions. Thus, they operate in the two successive phases, estimating the distance or the angle between neighboring nodes according to the characteristics of the received signal, then computing the actual positions from all these distance or angle estimates. However, the accuracy depends on some factors such as the measurement techniques used, the calibration noise of the device, and the environmental conditions. In some works, the feasibility and quality of self-localization in nodes are evaluated using RSS measures [40]. It assumes that each node is equipped with an array of directional antennas. However, the drawback is that it does not consider the path loss model for signal propagation. Likewise, another work is based on the estimation of AoA information between neighboring nodes in a noisy environment [41]. It considers the propagation of beacon messages over several hops. It shows that good accuracy is achievable despite inaccurate angle estimates and with fewer beacon messages. Also, other works have discussed various methods for estimating the position of nodes based on angle or distance measurements in WSNs [19, 42, 43].

2.4.5 Centralized

These systems collect all information regarding connectivity and distance/angle measurement per pair in the entire network and send it to a central Base Station (BS) powerful enough to analyze and compute the positions of all nodes. The computed positions are then sent back to the respective nodes. Thus, the main advantages of such systems are their simplicity and ease of implementation. They are more accurate than other implementations due to the existence of global information. Moreover, they avoid computation in each individual node, reducing computational overhead and energy consumption at the nodes. On the other hand, they suffer from the critical risk of single point of failure and usually require the central master node with high computing and networking capabilities. Data transmission in the network (sending all information to the base station and computed positions to the nodes) leads to latency, more energy consumption (communication cost) and bandwidth. Another drawback is the inability to access data appropriately and inadequate scaling. Thus, it is more accessible for small-scale networks. However, these systems often use two or three central base stations to reduce communication overhead and solve the scalability problem. In the literature, some popular centralized localization systems are MDS-MAP (Multi-Dimensional Scaling-Mobile Assisted Programming), LBSA (Simulated Annealing Based Localization), and SDP (Semi Definite Programming) [44].

Some work, called IMDS-MAP, is an improved version of MDS-MAP providing greater accuracy in large-scale network applications [45]. Basically, the MDS-MAP displays distance type data in a geometric visualization using data analysis and information visualization. It creates a distance matrix computing the shortest distance between all pairs of nodes, then applies multi-dimensional scaling to determine the relative location of the nodes. It estimates the absolute position of nodes by transforming relative positions when sufficient anchors are available. But, it has some drawbacks such as requirement of global information and high communication and computational cost. Also, some work based on TDoA measurement with ultrasound and RF signals is reported for indoor applications. It performs 3-D (Three-Dimension) localization forming a cluster over the entire network [46].

2.4.6 Distributed

These systems allow all nodes to compute their positions autonomously, often requiring only limited communication with neighboring nodes. Nodes refine their own estimate for more accuracy by combining information received from neighboring nodes. Thus, the main advantages of such systems are reduced computational complexity (equal load on each node for small traffic) and communication cost (no need to send all collected data to a central node and transmit computed positions at all nodes), resulting in significant energy savings. They are more scalable and flexible for large-scale networks. However, they also have some drawbacks such as their difficult implementation, and the localization accuracy often depends on the number of anchors and the distribution characteristics of the nodes. In some work, called DGL (Distributed Grid-based Localization), each node computes the position in a rectangular coordinate system dividing the network into small square grids [47]. Anchors change the communication range by varying their transmission power. Another work, called DRL (Distributed Range-free Localization), uses a voting mechanism to locate nodes (in 3-D) in an environment of irregular radio propagation [48]. It divides the area of interest into several cubic cells where each anchor votes. Each node estimates the position by taking the average of the center of gravity of the cubic cells (with the highest votes). Likewise, some works are based on the support vector machine that uses the measurement of connectivity as training data [49]. Moreover, some other works are based on RSS information and suitable for indoor applications in mobile networks [50, 51]. However, these systems still assume direct signal communication between all nodes and anchors, which is not feasible for large-scale deployment.

2.4.7 Single-hop

These systems enable direct data communications between source and sink nodes. Thus, they remain cost-effective without requiring relay nodes. However, these systems often remain less energy-efficient as the energy consumption is directly proportional to the hop size (transmission distance). In some works, a type of fingerprint test is used to measure RF signal strength offline for the design of signal strength maps [52]. These maps are used to estimate distance information for an indoor localization system. Likewise, in another work, a few beacons are fixed to broadcast local geographic information to nodes [53]. It uses ultrasound signals to increase the accuracy of distance estimation. Nodes estimate distances in an indoor localization system by receiving consecutive radio frequency signals and ultrasound pulses from anchors. Moreover, in other works, the radio connectivity between a node and a set of anchors is used to determine its coordinates [36]. The nodes are localized by computing the centroid of all the references obtained from the nearby anchors. Some works use an area-based approach for node localization [38]. It divides the network into several triangular regions in which the position of a node is estimated by reducing the potential area of its existence. Besides, another work built on the Monte Carlo aims to improve the localization accuracy in mobile wireless sensor networks [54].

2.4.8 Multi-hop

These systems involve multiple relay nodes in transmitting data packets from source to sink nodes. Thus, they are energy-efficient, reduce hop size, and are suitable for large-scale network applications. However, the disadvantages of such systems are higher deployment cost and localization error (distance/angle estimation error propagates through hops). A localization system based on the Distance Vector (DV) often requires simple computations [55]. It computes the shortest paths between distant nodes using neighborhood information. It estimates the size of the hop using the known locations of the anchors, then guesses the distances between the anchors and the unknown nodes by calculating the shortest path hop distances. Besides, some studies improve location estimates through the exchange of information at neighborhood [39]. It also uses offline hop distance estimates like a DV-Hop system. Likewise, in another work, the accuracy is improved by avoiding the accumulation of errors on the network [56]. It uses the bounding box approach in computing position. Also, the self-positioning of nodes in a mobile ad-hoc network is possible by using the distances between nodes [57]. It primarily builds a relative coordinate system for all nodes in 2-D. Finally, it allows nodes to compute their positions with the highest density factor in the n-hop neighborhood coordinate system.

2.4.9 Directional/smart antenna-based

These systems offer better performance for several advantages such as interference reduction, long transmission range and spatial diversity. The use of directional antennas in localization systems achieves three times the accuracy of omni-directional antennas [58]. It is implemented with Berkeley MICA2 motes, and works well in WSNs for different types of node deployment scenarios. Also, a range-independent system, called SeRLoc (Secure Range-independent Localization), which equips anchors with high-power directional antennas to broadcast beacon messages using asymmetric transmission [59]. A beacon message contains the current position of the anchor and the antenna sector in which the packets are broadcast. A similar system, called HiRLoc (High-resolution Robust Localization), allows nodes to passively estimate their positions with high resolution, without increasing the number of reference points or the hardware complexity of each reference point [60]. The positions of the nodes are computed based on the intersection of the areas covered by the anchors. It has greater accuracy with increased computational and communication complexity. Besides, using an adaptive antenna array, a precise and scalable localization system is reported in other works [61, 62, 63].

2.5 Fragility of localization systems

The overall performance of a localization system always depends on the level of accuracy achieved from its individual components. Therefore, it is essential to protect all components against malicious attacks. To achieve the best result, a localization system must authenticate data before it is processed by a component and preserve the confidentiality and integrity of the data while passing it to its next component. However, it is really a challenging task because localization systems often remain vulnerable to security threats in a hostile environment. The possible attacks on each component and their cumulative effects in a localization system are discussed below.

2.5.1 Attacks on distance/angle estimation

Distance estimations are usually based on measuring received signal strength, time of arrival, time difference of arrival, and hop count. Therefore, by exploiting a few compromised nodes, an adversary can easily generate erroneous distance estimates in RSS-based systems. Neighboring nodes would be tricked into considering them closer/farther from their actual locations as they send data packets with higher/lower transmission power. Moreover, the delay in the transmission of data packets can lead to distance estimation problems in ToA/TDoA-based systems.

Likewise, compromised nodes can mislead distance estimates in neighboring nodes by tampering computed data and advertising erroneous hop count information in hop-count-based systems. On the other hand, an adversary can alter the physical medium of signal propagation by introducing noise, obstacles, or smoke on particular regions in a compromised environment (space/radio). Thus, it interferes with the measurement of signal strength and time of arrival. Also, deploying magnets in some specific areas can produce erroneous angle estimates in AoA-based systems.

2.5.2 Attacks on position computation

Position computations mainly involve estimated distances/angles and known positions of a few reference nodes. An unknown node can compute its position using at least three reference nodes when such computation is based on the estimated distances. Likewise, it can perform the same task requiring only two reference nodes when the computation is based on the estimated angles. Thus, an attack on the distance/angle estimates can also affect the position computation. However, direct attacks by advertising erroneous known positions of the reference nodes often affect the position computation more severely. In such cases, the computation of the position would become erroneous even for a correct estimation of the distance. This can happen when a compromised node sends the data packets with incorrect position information and deceives neighboring nodes into appearing as a separate node identity. On the other hand, GPS data retrieval can be erroneous in a compromised environment (the jamming of GPS signals leads to wrong estimates of beacon node positions from GPS receivers).

2.5.3 Attacks on localization algorithm

Due to the characteristics of the multi-hop network, a distributed algorithm is often preferred for a localization system. Therefore, it inherently has the same vulnerabilities as other distributed systems. Common attacks on a distributed system are as follows [13].

- *Denial of service* (adds malicious data by capturing and reprogramming some benevolent nodes).
- *Replication/clone* (produces huge replicas/clones of captured nodes by extracting encryption/decryption keys).
- *Sybil* (sends messages by fabricating a new identity or stealing an identity from a legitimate node).
- *Wormhole* (tunnels a distorted data packet to another location by replaying it several times).
- *Forgery, alteration and interference* (sends misleading information by tampering with them).
- *Replay* (replays outdated information by making congestion in data transmission).
- *Selective forwarding* (drops sensitive messages by refusing to forward them anymore).

Such attacks introduce erroneous information regarding distance/angle estimates, reference positions, number of hops, and nonexistence of nodes/beacons and deteriorate the performance of localization systems.

2.6 Overview of secure localization systems

In WSNs, attacks should be avoided to ensure the reliability of relayed data packets to accomplish the desired tasks. Thus, the incorporation of a secure localization system is always essential for the proper functioning of the network deployed in hostile environments. Moreover, no system can provide a complete safety and security solution against all kinds of attacks.

This issue becomes more complicated due to resource constraints and always requires a simple and acceptable solution for a specific problem. In the literature, several security proposals rely on cryptography as a second line of defense and are usually combined with other techniques. Some pioneering works are discussed to show their effectiveness and potential weakness in providing security against attacks.

2.6.1 Cryptography

Cryptography is very useful to protect the localization systems against external attacks. It preserves the message integrity in relayed data packets and confidentiality of node identities using an authentication process. In some cryptographic schemes, pairwise key distribution is used to secure data communications between nodes. Thus, it ensures the security against node capture attacks by establishing a triple key among three nodes [64] and integrating the keys refreshment phase [65]. However, cryptography generally introduces more communication and computational overheads. Moreover, these methods are not suitable as the keys/passwords stored in the captured nodes may be easily accessible to attackers. As a result, other works are based on the process of updating and revoking keys to resist insider attacks [66]. It provides security against packet drop attacks using higher connectivity and a lower compromise ratio in a low-cost network infrastructure. Likewise, some methods eradicate node replication attacks by verifying the number of shared starting keys [67]. Such random key distribution requires low power consumption, but needs higher memory storage and communication costs. The combined localization and keys distribution protocol [68] becomes sufficiently robust against wormhole attacks. It uses a single master node to manage all interactions with the GPS device and achieves high localization accuracy. But, this requires an IG (Inertial Guidance) module and a GPS device when deploying the nodes. Also, some other works describe the self-healing key managements based on a modified access polynomial and a sliding window [69]. It ensures the security of the access polynomial by dynamically updating the pairwise keys shared between member nodes and group managers. The sliding window mechanism reduces communication cost and resource consumption. However, the computational cost becomes significant in this case.

2.6.2 Anomaly detection

Identifying and eliminating all misbehaving nodes/anchors always ensures greater accuracy in computing the positions of benign nodes in the network. Thus, some deterministic distributed protocols [70, 71] and voting techniques [72, 73] with a modified mesh of different trust grades [74] effectively recognize and isolate malicious nodes/anchors. Distributed protocols are very efficient in communication time, power, memory, and computation. But the detection of false nodes depends on the prescribed number of trusted nodes. On the other hand, the voting techniques are simple and do not require any additional hardware, but employ complex computations attaining a bounded estimation error. Trust evaluations ensure a higher detection rate and localization accuracy, but result in extra delays. Other methods also identify malicious nodes by using witness nodes in a cell [75], or hide benevolent node locations with an orthogonal code [76]. The use of witness nodes reduces communication costs effectively. It also improves the detection rate, but requires higher memory. However, an orthogonal code often shows a higher detection rate at lower communication costs, memory usage, and power consumption. But it makes the algorithm a bit more complex. Some work also prevents the legitimate nodes from self-destruction by misleading adversaries with false information [77]. It shows negligible costs to communication, memory, and processing, but increases power consumption. The sequential hypothesis testing [78] and software attestation methods [79] detect and revoke the compromised nodes. They minimize power consumption and communication costs. But any change in the sampling strategy affects the attestation costs.

2.6.3 Location verification

Checking the consistency of the localization accuracy based on some known models and observations improves the reliability of the computed positions. It often involves deployment knowledge using a group-based deployment model and known physical properties of both radiofrequency and ultrasound to compute distances/angles. Thus, some works verify the positions with a few nodes called verifiers [80] and applying probability theory [81]. However, verifiers frequently need special equipment like directional antennas and precise time measuring devices which introduce huge overheads. Moreover, it incurs little computation and communication costs. But, a higher detection rate is possible with a lower false-positive rate. In the probability theory, malicious nodes are detected treating them as competitors in a non-cooperative game. But it requires a large number of trusted verifiers to minimize the maximum deception from the malicious node. Likewise, in some works, false locations are detected during Time of Arrival (ToA) measurements [82] and a bounded localization error is ensured by keeping the number of cheating anchors in the prescribed limit [83]. A higher detection rate is obtained by assuming zero synchronization errors among the verifiers. But, it is almost unrealistic to ensure always zero synchronization errors. However, a mathematical analysis derives the necessary and sufficient conditions for a secure and robust distance-based algorithm. But, it shows considerable execution complexity for a polynomial-time algorithm. Also, a centralized method based on SDN (Software-Defined Networking) protects the privacy of the location and identity of the anchor in a heterogeneous WSN [84]. It has better energy efficiency and higher positioning accuracy with low communication costs. But it requires additional hardware like an SDN controller.

2.6.4 Robust computation

Filtering inconsistent distance/angle measurements becomes useful to improve position computation accuracy in harsh environments. Thus, some works based on the information theory [85] or mutual authentication and validation [86] are suitable for this case. The information theory provides a high detection rate and minimal localization error but has significant time complexity. In contrast, mutual authentication and validation shows higher efficiency in time consumption and localization accuracy but have communication overhead. Likewise, the reputation management based on a game theory [87] and the sequential probability ratio testing [88, 89] are very effective in the verification of the anchor reliability and the detection of static/mobile compromised and replicated nodes. The game theory optimizes the energy consumption of nodes but becomes less accurate in position estimations. Instead, sequential hypothesis testing dramatically reduces false-positive and false-negative rates but offers significant storage and computational costs to each node. In some works, the replicated nodes are also detected by keeping neighbors and liars below a given threshold [90]. It minimizes the necessity of trust and communication costs but requires prior knowledge of node deployment. Besides, the iterative gradient descent approach [91] and multiregional algorithm based on compressive sensing [92] reduce the impact of inconsistent measurements due to malicious nodes in hostile environments. However, the iterative gradient descent reports higher localization accuracy in the mobile network environment but does not emphasize false alarms against the elementary attacks. The compressive sensing also achieves a higher level of localization accuracy at lower power consumption but includes much communication and computational costs.

2.6.5 Simple systems with extra hardware

Some proposals based on FM (Frequency Modulation) [93] or analysis of the neighboring node information [94] improves the security performance and positioning accuracy in the networks. The frequency modulation shows lower communication and computational costs at a lower system complexity. But it always requires additional hardware such as an FM signal receiving module to load onto each node.

The neighboring node information always ensures a low false detection rate in a higher density network while increases memory usage and power consumption. Some distance bounding protocols with UWB (Ultra-Wideband) transceivers isolate malicious nodes in indoor environments in real-time [95, 96, 97]. The UWB technology often requires special equipment like ultrasonic transceivers that are very expensive. However, it leads to higher detection accuracy and a lower false alarm rate but introduce sufficient communication costs. It ensures a high security, low energy consumption, and high precision in position estimations with a minimum number of anchors. Also, it shows that a probabilistic model is well-suited to evaluate the security of the ranging/positioning solution during enlargement attacks.

2.7 Summary

In this chapter, we have studied some popular methods that work on individual components of a localization system. For its proper functioning, the level of accuracy, the computational complexity and the feasibility of the application of these methods are also described. Next, we discussed common security issues and vulnerabilities of each component to malicious attacks. A brief overview of existing solutions for securing localization systems is also included. It is observed that most of the state-of-the-art localization systems use anomaly detection and blocking to mitigate attacks in WSNs. Few systems also consider location verification in their final stages or use probabilistic outlier filtering to eliminate any inconsistent estimates. However, these systems often overlook security concerns in the physical layer and thus localization systems become fragile. In the literature, only a few systems have addressed this problem using the transmit power control/jamming technique. But, they are also not so effective in a compromised environment. For several important WSN applications, especially military operations need precise estimation of node locations for target detection and tracking. Thus, existing localization systems may not be sufficient to fulfill such requirements by isolating malicious nodes under physical layer attacks. From this perspective, the introduction of smart antennas would be a good strategy to keep them away from radio links. This is an open research area and hence, this research work mainly focuses on the design and implementation of secure localization systems based on smart antennas for wireless sensor networks.

Chapter 3

Resilient localization and anchor mobility control strategies

3.1 Introduction

A wireless sensor network consists of many tiny nodes with limited resources. For better performance and longer life, it is always necessary to reduce factors such as errors, energy consumption, system complexity and deployment cost. The literature review reveals that range-based systems have the highest accuracy, while energy efficiency often remains higher in multi-hop systems. Likewise, distributed systems reduce computational complexity and anchor-based systems are able to compute global coordinates of nodes. On the other hand, nodes require special hardware to estimate the distance/angle of received signals and each anchor needs a GPS receiver to get its current reference. Thus, it increases the implementation costs to some extent. Also, the signal measurement always depends on the channel characteristics and keeping the error within an acceptable limit remains a difficult task. Therefore, a trade-off between all these factors is always important when implementing a localization system. From this point of view, the use of a single mobile anchor with a GPS receiver and a smart antenna would be useful to reduce the cost. By controlling its mobility on trajectory, it is possible to keep the communication range between the anchor and the nodes to a minimum, which leads to the least energy consumption during the relay of data packets. It also ensures the least probability of error in distance/angle estimation by using the smart antenna for signal measurement.

In this chapter, we have proposed a robust localization system, called Row Matching Algorithm (RMA). It estimates node positions based on RSS and AoA information. The anchor estimates the distance and angle of signals received from neighboring nodes. It conveys messages (containing this information and its current reference) to neighboring nodes. Nodes can autonomously estimate their positions by obtaining only two of these messages from the anchor. It is a simple system producing less errors and energy consumption in WSNs. Moreover, we introduced two new mobility control strategies for the anchor, based on the centroid (deterministic) method and fuzzy logic. The proposed mobility control methods show an efficiency comparable to the conventional random walk model. However, they often outperform the conventional model against malicious attacks.

The rest of this chapter is organized as follows. In Section 3.2, an overview of network architecture and protocols is discussed. In Section 3.3, the description of the proposed localization system is explained in detail. In Section 3.4, the implementation strategy of the proposed mobility control methods is discussed. Simulation results and performance analysis are presented in Section 3.5. Finally, Section 3.6 discussed the limitations and possible modifications of this system in the future.

3.2 System design preliminaries

In the network, the selection of an appropriate trajectory for the movement of the anchor always leads to faster convergence in the localization process and ensures optimal energy efficiency. Also, completing the localization process in two consecutive anchor points often makes the system secure and robust with feasible infrastructures. In this section, we have described network configurations and radio propagation characteristics for an energy-efficient and robust localization system. All necessary assumptions are also mentioned to facilitate the performance of the proposed localization system and mobility control strategies.

3.2.1 Network architectures and protocols

In WSNs, nodes do not know their location after deployment and use the localization system to estimate their locations autonomously. Thus, network architectures and protocols are also very important for the implementation of a robust localization system. Several assumptions considered for this work are mentioned below.

1. Network infrastructures

- A large number of nodes and an anchor are randomly deployed over the region of interest. Nodes remain stationary and scattered over a 2-D field without obstacles.
- Nodes are configured with unique IDs and limited battery power. They also have limited memory, a simple processor and a half-wave dipole antenna.
- Anchor has a GPS receiver and a smart antenna. It is also mounted on wheels/vehicles to integrate mobility functions. It has ample memory, a computationally efficient processor and the flexibility to recharge/refill its battery power.
- The Base Station (BS) is fixed in the center of the field so that it can communicate equally with all the nodes of the WSNs.

2. Data communications

- The nodes are identical (homogeneous) with similar sensing and processing capabilities. They are configured to periodically broadcast messages with respective IDs. On the network, all nodes consume equal power in transmission and reception of data packets.
- The nodes and the anchor are equipped with an auto-configuration function to switch themselves into different modes (e.g., active, receive, idle, and sleep) to optimize their energy consumption during the localization process.
- Anchors are able to estimate distance/angle information with high resolution for RSS exceeding a prescribed threshold on signal-to-noise ratio (SNR_{th}) using path loss and ESPRIT algorithm.
- In order to have better network coverage and connectivity, the Quality of Service (QoS) in data communications is assumed to be constant within the maximum communication range (d_{max}).

3.2.2 Radio propagation characteristics

Since nodes usually transmit signals in free space, they experience several propagation characteristics such as reflection, refraction, diffraction and dispersion, etc. in the radio environment. Thus, mean power of the received signal is considered as a decaying function of the distance travelled (inversely proportional to the square of distance). This phenomenon is known as path loss which is used to measure distance information in most RSS-based localization systems [98]. Therefore, an appropriate choice for the signal propagation model, defining the path loss more realistically, is also important for developing an energy-efficient localization system [99]. Some assumptions made in this regard are described below.

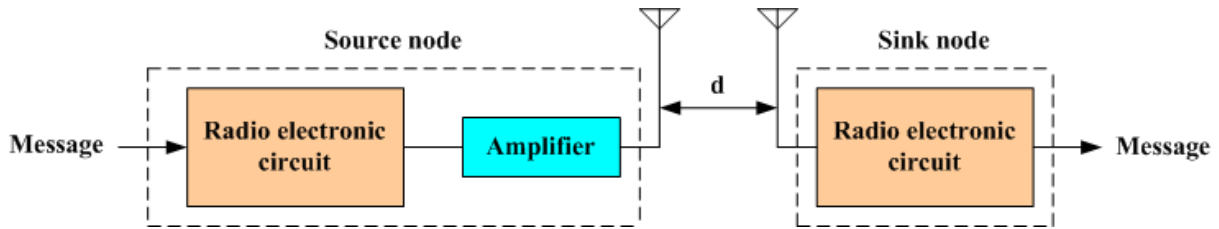


Fig. 3.1. Functional blocks of the MICAz mote

1. Path loss

- Signals flowing through the radio environments are a composite form of a large-scale path loss component, a medium-scale slow varying component and a small-scale fast varying component. The medium-scale slow varying component uses log-normal distributions and the small-scale fast varying component uses Rician or Rayleigh distributions in LoS (Line of Sight) or NLoS (Non-Line of Sight) data communications, respectively.
- Path loss is determined using small scale propagation characteristics with fast signal variations due to multipath fading over short distances (on the order of a few wavelengths) or short duration (of the order of a few seconds).

2. Link budget

- The link budget equations are formulated assuming equal energy dissipation per bit in the electronic circuits (E_{elec}) of the RF transceivers (as shown in Fig. 3.1). Also, the extra energy consumption in the power amplifier circuit (E_{amp}) is considered when transmitting each bit of data at a unit distance over the free space.
- The radio propagation model is considered compatible with MICAz mote [7]. It follows the IEEE 802.15.4 protocols in Wireless Personal Area Networks (WPAN) [100]. In accordance with Media Access Control (MAC) protocols, each node remains in active and receive mode prior to the localization process and then switches to sleep mode.

The power dissipation (Joules per second) of a node is considered as the sum of the power dissipated on its two radio links (forward link and reverse link): to broadcast a message in free space (E_b) and to receive data packets from the anchor (E_r). Thus, the total energy consumption (E_{total}) of the k -th node during localization is expressed (in Joules) as follows.

$$E_{total} = t'_k E_b + 2E_r \quad (3.1)$$

where, t'_k is the number of anchor points required to localize the k -th node.

Now, keeping the analogy with the LEACH (Low Energy Adaptive Clustering Hierarchy) algorithm, where the energy consumption in a node during the transmission (E_b) and reception (E_r) of an n -bit data packet is expressed (in Joules) as follows [3].

$$E_b = n (E_{elec} + E_{amp} d_{max}^2) \quad (3.2)$$

$$E_r = n E_{elec} \quad (3.3)$$

In this work, we have chosen a low power dissipation model for MICAz mote which operates on the 2.4 GHz ISM (Industrial, Medical and Scientific) band and has other parameters like $n = 250$ kbps, $E_{elec} = 50$ nJ/bit, $E_{amp} = 10$ pJ/bit/ m^2 and $d_{max} = 90$ m.

3.3 Row matching algorithm

It is a simple system based on a mobile anchor. Anchor uses the direction-finding and beamforming attributes of the smart antenna to generate and transmit beacon messages. It is possible to compute the position of any node by comparing the estimated positions obtained for two distinct anchor points. The node stores two unique sets of beacon messages received at two separate anchor points. For each beacon message, it produces two sets of positions (actual and imaginary) and keeps them as two separate rows in a table. Thus, the positions computed for these two beacon messages are stored in two separate tables. By comparing the rows of the tables with each other, the position of the node is computed as the mean of the positions which causes the least deviation. Therefore, the localization system operates in three steps, such as

- Cluster formation,
- Beacon message generation/transmission, and
- Position computation.

Here is a detailed description of each step provided below. This system is realized with a random mobility model for the anchor to select the succeeding paths in the network.

3.3.1 Cluster formation

The nodes periodically (with a period T) broadcast messages with their individual IDs by radiating signals of equal transmission power (active mode). The anchor is configured to receive signals (receive mode) above a prescribed signal-to-noise ratio (SNR_{th}) threshold [101]. For each anchor point, a unique cluster is formed with a few neighboring nodes whose RSS exceeds the SNR_{th} . For all clustering nodes, an anchor stores the information needed to estimate the distance/angle and its current reference in a table, U (idle mode). It also keeps a record of the cluster sizes (number of nodes in a particular cluster) it has encountered so far by enabling a counter, Ptr . Algorithm 1 explains the cluster formation process. Each time, it begins with a random anchor point and a finite number of nodes, and continues until the maximum permissible anchor points (t_{max}) are attained or the localization of all nodes is complete (when counter, Ptr reads $2N_s$, i.e. twice the number of deployed nodes). Several node and anchor data are required for cluster formation and these are used as input variables. Likewise, the cluster size, operating mode and counter value are considered output variables.

3.3.2 Beacon message generation/transmission

Due to multipath fading, the signals transmitted from the nodes arrive at the anchor with different strengths and angles. Therefore, the smart antenna is used to eliminate interferences and to estimate AoA information. A simple eigen structure-based AoA estimation algorithm, called TLS-ESPRIT, is used for this work. For each cluster, the anchor evaluates the distance and angle information of all signals received from neighboring nodes using equation (1.4) and equation (1.11), respectively. It stores this data with the respective MAC IDs and also merges its current reference retrieved from the GPS receiver. However, a GPS receiver expresses the anchor reference in latitude (γ') and longitude (ψ') in geodetic coordinates. For position estimation by the geometric method (triangulation), it is necessary to convert it into Cartesian coordinates (x_a, y_a) before such amalgamation. It performs such coordinate transformation as follows [102].

$$\left. \begin{aligned} x_a &= r \cos \gamma' \cos \psi' \\ y_a &= r \cos \gamma' \sin \psi' \end{aligned} \right\} \quad (3.4)$$

where, $r = \frac{a}{\sqrt{1-e^2 \sin^2 \gamma'}}$ is the radius of curvature of the earth at the anchor reference and $e = \sqrt{1 - \frac{b^2}{a^2}}$ is the eccentricity.

Algorithm 1 Pseudo-code for forming clusters and keeping their databases

Input: Maximum number of permissible anchor points (t_{max}), Number of nodes in active mode (N_s), Threshold level in signal-to-noise ratio (SNR_{th}), Received signal strength (RSS), Angle of arrivals of the received signals (AoA), Neighboring node IDs: $[ID]$, Position of the anchor (x_a, y_a) in Cartesian coordinates, and Receive mode of anchor: mode state (ms) = ‘Set’

Output: Cluster size (CS), Pointer value in data table (Ptr), Idle mode of the anchor: mode state (ms) = ‘Reset’

```

1: Initialize:  $Ptr, R = 0, t = 1, ms = \text{'Set'}, N_s, SNR_{th}, t_{max}$ 
2: while ( $t \leq t_{max}$ ) do
3:    $x_a \leftarrow 1000 * rand(); y_a \leftarrow 1000 * rand(); CS \leftarrow 0; i \leftarrow 1$ 
4:   while ( $i \leq N_s$ ) do
5:     if ( $RSS[i] \geq SNR_{th}$ ) then
6:        $CS \leftarrow CS + 1; Ptr \leftarrow Ptr + 1; U[Ptr][1] \leftarrow ID[i]; U[Ptr][2] \leftarrow RSS[i];$ 
7:        $U[Ptr][3] \leftarrow AoA[i]; U[Ptr][4] \leftarrow x_a; U[Ptr][5] \leftarrow y_a$ 
8:     end if
9:      $i \leftarrow i + 1$ 
10:  end while
11:  if ( $Ptr = 2N_s$ ) then
12:    break
13:  end if
14:   $t \leftarrow t + 1$ 
15: end while
16:  $ms \leftarrow \text{Reset}$ 
17: return  $ms, CS, Ptr$ 

```

Node ID (bits)	Distance (meter)	AoA (rad)	Anchor reference (meter)
k	d_k	θ_k	$[x_a \ y_a]$

Fig. 3.2. Format of the beacon message

Also, a and b are respectively the semi-major axis and the semi-minor axis of the ellipse. Thus, for each clustering node, it prepares beacon messages (idle mode) according to the format shown in Fig. 3.2. And, within a cluster, beacon messages are relayed to each node one by one using the First Come, First Served (FCFS) scheduling mechanism (active mode). Each time, it establishes a stable and direct point-to-point radio link via an adaptive beam pattern produced by the smart antenna, where the deep nulls are steered to the directions of the remaining nodes. Fig. 3.3 illustrates such a pattern for a cluster of five nodes, where the main beam is steered to the node located at 20° and the nulls are steered to the others placed at $-30^\circ, -5^\circ, 45^\circ$ and 60° , respectively. Algorithm 2 explains the method of generating/transmitting beacon messages. It requires various data from a cluster (e.g., node IDs, distance, angle, etc.) and antenna design parameters and these are used as input variables. On the other hand, it considers the format of the beacon message and the optimal antenna pattern as output variables.

3.3.3 Position computation

The position computation is based on simple mathematical formulas and takes place at each node. The nodes are periodically regulated to different modes until they accumulate two beacon messages from the anchor. In a cluster, each node receives a beacon message from the anchor (receive mode) and store it in its memory locations shown in table U' . It also periodically checks its database for received beacon messages (idle mode). When a node discovers two beacon messages in its database, it initiates the position computation process.

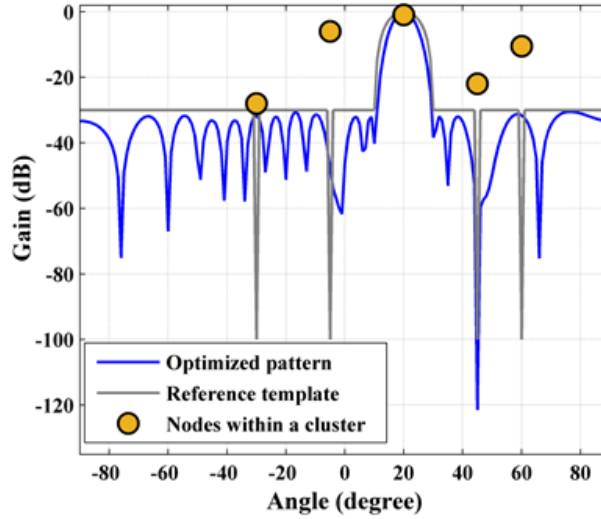


Fig. 3.3. An adaptive beam pattern of smart antenna

Algorithm 2 Pseudo-code for generating and transmitting beacon messages

Input: Angle estimated from the received signals (φ), Distance estimated from the received signals (d), Cluster size: $[CS]$, Clustering node IDs: $U[Ptr][1]$, Anchor reference: $(U[Ptr][4], U[Ptr][5])$, Number of antenna elements ($2N$), Beamwidth between the first nulls ($FNBW$), Sidelobes level (SLL), Depth of nulls (K), Scanning angle (θ) and Idle mode of anchor: mode state (ms) = ‘Reset’

Output: Generated beacon messages: B_m [size], Optimized beam pattern (AF_p), Active mode of the anchor: mode state (ms) = ‘Set’

```

1: Initialize:  $i = R + 1, j, k, n = 1, AF_p = 0, \theta = -90^0, size = 5, K, N, SLL, FNBW, ms = \text{‘Reset’}$ 
2: while ( $i \leq Ptr$ ) do
3:    $\theta_d \leftarrow \varphi[1]$ 
4:   while ( $j \leq size$ ) do
5:      $B_m[j] \leftarrow U[i][1]; U[i][1] \leftarrow d[i]; d[i] \leftarrow \varphi[1]; \varphi[1] \leftarrow U[i][4]; U[i][4] \leftarrow U[i][5]; j \leftarrow j + 1$ 
6:   end while
7:   while ( $\theta \leq 90^0$ ) do
8:     if ( $|\theta - \theta_d| \leq \frac{FNBW}{2}$ ) then
9:        $AF_p \leftarrow 1$ 
10:    else
11:       $AF_p \leftarrow SLL$ 
12:    end if
13:    while ( $k \leq CS - 1$ ) do
14:       $\theta_n \leftarrow \varphi[k + 1]$ 
15:      if ( $\theta = \theta_n$ ) then
16:         $AF_p \leftarrow K$ 
17:      end if
18:       $\varphi[k] \leftarrow \varphi[k + 1]; k \leftarrow k + 1$ 
19:    end while
20:    while ( $n \leq N$ ) do
21:       $AF_p \leftarrow AF_p + W[n] * \cos \left[ \left( n - \frac{1}{2} \right) * \pi * (\sin \theta - \sin \theta_d) \right]; n \leftarrow n + 1$ 
22:    end while
23:     $\theta \leftarrow \theta + 1$ 
24:  end while
25:   $\varphi[CS] \leftarrow \theta_d; i \leftarrow i + 1$ 
26: end while
27:  $R \leftarrow R + CS$ 
28:  $ms \leftarrow Set$ 
29: return  $ms, AF_p, B_m[size]$ 

```

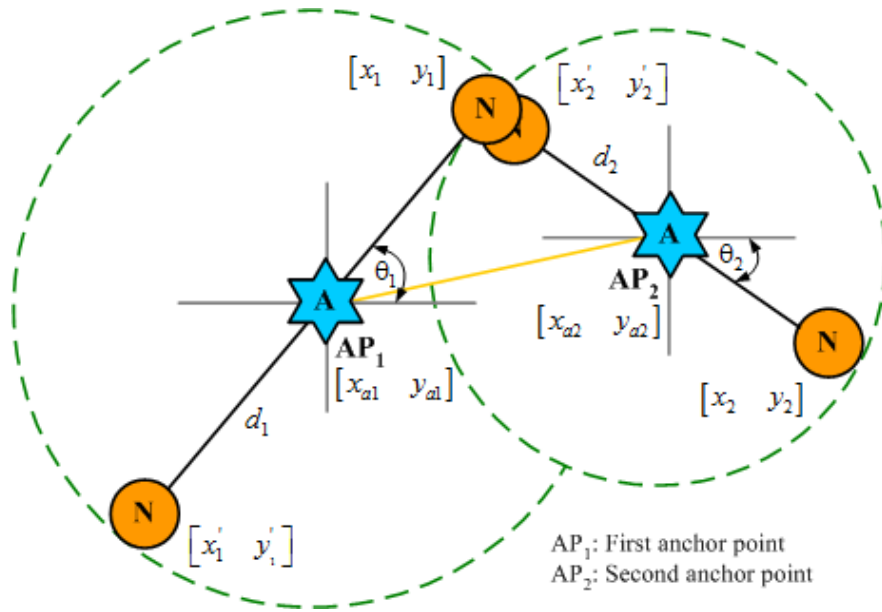


Fig. 3.4. Original and imaginary node positions at two anchor points

Otherwise, it broadcasts the signal (active mode) again. As soon as the node completes the localization task, it turns off its power-regulator (sleep mode) to save energy [103]. For the position computation, each node produces two sets of position data with the two beacon messages and keeps them in two separate tables, P' and Q' , shown in Fig. 3.5. However, each table contains two distinct positions for a node. One is actual and the other is imaginary. The nodes compute the actual and the imaginary position using the estimated angle and the angle (image), respectively. An angle estimated by the anchor remains in the range of $\{-90^0, 90^0\}$. Also, the anchor always estimates the same angular value for a clustering node located at one of the two positions in WSNs. It occurs due to the symmetry property of the geometry. So, the nodes have to calculate the angle θ' , which is supposed to be an image of the estimated angle θ . The angle (image) is evaluated in a counterclockwise direction as follows.

$$\theta' = \pi + \theta \quad (3.5)$$

For i -th anchor points ($i = 1, 2$), the actual and the imaginary positions of a node are expressed as follows.

$$[x_i \ y_i] = [x_{ai} + d_i \cos \theta_i \quad y_{ai} + d_i \sin \theta_i] \quad (3.6)$$

$$[x'_i \ y'_i] = [x_{ai} + d_i \cos \theta'_i \quad y_{ai} + d_i \sin \theta'_i] \quad (3.7)$$

Here, (x_a, y_a) represents the anchor reference.

Fig. 3.4 illustrates the generation of actual and imaginary positions of a node at two anchor points. It shows that a node may exist at any of the three locations (of which one is actual and the other two are imaginary). However, the actual location may slightly deviate in two anchor points due to an erroneous RSS/AoA measurement. Thus, the nodes must take into account a total of four positions. A node employs two steps to compute its correct position, as illustrated in Fig. 3.5. Tables P' and Q' first compare the locations generated at the two anchor points. Finally, it estimates the mean position from the tables producing the lowest deviation (D'_0). Each node keeps the computed position data and the MAC ID in separate memory locations shown in table V' . Algorithm 3 explains the process of beacon message accumulation and position computation. It uses accumulated beacon messages to compute node positions. So, these are considered as the input and output variables. Thus, the nodes and the anchor are both periodically switched to different modes over an acceptable time interval (t_{max}) for position computations. Table 3.1 gives the time allocated to each mode of operation.

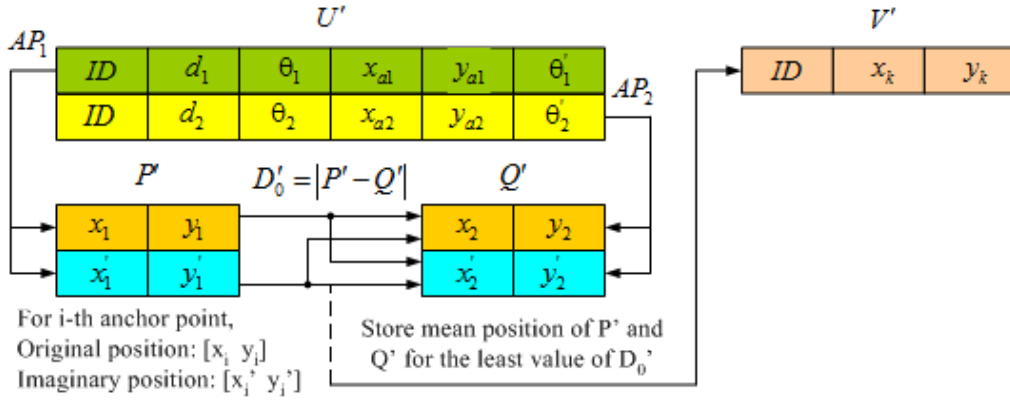


Fig. 3.5. Position computation of nodes in RMA

Algorithm 3 Pseudo code for accumulating beacon messages and computing position in the node

Input: Maximum number of permissible anchor points (t_{max}), Anchor generated beacon messages: B_m [size] and Receive mode of node: mode state (ms) = 'Set'

Output: Accumulated beacon messages: U' [count] [size], Estimated position of the node: $[X]$ and Power saving mode of node (Idle/Sleep): mode state (ms) = 'Reset'

```

1: Initialize:  $i, j, k, t = 1, count = 0, D'_{min} = \infty, ms = \text{'Set'}, t_{max}$ 
2: while ( $t \leq t_{max}$ ) do
3:    $U'[count][size] \leftarrow B_m[size]; count \leftarrow count + 1$ 
4:   if ( $count = 2$ ) then
5:     break
6:   else
7:      $ms \leftarrow \text{Set(Active)}$ 
8:   end if
9:    $t \leftarrow t + 1$ 
10: end while
11:  $ms \leftarrow \text{Reset(Idle)}$ 
12: while ( $k \leq 2$ ) do
13:    $\theta'[k] \leftarrow \pi + \theta[k]; U'[k][6] \leftarrow \theta'[k]$ 
14:    $x[k] \leftarrow U'[k][4] + U'[k][2] * \cos U'[k][3]; y[k] \leftarrow U'[k][5] + U'[k][2] * \sin U'[k][3];$ 
15:    $x'[k] \leftarrow U'[k][4] + U'[k][2] * \cos U'[k][6]; y'[k] \leftarrow U'[k][5] + U'[k][2] * \sin U'[k][6]$ 
16:   if ( $k = 1$ ) then
17:      $P'[1] \leftarrow [x[k] \ y[k]]; P'[2] \leftarrow [x'[k] \ y'[k]]$ 
18:   else
19:      $Q'[1] \leftarrow [x[k] \ y[k]]; Q'[2] \leftarrow [x'[k] \ y'[k]]$ 
20:   end if
21:    $k \leftarrow k + 1$ 
22: end while
23: while ( $i \leq 2$ ) do
24:   while ( $j \leq 2$ ) do
25:      $D'_0 \leftarrow |P'[i] - Q'[j]|$ 
26:     if ( $D'_0 \leq D'_{min}$ ) then
27:        $D'_{min} \leftarrow D'_0; X \leftarrow \frac{P'[i] + Q'[j]}{2}$ 
28:     end if
29:      $j \leftarrow j + 1$ 
30:   end while
31:    $i \leftarrow i + 1$ 
32: end while
33:  $ms \leftarrow \text{Reset(Sleep)}$ 
34: return  $ms, X, U'[count][size]$ 

```

Table 3.1: Clock time allotted for various modes

Clock time allotted (in seconds)	Mode of operation	
	Anchor	Node
t_0	Receive	Active
t_1	Idle	Idle
t_2	Active	Receive
t_3	Idle	Idle
t_4	Sleep	Sleep

Table 3.2: Control signal input for various modes

Anchor	Node
00	Sleep
01	Idle
10	Receive
11	Active

Two separate counters are activated for the nodes and the anchor to acknowledge the prescribed number of beacon/broadcast messages. Fig. 3.6 shows the timing sequence and Table 3.2 gives the control signal to trigger each mode. Here, the first and last bit represent the status (on/off) in the node/anchor sensing and processing devices, respectively.

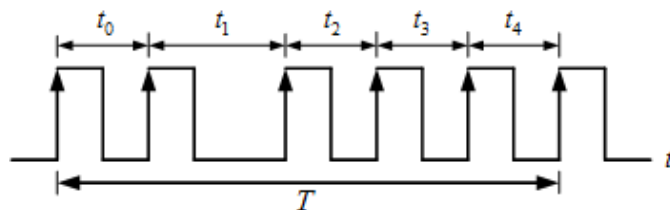
3.4 Mobility control strategies

Intelligent control of the anchor mobility often becomes useful for the development of an energy-efficient and secure localization system. The proposed localization system requires only two beacon messages from two separate anchor points. Thus, keeping the anchor on such a trajectory that explores a larger number of nodes in a cluster (larger cluster size) speeds up the localization process. Unlike random mobility, keeping the same nodes in two successive clusters ensures better energy efficiency in nodes with limited resources. Similarly, such a strategy is worthwhile for identifying possible anomalies between two successive clusters and blocking malicious nodes. However, sometimes the anchor cannot find any nodes (active mode) in a cluster. This can happen when most nodes are localized (sleep mode) and the localization process is about to complete in the WSNs. In this case, the anchor is supposed to follow a random trajectory until it discovers a nearby node.

We have proposed two strategies for controlling the mobility of the anchor, such as

- Centroid-based (deterministic), and
- Fuzzy-based.

These strategies are mainly focused on achieving two basic objectives, such as

**Fig. 3.6.** Synchronizing pulse for mode switching

- improve energy-efficiency, and
- preserve the security of localization systems.

To make localization systems energy-efficient, the anchor must relay beacon messages earlier to nodes with lower RSS within a cluster. Likewise, to secure localization systems against attacks, the anchor must monitor the behaviors of nodes by keeping them in two consecutive clusters. For these purposes, the anchor is controlled to move on a path where the next cluster must include at least the nodes first encountered in the present cluster. The selection of such a path depends on the size of the present cluster (considering only the nodes encountered first) and the priority index given to these nodes. A priority index is inversely proportional to the RSS values of the nodes. Therefore, the proposed mobility control strategies need to find two parameters, such as

- displacement (d_a), and
- direction (ϕ) of the new trajectory.

These parameters are evaluated based on the cluster size (CS), distances (d_k), and priority indices (w_k) of nodes in a cluster. Thus, the distances and angles of the nodes are estimated using equation (1.4) and equation (1.11), respectively. The priority index is expressed as follows.

$$w_k = \frac{d_k}{d_{max}} \quad (3.8)$$

After evaluating these two mobility parameters, displacement and direction, the new reference (x_{an}, y_{an}) of the anchor is determined as follows.

$$\left. \begin{aligned} x_{an} &= x_a + d_a \cos \phi \\ y_{an} &= y_a + d_a \sin \phi \end{aligned} \right\} \quad (3.9)$$

where, (x_a, y_a) denotes anchor reference in the present cluster.

3.4.1 Centroid-based strategy

In this case, displacement and direction are calculated as follows.

$$d_a = \frac{\sum_{k=1}^{CS} w_k (d_{max} - d_k)}{\sum_{k=1}^{CS} w_k} \quad (3.10)$$

and

$$\phi = \frac{\sum_{k=1}^{CS} w_k (\theta'_k + \theta_k)}{2 \sum_{k=1}^{CS} w_k} \quad (3.11)$$

where, $k = 1, 2, \dots, CS$ and θ'_k is called the angle (image) of the estimated angle θ_k , which is evaluated using equation (3.5).

Algorithm 4 explains the centroid-based mobility control strategy for the anchor. It requires several data (e.g., RSS, AoA, and cluster size, etc.) from a cluster to start the process and hence, these are considered as the input variables. The new anchor reference is obtained at the end of the process and is the output variable.

Algorithm 4 Pseudo-code for centroid based anchor mobility control

Input: Maximum number of permissible anchor points (t_{max}), Number of nodes (N_s), Threshold level in signal-to-noise ratio (SNR_{th}), Maximum communication range (d_{max}), Received signal strength (RSS), Angle of arrivals of the received signals: $[\theta]$, Neighboring node IDs: $[ID]$, Neighboring node distance: $[d]$, Cluster size (CS) and Receive mode of anchor: mode state (ms) = ‘Set’

Output: Position of the next anchor point: (x_{an}, y_{an}) and Power saving mode of anchor (Idle/Sleep): mode state (ms) = ‘Reset’

```

1: Initialize:  $i, k, t = 1, CS, sum1, sum2, sum3 = 0, SNR_{th}, N_s, ms = \text{‘Set’}, t_{max}$ 
2:  $x_a \leftarrow 1000 * rand(); y_a \leftarrow 1000 * rand()$ 
3: while ( $t \leq t_{max}$ ) do
4:   while ( $i \leq N_s$ ) do
5:     if ( $RSS[i] \geq SNR_{th}$ ) then
6:        $CS \leftarrow CS + 1; ID[CS] \leftarrow ID[i]; d[CS] \leftarrow RSS[i]; \theta[CS] \leftarrow \theta[i];$ 
7:        $w[CS] \leftarrow \frac{d[CS]}{d_{max}}; \theta'[CS] \leftarrow \pi + \theta[i]$ 
8:     end if
9:      $i \leftarrow i + 1$ 
10:  end while
11:  if ( $CS \neq 0$ ) then
12:    while ( $k \leq CS$ ) do
13:       $sum1 \leftarrow sum1 + w[k] * (d_{max} - d[k]); sum2 \leftarrow sum2 + w[k]$ 
14:       $sum3 \leftarrow sum3 + w[k] * \frac{(\theta[k] + \theta'[k])}{2}; k \leftarrow k + 1$ 
15:    end while
16:     $d_a \leftarrow \frac{sum1}{sum2}; \phi \leftarrow \frac{sum3}{sum2}$ 
17:     $x_{an} \leftarrow x_a + d_a * \cos \phi; y_{an} \leftarrow y_a + d_a * \sin \phi$ 
18:  else
19:     $x_{an} \leftarrow 1000 * rand(); y_{an} \leftarrow 1000 * rand()$ 
20:  end if
21:   $x_a \leftarrow x_{an}; y_a \leftarrow y_{an}; t \leftarrow t + 1$ 
22: end while
23:  $ms \leftarrow \text{Reset}$ 
24: return  $ms, x_{an}, y_{an}$ 

```

3.4.2 Fuzzy-based strategy

In this case, the displacement (d_a) and the direction (ϕ) are obtained on a fuzzy inference system. These mobility parameters are evaluated based on the cluster size and the position of the highest priority node (having the lowest RSS value) in a cluster. The fuzzy controller takes the cluster size (CS), distance (d_p), and direction (θ_p) of the priority node as linguistic variables for crisp inputs to obtain the displacement (d_a) and direction (ϕ) of the new anchor reference as linguistic variables for crisp outputs (as shown in Fig. 3.7). Inputs and outputs are defined with five distinct fuzzy sets: *very small* (VS), *small* (S), *medium* (M), *large* (L) and *very large* (VL). In this work, triangular membership functions are used to represent fuzzy sets like *small* (S), *medium* (M) and *large* (L) and trapezoidal membership functions are used to represent fuzzy sets specifying *very small* (VS) and *very large* (VL). Fig. 3.8 illustrates the membership functions. Fuzzy rules are formulated by combining various fuzzy sets of inputs and outputs and expressed as a collection of **if-and-then** statements [104]. For example, **if** the CS is VL **and** the d_p is VL **then** the d_a is VS . Moreover, **if** the CS is VL **and** the θ_p is VL **then** the ϕ is S . Thus, it produces a total of $5 \times 5 = 25$ (twenty five) fuzzy rules for the individual output variables. This rule base data are given in Tables 3.3 and 3.4. For defuzzification, the Center of Gravity (CoG) is estimated over a number of sample points by aggregating the output membership functions as follows.

$$CoG = \frac{\sum n\rho(n)}{\sum \rho(n)} \quad (3.12)$$

where, $\rho(n)$ is the degree of membership function at the fuzzy point n .

However, the control surface describes the dynamics of the fuzzy controller. The control surface obtained with the simulation on the MATLAB fuzzy logic toolbox is shown in Fig. 3.9. It provides a unified decision of the proposed fuzzy logic controller named *anchormobility.fis*. The MATLAB commands to achieve this are as follows.

```
FIS = readfis('anchormobility.fis');
out = evalfis([CS dp thetap], FIS);
da = out(1); phi = out(2);
```

The displacement of the anchor changes according to the difference value of ($d_{max} - d_p$) because the priority node has the greatest communication range among all. Thus, the anchor can move in a smaller displacement when the cluster size becomes moderately large, and the distance to the priority node remains closer to d_{max} and vice-versa. Likewise, the anchor is always driven with a virtual pulling force towards a direction perpendicular to the line passing through its center at an angle (θ_p) and joining the priority node. This force goes upwards if the slope of the line becomes negative and vice-versa. Algorithm 5 explains the fuzzy-based mobility control strategy for the anchor. It requires several data (e.g., RSS, AoA, and cluster size, etc.) from a cluster to start the process and hence, these are considered as the input variables. The new anchor reference is obtained at the end of the process and is the output variable.

3.5 Performance evaluation

The localization system must operate faster to become energy-efficient. To become robust, it must also produce the least computation error. For this reason, the proposed system is based on the relay of only two beacon messages from two distinct anchor points, by which the nodes compute their positions autonomously using simple mathematical formulas. The use of a smart antenna always ensures the least probability of error in the distance/angle estimates.

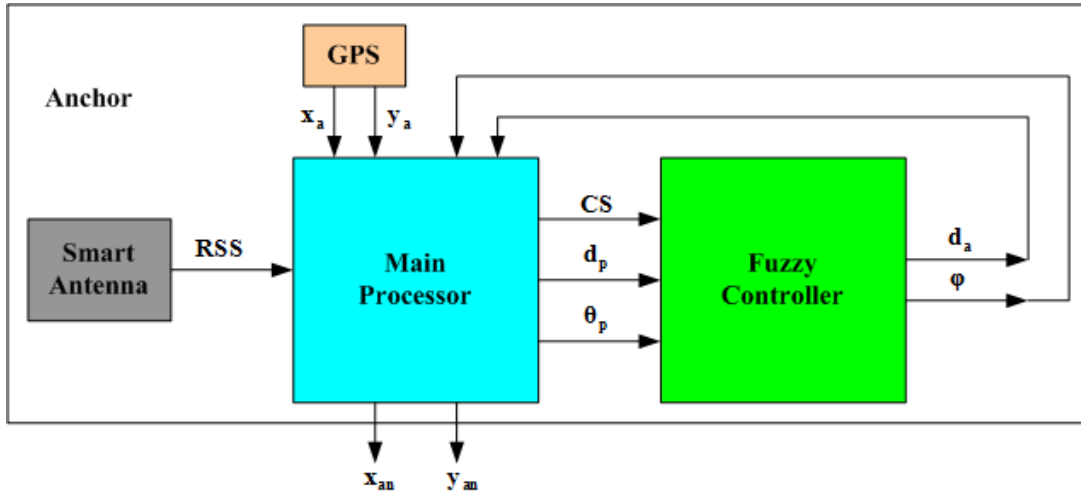


Fig. 3.7. Data flow among various components of anchor

 Table 3.3: Rule base for anchor displacement (d_a)

Fuzzy sets	Distance (d_p)				
Cluster size (CS)	VS	S	M	L	VL
VS	VL	L	M	S	VS
S	L	M	M	S	VS
M	M	S	S	VS	VS
L	S	S	VS	VS	VS
VL	S	S	VS	VS	VS

 Table 3.4: Rule base for anchor direction (ϕ)

Fuzzy sets	Direction (θ_p)				
Cluster size (CS)	VS	S	M	L	VL
VS	VL	VL	VS	M	L
S	VL	VL	VS	M	L
M	M	M	L	VL	S
L	M	M	L	VL	S
VL	M	M	L	VL	S

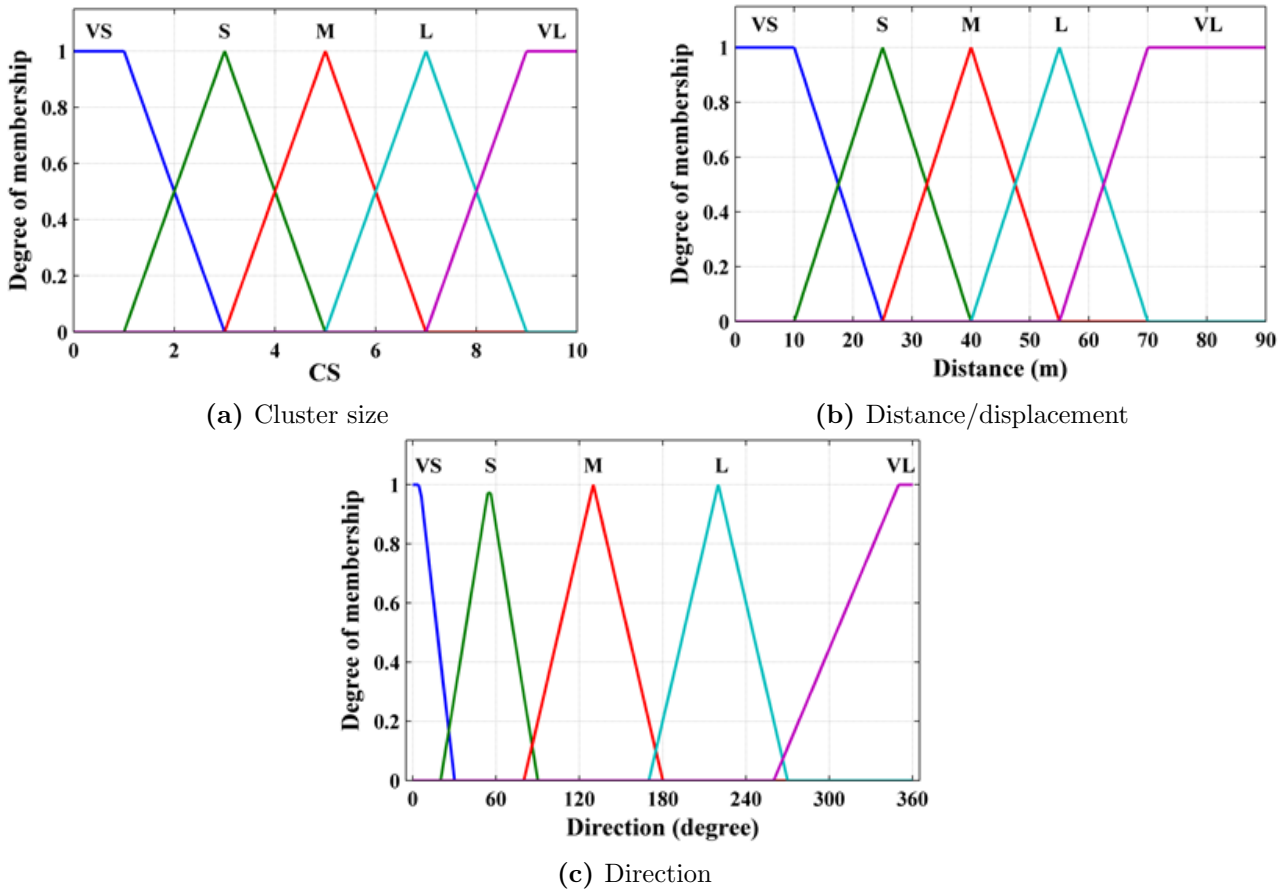


Fig. 3.8. Input/output membership functions

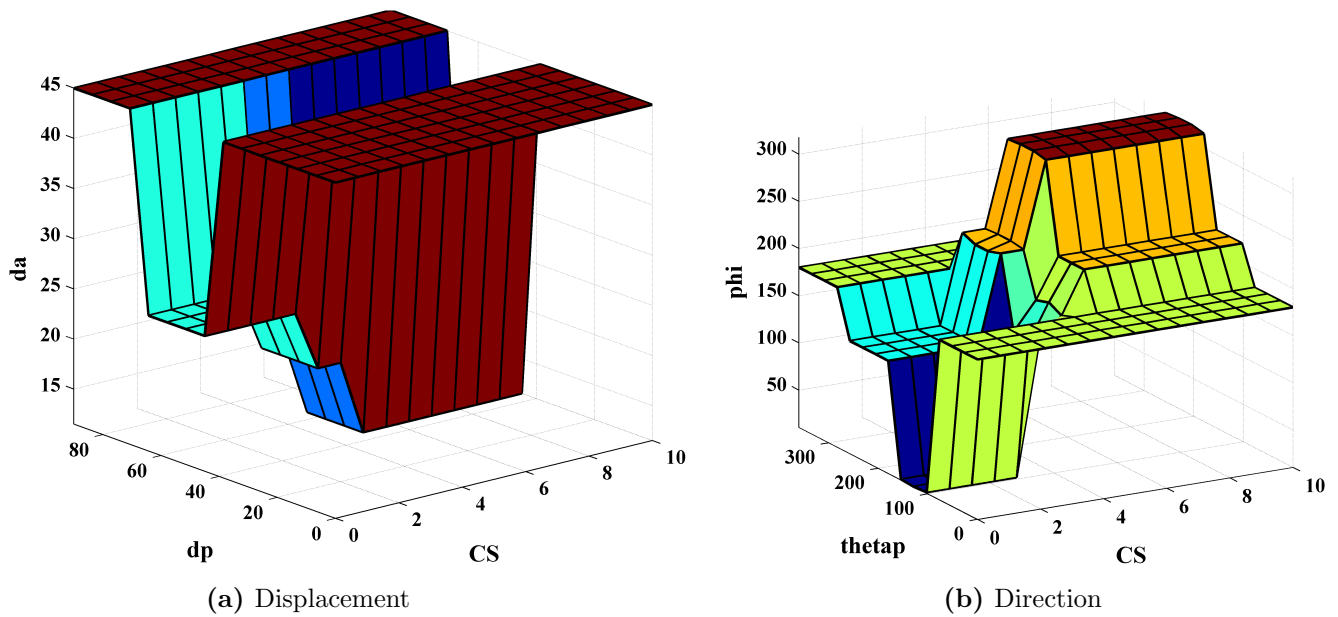


Fig. 3.9. Control surfaces of the output variables

Algorithm 5 Pseudo-code for fuzzy based anchor mobility control

Input: Maximum number of permissible anchor points (t_{max}), Number of nodes (N_s), Threshold level in signal-to-noise ratio (SNR_{th}), Received signal strength (RSS), Angle of arrivals of the received signals: $[\theta]$, Neighboring node IDs: $[ID]$, Neighboring node distance: $[d]$, Cluster size (CS) and Receive mode of anchor: mode state (ms) = ‘Set’

Output: Position of the next anchor point: (x_{an}, y_{an}) and Power saving mode of anchor (Idle/Sleep): mode state (ms) = ‘Reset’

```

1: Initialize:  $i, t = 1, CS = 0, SNR_{th}, N_s, ms = \text{'Set'}, t_{max}$ 
2:  $FIS \leftarrow readfis('anchormobility.fis')$ 
3:  $x_a \leftarrow 1000 * rand(); y_a \leftarrow 1000 * rand()$ 
4: while ( $t \leq t_{max}$ ) do
5:   while ( $i \leq N_s$ ) do
6:     if ( $RSS[i] \geq SNR_{th}$ ) then
7:        $CS \leftarrow CS + 1; ID[CS] \leftarrow ID[i]; d[CS] \leftarrow RSS[i]; \theta[CS] \leftarrow \theta[i]$ 
8:     end if
9:      $i \leftarrow i + 1$ 
10:  end while
11:  if ( $CS \neq 0$ ) then
12:     $[d_p, Index] \leftarrow max\{d\}; \theta_p \leftarrow \theta[Index]$ 
13:     $[d_a, \phi] \leftarrow evalfis([CS, d_p, \theta_p], FIS)$ 
14:     $x_{an} \leftarrow x_a + d_a * \cos \phi; y_{an} \leftarrow y_a + d_a * \sin \phi$ 
15:  else
16:     $x_{an} \leftarrow 1000 * rand(); y_{an} \leftarrow 1000 * rand()$ 
17:  end if
18:   $x_a \leftarrow x_{an}; y_a \leftarrow y_{an}; t \leftarrow t + 1$ 
19: end while
20:  $ms \leftarrow Reset$ 
21: return  $ms, x_{an}, y_{an}$ 

```

Also, considering the mean of the estimated positions ensures a better result with MMSE (Minimum Mean Square Error). Thus, the power dissipation will be reduced when a node obtains beacon messages from two successive anchor points (because the node must stay in active/receive mode until it gets two beacon messages). The proposed mobility control strategies aim to minimize the time interval between two beacon messages. The performance of the proposed system is evaluated under several benchmarks maintaining an analogy with a real-time node deployment scenario. In this section, we have discussed the environment of the simulations, the metrics used and the results obtained.

3.5.1 Performance metrics

The performance of a localization system often depends on the types of mobility control strategies used for the anchor in WSNs. Therefore, to verify the competence of these two new mobility control strategies with respect to the proposed localization system, we considered three metrics [33, 103], such as

- *Localization error* (ε): It is the difference obtained between the estimated position (x', y') and the actual position (x, y) for a node. For the k -th node, it is expressed as follows.

$$\varepsilon_k = \sqrt{(x_k - x'_k)^2 + (y_k - y'_k)^2} \quad (3.13)$$

Further, the average localization error in the network is expressed as follows.

$$\varepsilon_{av} = \frac{1}{N_s} \sum_{l=1}^{N_s} \varepsilon_l \quad (3.14)$$

where, N_s is the total number of nodes in the WSNs.

- *Energy consumption* (ξ): It is the amount of energy consumed by a node in transmitting and receiving data packets during the localization process. For the k -th node, it is expressed as follows.

$$\xi_k = E_{total} \quad (3.15)$$

Further, the average energy consumption in the network is written as follows.

$$\xi_{av} = \frac{1}{N_s} \sum_{l=1}^{N_s} \xi_l \quad (3.16)$$

- *Localization time* (τ'): It is the time taken to complete the localization process in any node. For node k , it is evaluated as follows.

$$\tau'_k = t'_k T \quad (3.17)$$

where, t'_k is the total number of anchor points required for the localization. Likewise, the average localization time is the time taken to complete the entire localization process in the network. Thus, it depends on the maximum number of anchor points used in the localization process and is expressed as follows.

$$\tau'_{av} = \max \{ \tau'_1, \dots, \tau'_{N_s} \} \quad (3.18)$$

Table 3.5: Simulation parameters

Parameters	Values
Network size	1000 m \times 1000 m
Number of nodes	100
Number of anchor	1
Maximum communication range	90 m
Broadcast signal interval	1 sec
Transmission power	165 mW
Receiving power	46.5 mW
SNR threshold level	-40 dBm

3.5.2 Simulation environments

A set of simulations for these metrics is performed using the MATLAB software package (version 14). We generated several offline data on PC (Personal Computer), keeping the analogy with the real-time WSN scenario. As a network architecture, a total of 100 nodes randomly deployed over a 2-D field of 1000 m \times 1000 m are considered. The nodes are supposed to be scattered maintaining a reasonable distance from each other to increase network coverage and avoid interference between them. The anchor is also assumed to move along prescribed trajectories on the network and relay necessary data packets to nodes via private links according to their priority indices in a cluster. For radio propagation in free space, a log-normal shadowing model is assumed. Simulation parameters are chosen keeping compatibility with common wireless narrowband transmission systems such as WPAN IEEE 802.15.4 [100]. These are summarized in Table 3.4.

3.5.3 Simulation results

In order to verify the robustness of the proposed localization system, several simulations are performed for the metrics mentioned above. A total of 30 runs of the program are considered in each case to justify the results. The results are compared to the conventional random anchor mobility model. Fig. 3.10(a), Fig. 3.10(b) and Fig. 3.10(c) respectively illustrate the localization error, energy consumption and localization time for each node obtained with different anchor mobility strategies. Moreover, Fig. 3.11(a), Fig. 3.11(b) and Fig. 3.11(c) respectively show the Empirical Cumulative Distribution Functions (ECDF) for an average estimate of these results. It is found that the least error is achieved with a random model, while the centroid-based strategy is more efficient in terms of energy consumption and localization time. However, the fuzzy-based strategy works the same as the random model.

A higher degree of convergence in the localization process can often be achieved with the fewest anchor points, forming a larger cluster size each time. The convergence speed of the localization process and the cluster size obtained at each anchor point are shown in Fig. 3.12(a) and Fig. 3.12(b), respectively. We notice that the nodes can localize faster in a centroid-based strategy, acquiring a larger cluster size than the other two strategies. The results obtained in the localization process (using random anchor mobility) for the average error and energy consumption are compared with some well-known methods and given in Table 3.5.

3.5.4 Performance analysis

The centroid-based mobility control strategy requires the least number of anchor points for the localization process. Thus, it works with the least localization time and lower energy consumption in WSNs. This happens because intelligent mobility control to keep the anchor on a proper trajectory ensures its frequent visits to regions with maximum number of nodes in each cluster. This would in turn, ensure less localization time and less energy consumption.

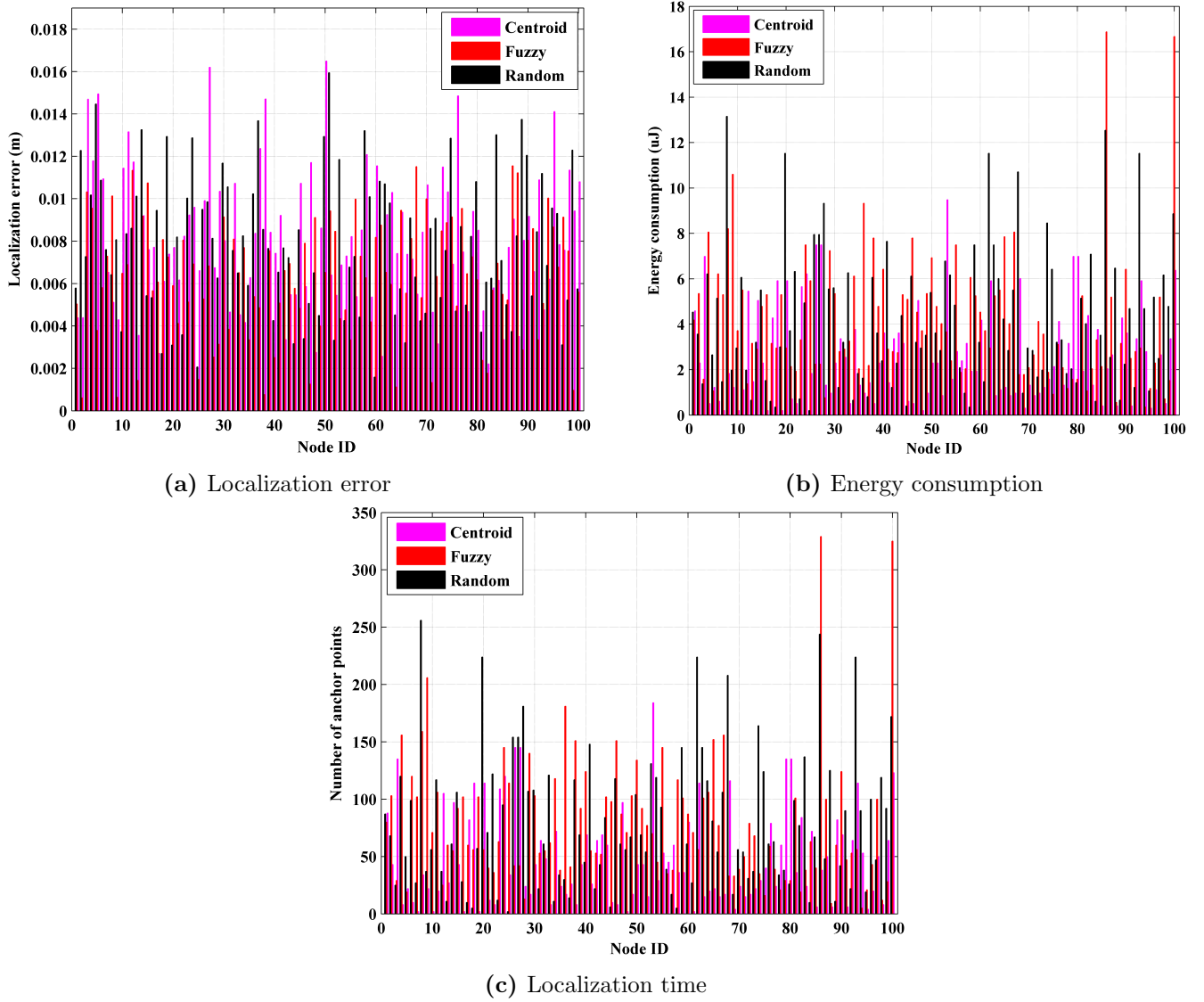


Fig. 3.10. Error, energy consumption and time for localization in each node

Table 3.6: Comparison of error and energy consumption in localization

Localization systems	Average localization error (m)	Average energy consumption (mJ)
Row matching algorithm	0.02	2.97
Range-free localization [30]	5.34	3.01
Three dimensional range-free localization [31]	0.82	2.99
Localization using directional antenna [33]	0.32	0.40
PSO-based localization [103]	0.08	2.98

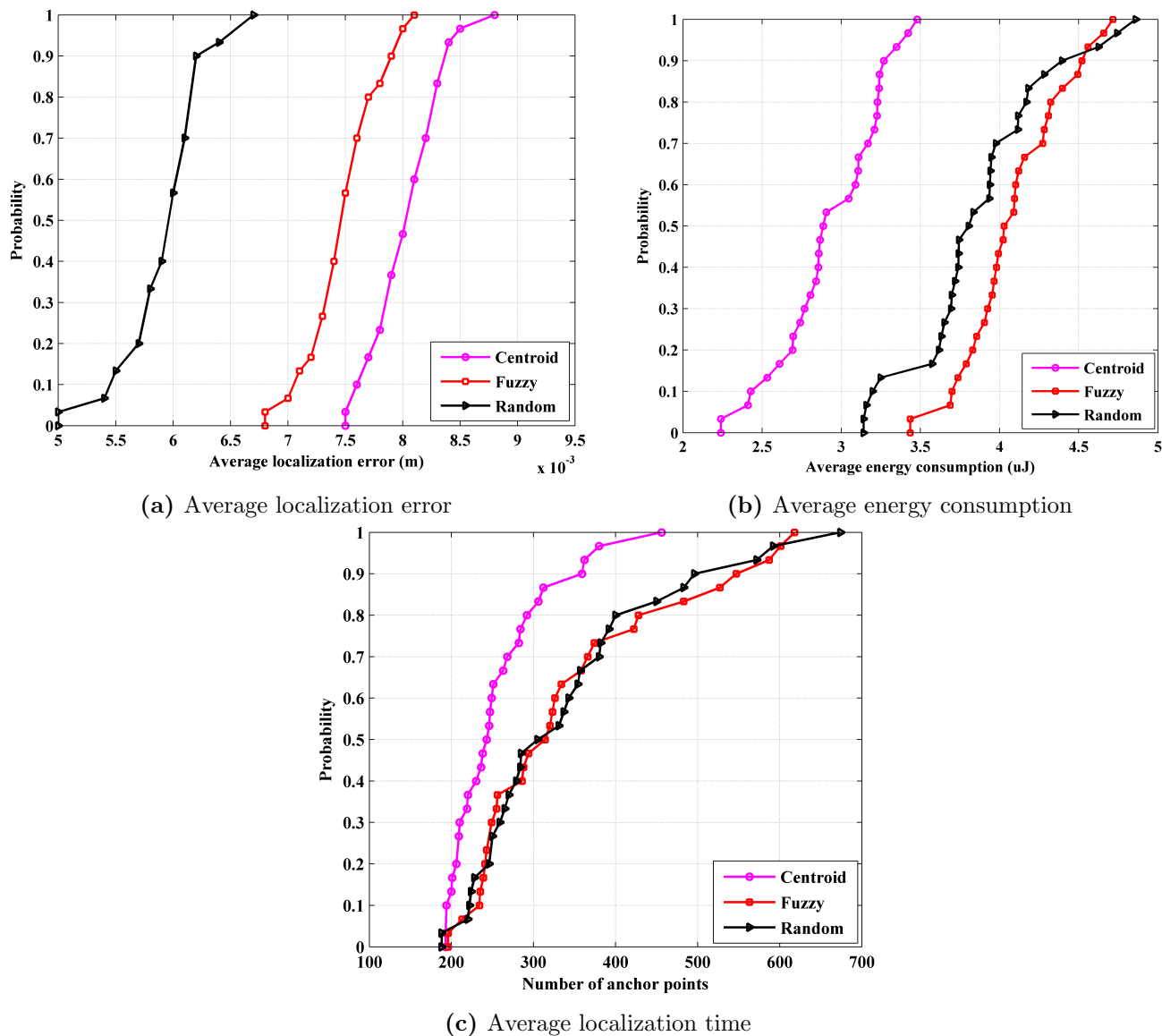


Fig. 3.11. Average error, energy consumption and time for localization in WSNs

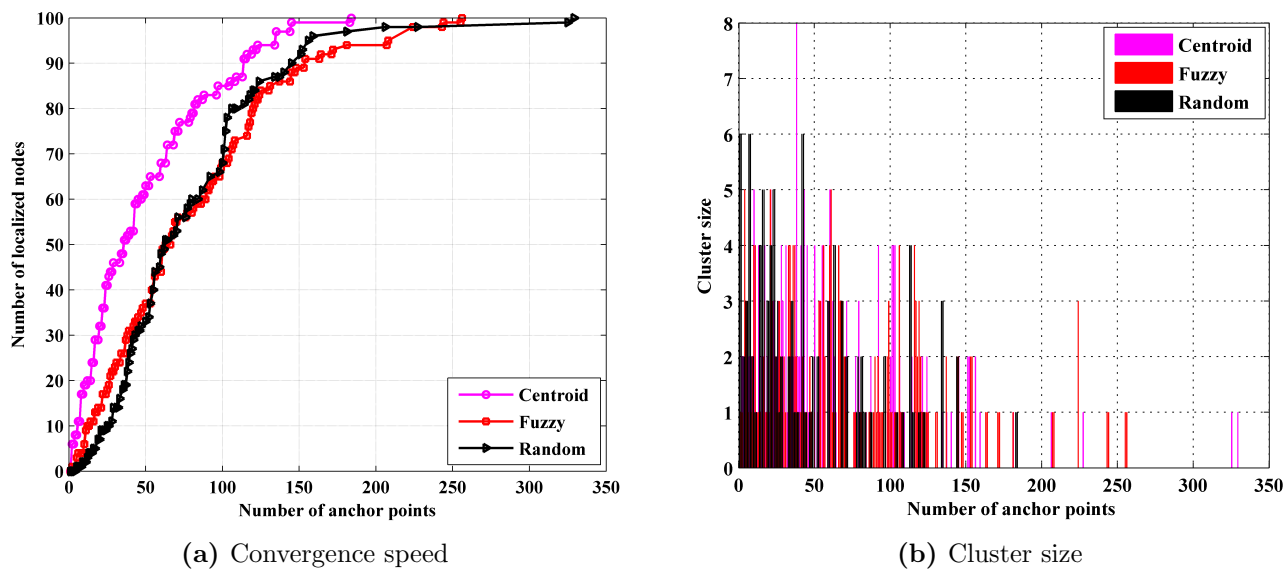


Fig. 3.12. Number of localized and clustering nodes at each anchor point

However, it often produces slightly more localization error compared to the other two strategies. This actually happens due to a minimum displacement of the anchor points between two successive clusters (when the cluster size is large) and the positions estimated with the angle (image) produce the least deviation. On the other hand, the fuzzy-based mobility control strategy works more likely with a random mobility model. They both have a large dynamic range of displacement in consecutive clusters (negligible probability of occurrence of two consecutive anchor points with minimum displacement). Thus, they perform better with lower localization error.

The proposed localization system must satisfy two key aspects such as energy-efficiency and localization accuracy. For its performance analysis, three different mobility control strategies for the anchor are investigated under a common node deployment scenario. The network includes homogeneous nodes and uses the protocols to periodically broadcast signals. Thus, it always estimates an equal amount of energy consumption for the nodes to cover the maximum range. However, switching a node to sleep mode after locating it would make the system energy-efficient. On the other hand, the localization process is delayed when the anchor undergoes several random walks in the field but finds no clusters in its vicinity (although there are still nodes physically awaiting localization in the network). This happens, in practice, when most nodes go into sleep mode and only a few are in active mode (the localization process is almost complete). Thus, it results in unnecessary energy consumption for these nodes. This can be optimized by controlling the mobility of the anchor to keep the nodes in two consecutive clusters and by completing the localization of these nodes in two consecutive anchor points.

3.6 Summary

In this chapter, we have proposed an anchor-based localization system suitable for applications in WSNs. It has the benefits of a single-hop distributed localization algorithm, where each time the anchor moves in a random trajectory and sends beacon messages to nodes on private links so that they can localize themselves autonomously. It is robust enough to use a smart antenna with the anchor and precise distance/angle estimation is possible by eliminating interference. It remains cost-effective to use a single anchor and does not impose much overhead on the nodes in terms of computation and power consumption. In addition, two new mobility control strategies (centroid-based and fuzzy-based) for the anchor are proposed to validate its energy-efficiency and security against attacks. However, the use of a single anchor reduces the convergence speed of the localization process and increases the energy consumption. Thus, improving energy-efficiency and security during the localization process by controlling the mobility of multiple anchors in WSNs is highlighted in the next chapter.

Chapter 4

Secure localization under node capture attacks via anchor mobility control

4.1 Introduction

Nodes remain vulnerable to malicious attacks in hostile environments. An adversarial attack capturing a few benevolent nodes and creating multiple duplicate identities makes the localization process very tricky in WSNs. Incorrect position estimates often lead to a wrong decision regarding the monitored event. Thus, a localization system should be as secure and robust as possible to prevent misleading information from malicious nodes. To achieve this, the confidentiality of localization data, the authenticity of node identities, and the integrity of data packets must also be preserved. Although several research proposals have been developed so far to address these security issues, existing systems often show lower success rates in anomaly detection under varying network conditions. Also, the complexity of these systems results in high energy consumption and requires more expensive equipment, which restricts their applications to network infrastructures with limited resources. From this point of view, the integration of a smart antenna and an expert in anchor mobility control rules offers significant advantages as discussed in the previous chapter. The smart antenna ensures the integrity and confidentiality of data packets relayed over stable and private point-to-point links between the nodes and the anchor. This ensures the least probability of errors in the localization process. Likewise, fuzzy logic-based systems offer a feasible solution to overcome imprecision and uncertainty in complex localization problems using well-defined mathematical frameworks [105]. Thus, the localization accuracy is significantly improved when a fuzzy logic controller is used to explore the new anchor trajectory, separating the captured nodes.

In this chapter, we have proposed a secure localization system that mitigates such attacks using both centroid (deterministic) and fuzzy-based anchor mobility control strategies. Fig. 4.1 shows the trajectories of the anchor in the WSNs. The movement of anchor is limited on the network to ensure identical node identities between two successive clusters. The nodes are treated as malicious to detect any discrepancy with their IDs during this period. Malicious nodes are filtered through multiple iterations until the localization process is complete. Benevolent nodes locate themselves autonomously after receiving two beacon messages and enter power-saving mode. It speeds up the localization process by reducing power consumption at each node.

The rest of this chapter is organized as follows. In Section 4.2, an overview of attack scenarios is given. In Section 4.3, the proposed security strategy is discussed. In Section 4.4, the methodology for implementing the proposed localization system is explained. Simulation results and performance analysis are presented in Section 4.5. Finally, Section 4.6 summarizes the limitations and possible modifications of this system in the future.

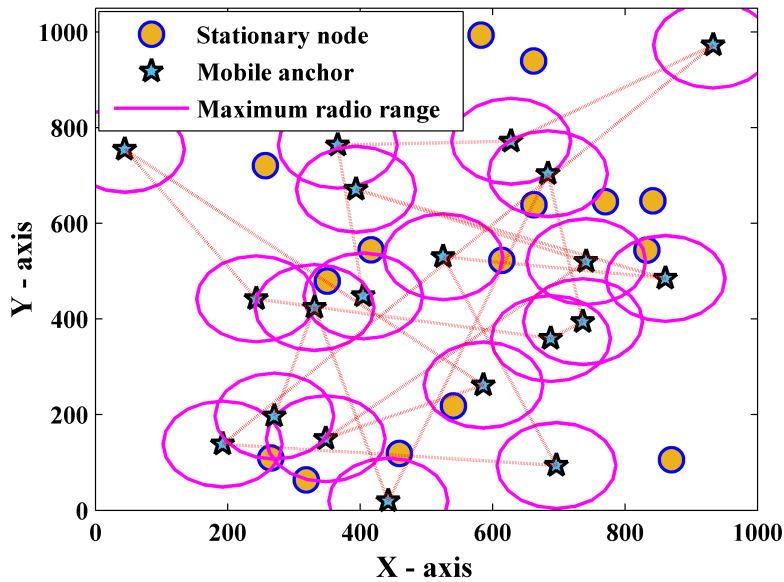


Fig. 4.1. Anchor trajectories in WSN

4.2 Attack scenarios

The purpose of the attacks is to disrupt the plan/decision made at the base station. By exploiting the captured nodes, it is possible to make the localization of benevolent nodes erroneous, either by relaying erroneous localization data or by replaying messages under the pretense of a legitimate node. However, attacks like Sybil, Replay and Wormhole, etc. are now emerging as the main security issues for localization systems. Since nodes remain exposed to eavesdropping threats in a wireless medium, they are captured and controlled by establishing passive links. In this work, we considered that some malicious nodes participate in the localization process by compromising a few benevolent nodes. Such attacks damage the functionality of localization systems generally in two ways, such as

- Sybil attack (captures some legitimate nodes to interfere with the localization process), and
- Replay attack (tampers with localization data to make estimated positions incorrect).

However, compromising nodes is the most fundamental attack that leads to other types of attacks in a network. This significantly degrades the overall performance of a localization system. Fig. 4.2 illustrates the attack scenario for this work.

In WSNs, Sybil attack (also known as node identity attack) occurs in different forms, such as

- *impersonation* where the malicious node (N4) silently repeats data packets relayed between the anchor (A) and a benevolent node (N6 or N7) which are not within direct communication range. This would mislead the anchor if it assumed node N6 or N7 is in the current cluster. Likewise, malicious node (N4) also invalidates a benevolent node (N3 or N5) by stealing all authentication credentials (e.g., cryptographic keys, passwords, etc.) and continuously updating itself with them. This would trick the anchor into accepting the non-existence of captured node identities (only N4 communicates with the anchor as a valid node, except for N3 and N5) within a cluster.
- *replication* where the malicious node (N4) produces replicas of the captured nodes (N3, N5, N6, and N7) by extracting their authentication/encryption keys. The anchor would consider these replicas to be legitimate nodes because they have similar IDs.

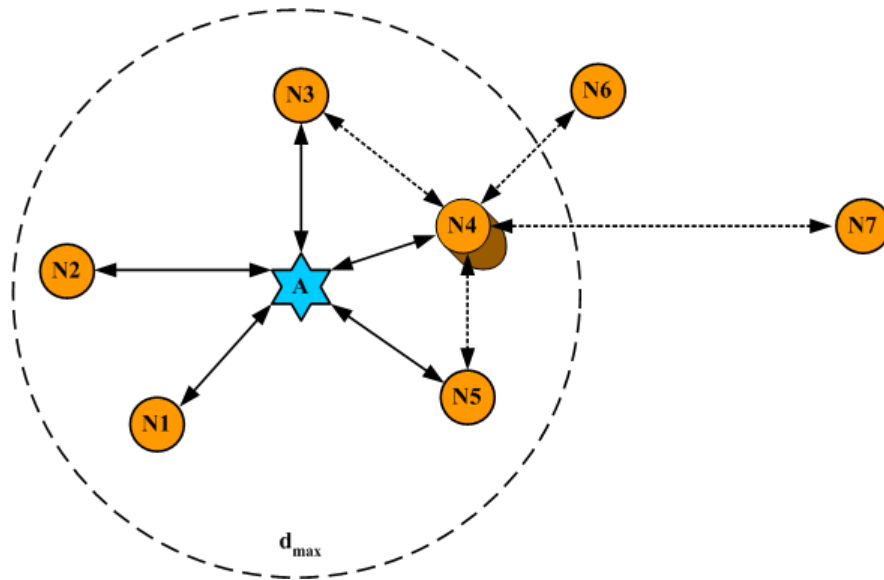


Fig. 4.2. Attack scenarios

Thus, the anchor would assume the presence of duplicate identities (if N3, N4, and N5 communicate with the anchor) in the current cluster.

- *fabrication* where each time the malicious node (N4) fabricates a new identity for each of the captured nodes (N3, N5, N6, and N7) and thus, masquerading themselves with several identities, they cause an ambiguity on the anchor between two subsequent clusters. Therefore, the localization process becomes very tricky when the captured nodes directly inject false information to communicate with the benevolent nodes.

On the other hand, Replay attack (also known as location information attack) occurs in different ways, such as

- *forgery* where the malicious node (N4) sends obsolete information causing congestion in the transmission of information between the anchor (A) and the captured nodes (N3, N5, N6, and N7).
- *alteration* where the malicious node (N4) modifies the information exchanged (e.g., reference coordinates, time stamps, transmission power or number of hops, etc.) between the anchor and the captured nodes (N3, N5, N6, and N7), and
- *interference* where malicious node (N4) interferes with signal measurements between anchor (A) and captured nodes (N3, N5, N6 and N7) by placing obstacles/magnets to extend the transmission time (ToA) in packet delivery, change angle of arrival (AoA) or weaken received signal strength (RSS). As a result, the captured nodes appear to be near/far from their actual locations, and thus the anchor includes/excludes them in a specific cluster.

Benevolent nodes also produce inaccurate position estimates using this outdated, altered, or disrupted information.

4.3 Security frameworks

Some malicious nodes still exist in the network and participate in the localization process. Thus, the important aspect of the security framework is to isolate benevolent nodes from malicious nodes. In this work, this is accomplished by checking the behaviors of neighboring nodes at two consecutive anchor points. Therefore, nodes are considered benevolent if they exist in two successive clusters (no discrepancy detected between node IDs). Otherwise, they remain suspicious.

However, suspicious nodes also include some benevolent nodes in the network. Thus, malicious nodes basically need to be filtered out by fine-tuning these suspicious nodes iteratively. Algorithm 6 explains the node discrimination method of this work. It uses cluster data (e.g., neighboring node identities, cluster sizes, etc.) to find the identities of suspicious and benevolent nodes. Accordingly, they become the input and output variables, respectively.

4.3.1 Suspicious node identification

In the attack scenario illustrated in Fig. 4.2, the malicious node (N4) appears with the identities of the captured node (N3 or N5) via replication/fabrication depending on the Sybil attack. This would produce duplicate identities for captured node (N3 or N5) within the cluster. Similarly, malicious node (N4) includes the identities of captured node (N6 or N7) within the cluster via impersonation/fabrication, although they do not physically belong to it. Nodes with duplicate identities are considered suspicious and hence blocked by the anchor at that particular stage. However, node identities via impersonation/fabrication seem to mislead the anchor. But they are also blocked by the adaptive beamforming of the smart antenna. Since the antenna pattern provides radio links for the erroneously estimated locations, it drops beacon messages relayed to those nodes (as the malicious node receives them), and the actual nodes remain active in that case. Moreover, the anchor travels several paths during the localization process. It also recognizes the identities of a few legitimate nodes at their physical locations that appeared earlier through impersonation/fabrication. This would again make the anchor ambiguous to achieve duplicate identities. They are also treated as suspicious.

On the other hand, the malicious node (N4) deceives the anchor by including the identities of some captured nodes (N6 or N7) to a particular cluster although they are outside a cluster domain. Likewise, it also misleads the anchor by excluding the identities of some captured nodes (N3 or N5) in that particular cluster although they remain inside a cluster domain. Fig. 4.2 shows this attack scenario. It occurs in the Replay attack when the malicious node (N4) alters the information exchanged or interferes with the signal measurements between the anchor and the captured nodes. The anchor would assume that captured node (N6 or N7) inside the cluster has increased transmit power. Likewise, the anchor would consider the existence of the captured node (N3 or N5) outside of a cluster domain having a delayed transmission time. If such changes in signal transmission remain consistent across the network, adaptive antenna pattern synthesis would be more effective in counteracting them. Otherwise, any inconsistency can often lead to a captured node not existing in the next cluster when it has already appeared. It is also suspicious and therefore blocked by the anchor at this particular stage.

Additionally, reprogramming captured nodes with malicious code could turn them into fraudulent nodes that would still masquerade their legitimate identities on the network. At this point, it would be more difficult to identify their original identities. Thus, the anchor should maintain the individual database for localized nodes and suspicious nodes. By periodically updating and checking these databases, the anchor satisfies the termination condition (if the sum of the two data equals the total number of nodes deployed in WSN) for its mobility control. The anchor scans the database of localized node and discovers any discrepancies for two possible cases, such as

- if a node ID is found localized two or more times, and
- if a node ID is found not to be localized at all during the entire process.

Fraudulent nodes can cause this situation, and the anchor should treat them as suspicious. Accordingly, it must update the database of suspicious nodes. Fig. 4.3 shows the process of discrimination between suspicious nodes and benevolent nodes.

Algorithm 6 Pseudo code for discriminating suspicious and benevolent nodes

Input: Maximum number of permissible anchor points (t_{max}), Cluster size: $[CS]$, Neighboring node IDs: $[ID]$, Updated clustering node IDs: $[nID]$, Updated cluster size: $[nCS]$ and Receive mode for anchor: mode state (ms) = ‘Set’

Output: Benevolent node IDs: $[bID]$, Suspicious node IDs: $[sID]$ and Idle mode for anchor: mode state (ms) = ‘Reset’

```

1: Initialize:  $i, l, k, count, t, idx_1, idx_2, idx_3 = 1, idx_4 = 0, j = i + 1, m = l, Flag, ms = \text{‘Set’}, t_{max}$ 
2: while ( $t \leq t_{max}$ ) do
3:   while ( $i \leq CS$ ) do
4:     while ( $j \leq CS$ ) do
5:       if ( $ID[i] = ID[j]$ ) then
6:          $count \leftarrow count + 1; Repeat[j] \leftarrow Flag$ 
7:       end if
8:        $j \leftarrow j + 1$ 
9:     end while
10:    if ( $Repeat[i] \neq Flag$ ) then
11:       $Repeat[i] \leftarrow count$ 
12:      if ( $count \geq 2$ ) then
13:         $sID[idx_1] \leftarrow ID[i]; idx_1 \leftarrow idx_1 + 1$ 
14:      else
15:         $nID[idx_2] \leftarrow ID[i]; idx_2 \leftarrow idx_2 + 1$ 
16:      end if
17:    end if
18:     $i \leftarrow i + 1$ 
19:  end while
20:   $nCS \leftarrow idx_2 - 1$ 
21:  if ( $t = 1$ ) then
22:     $fCS \leftarrow nCS; fID \leftarrow nID$ 
23:  else
24:    while ( $k \leq fCS$ ) do
25:      while ( $l \leq nCS$ ) do
26:        if ( $fID[k] = nID[l]$ ) then
27:           $bID[idx_3] \leftarrow fID[k]; idx_3 \leftarrow idx_3 + 1; nCS \leftarrow nCS - 1$ 
28:          while ( $m \leq nCS$ ) do
29:             $nID[m] \leftarrow nID[m + 1]; m \leftarrow m + 1$ 
30:          end while
31:        else
32:           $idx_4 \leftarrow idx_4 + 1$ 
33:        end if
34:         $l \leftarrow l + 1$ 
35:      end while
36:      if ( $idx_4 = nCS$ ) then
37:         $sID[idx_1] \leftarrow fID[k]; idx_1 \leftarrow idx_1 + 1$ 
38:      end if
39:       $idx_4 \leftarrow 0; k \leftarrow k + 1$ 
40:    end while
41:     $idx_2 \leftarrow 1; fCS \leftarrow nCS; fID \leftarrow nID$ 
42:  end if
43:   $t \leftarrow t + 1$ 
44: end while
45:  $ms \leftarrow Reset$ 
46: return  $ms, bID, sID$ 

```

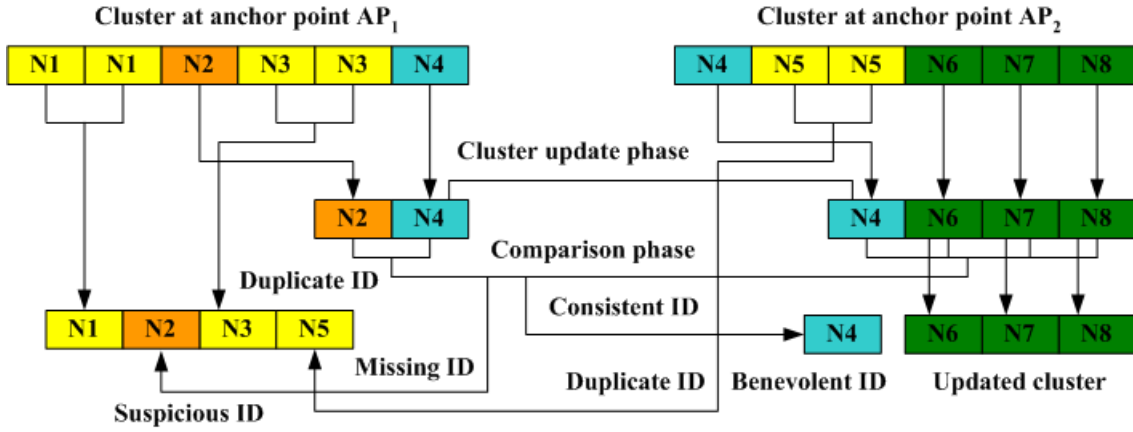


Fig. 4.3. Segregation of benevolent nodes

Suppose a cluster with a cluster size (CS) of six nodes (having an RSS exceeding the prescribed threshold) has formed at the first anchor point. It contains only four distinct node identities (N1, N2, N3, and N4). Duplicate-produced nodes (N1 and N3) are registered as suspicious in the anchor database and removed from this cluster. The cluster size is now updated to have two nodes (N2 and N4). Based on this updated cluster size (nCS), the second anchor point is determined.

Suppose another cluster with a cluster size (CS) of six nodes (having an RSS exceeding the prescribed threshold) is formed at the second anchor point. It also only contains five distinct node identities (N4, N5, N6, N7, and N8). Again, the anchor database is updated to record all suspicious nodes producing duplicates (N5) or disappearing (N2) in this cluster. The common node (N4) between the two consecutive clusters is considered benevolent. It localizes itself autonomously and switches into sleep mode. The anchor database is updated to save the localized nodes. The cluster size is updated to have three nodes (N6, N7, and N8) removing the suspicious node (N5) and the localized node (N4). The anchor moves to the third anchor point based on these new cluster attributes in the same way. This process continues until the termination criterion is met.

Therefore, this work identifies suspicious nodes in the following ways.

Definition 1: any node identity that appears two or more times in a cluster (if multiple malicious nodes capturing the same nodes exist).

Definition 2: any node identity of a cluster that disappears into its next cluster.

Definition 3: any node identity that appears localized two or more times throughout the process.

Definition 4: any node identity that remains missing throughout the process.

4.3.2 Malicious node elimination

The localization process rejects all suspicious nodes which also include a few benevolent nodes. These are called false-positive nodes. Thus, these nodes must be taken into account again for a new iteration. The anchor transfers all suspicious node identities to the base station at the end of any iteration.

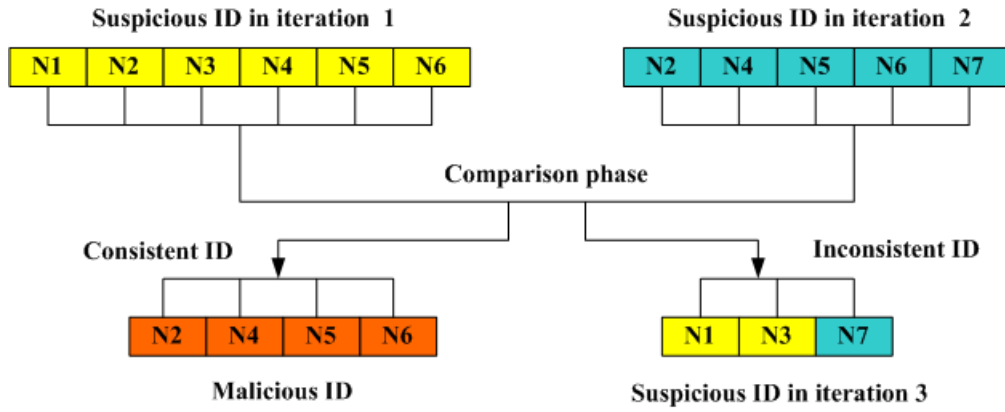


Fig. 4.4. Removal of malicious nodes

The base station broadcasts control signals only to these nodes to switch them to active mode at the next iteration. Since benevolent nodes go into sleep mode after being localized, the neighboring nodes of a malicious node drastically shrink over the iterations. This would, in turn, reduce the probability of being captured and also produce fewer suspicious nodes. However, the malicious nodes remain consistent in the network, and are exposed to the anchor gradually over the iterations.

Fig. 4.4 shows the process of removing malicious nodes. Suppose the anchor database for suspicious nodes contains only six distinct nodes (N1, N2, N3, N4, N5, and N6) at the end of the first iteration. This database is transferred to the base station to activate these nodes during the second iteration. Again, the anchor identifies five suspicious nodes (N2, N4, N5, N6, and N7) existing in its database at the end of the second iteration. Therefore, suspicious nodes (N2, N4, N5, and N6) existing in both iterations are considered malicious. The anchor updates its database by deleting these nodes. Now, it recommends in the third iteration to replicate all existing inconsistent nodes (N1, N3, and N7) in the anchor database and examine them in the same way. This process will continue until a suspicious node exists in the network. The anchor updates the database of suspicious nodes by periodically removing malicious nodes from the second iteration. This iterative process ends when no suspicious nodes exist in the anchor database. Thus, selective pruning of suspicious nodes effectively eliminates malicious nodes. This is defined as follows.

Definition 5: any node identity that is consistent as to be suspicious over two subsequent iterations.

4.4 Localization system

In this work, we proposed a distributed localization system based on the control of the mobility of an anchor. Due to resource constraints, a localization system should always be as simple as possible in WSNs. For this reason, we suggested integrating a mobility controller (centroid/fuzzy-based) and a smart antenna both with the anchor. It makes sure not to impose additional computational overhead on tiny nodes. An intelligent controller (based on deterministic/fuzzy logic) controls the mobility of the anchor in the network. The basic purpose of such a controller is to discover malicious nodes so that they can be blocked before the localization process begins. To achieve this, two consecutive anchor points must keep at least the same neighboring nodes to form individual clusters.

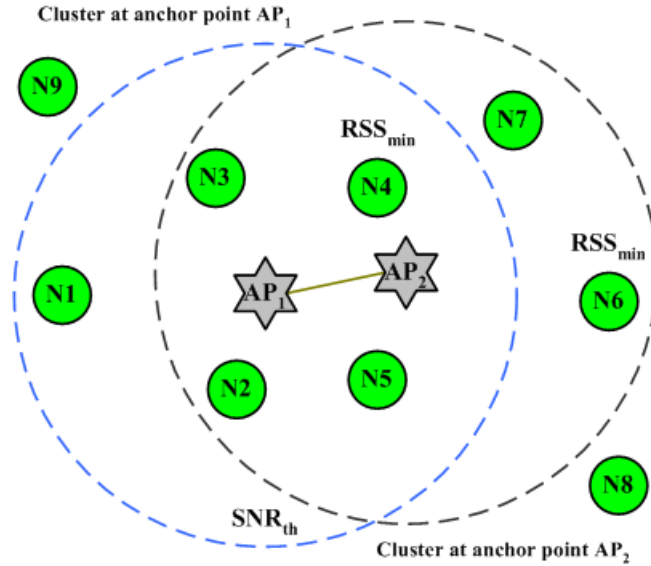


Fig. 4.5. Cluster formation in two successive anchor points

Nodes with discrepancies in IDs between two subsequent clusters are considered suspicious. On the other hand, the benevolent nodes existing in the two clusters localize themselves autonomously by obtaining two beacon messages from two distinct anchor points. Thus, it makes the localization process faster and more energy-efficient.

4.4.1 Mobility controls

As discussed in the previous chapter, the anchor mobility controller (centroid/fuzzy-based) must retain all nodes in the current cluster to exist in the next cluster. It finds the new anchor position based on two factors: the updated cluster size (nCS) and the positions of all nodes in the updated cluster. The priority indices of these nodes are evaluated using equation (3.8). The distance (d_k) and direction (θ_k) of these nodes are estimated using equation (1.4) and equation (1.11), respectively. The displacement (d_a) and direction (ϕ) for the second anchor point are determined using a fuzzy logic controller and a deterministic (centroid) controller, as follows.

- For a fuzzy logic controller, the fuzzy rules are formulated according to the position of the highest priority node (having the lowest RSS) and the updated cluster size. The values of these parameters are obtained from the control surface describing with equation (3.12).
- For a deterministic (centroid) controller, the values of these parameters are evaluated against all nodes in the updated cluster using equation (3.10) and equation (3.11), respectively.

The position of the second anchor point is evaluated using equation (3.9). Fig. 4.5 and Fig. 4.6 show respectively the cluster formation and anchor mobility control strategies in the two successive anchor points. The mobility controller keeps the anchor on such a trajectory which not only explores a better number of neighboring nodes in a cluster but also ensures fast localization by keeping them in the next cluster.

4.4.2 Position computation

The anchor is assumed to periodically receive signals from neighboring nodes whose RSS exceeds a prescribed threshold. Thus, it forms a unique cluster each time with such neighboring nodes at each anchor point. It also checks the cluster in each interval and rejects (suspicious) nodes that create duplicated IDs or remain missing in the next cluster. Each time, it estimates the AoA information of all received signals from the remaining (benevolent) nodes using the smart antenna.

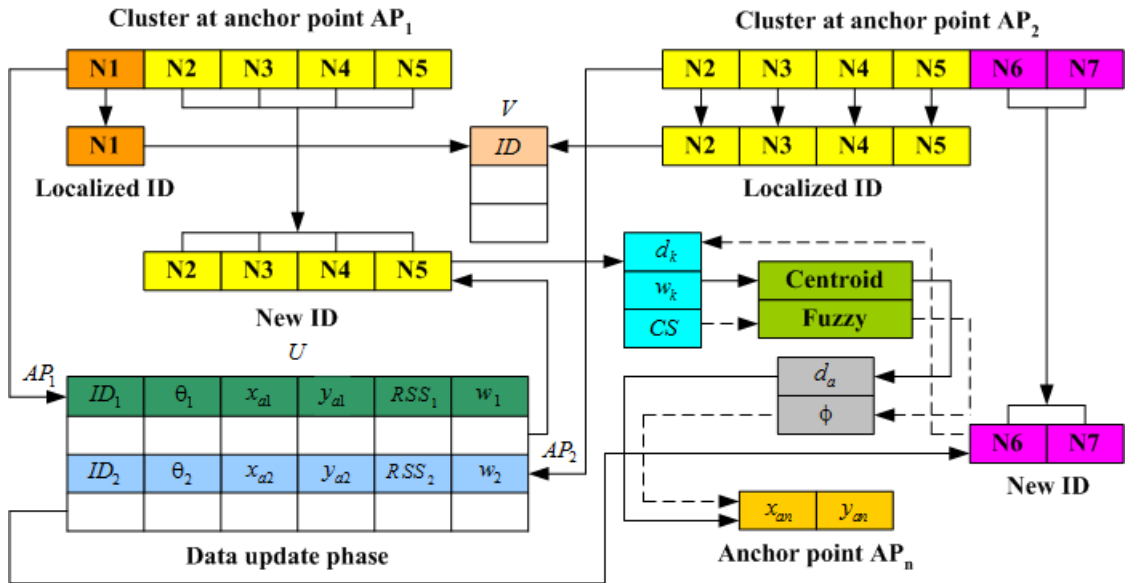


Fig. 4.6. Anchor mobility control strategies for secure localization

In each cluster, the AoA information and the current anchor reference are used to generate beacon messages, and they are serially transferred to the corresponding nodes. A beacon message contains vital information needed to compute the position of a node. Fig. 4.7 shows the format of such a beacon message.

Each time, the anchor makes a direct link to the desired node with an optimized pattern in which the deep nulls point to the directions of the remaining nodes. Such an antenna pattern becomes effective in providing a second layer of protection in the localization process in the event of malicious attacks. It would stay out of any captured nodes while transmitting the beacon message directing the deep nulls and keeping the sidelobe level lower in the pattern. Beam steering and null steering depend on the precise measurement of localization data. Therefore, if a captured node stays in the direction of the main beam, the beacon message may also drop due to receiving incorrect localization information. A smart antenna provides stable radio links through the process of adaptive beamforming. However, a beam pattern with a narrower beamwidth and lower sidelobe level is still desirable for transmitting the beacon message. Algorithm 7 explains the process of generating and transmitting beacon messages. Since beacon message generation and transmission requires several data from a cluster (e.g., updated cluster node identities, AoA, etc.) and antenna design parameters, they are used as input variables. On the other hand, the desired antenna pattern and the beacon message format are considered as output variables.

A benevolent node computes its position by receiving only two beacon messages in two consecutive anchor points. Localization of nodes is done based on AoA information. Therefore, each node stores only two beacon messages within an acceptable time interval (t_{max}). The node enters sleep mode each time it completes the localization process. Algorithm 8 explains the process of accumulating beacons and node localization. Since the position estimation uses the beacon message and the mode switching is controlled based on the operations, they are considered as the input and output parameters. Each node periodically switches to receive mode over an allowable time interval. A counter is activated to validate receipt of the prescribed beacon messages in its memory. It enters idle mode after successful accumulation. Otherwise, it switches to active mode to broadcast the signal. In this work, the triangulation method is used to compute the position of each benevolent node. Thus, a pair of straight-line equations is formulated with the beacon messages stored in each node. The point of intersection between these straight lines is calculated as the position of that particular node. Fig. 4.8 illustrates the triangulation process.

Node ID (bits)	AoA (rad)	Anchor reference (meter)
k	θ_k	$[x_a \ y_a]$

Fig. 4.7. Format of the beacon message

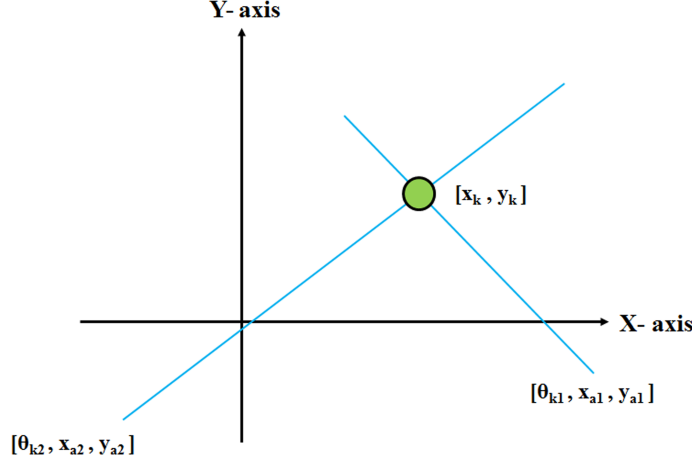


Fig. 4.8. Triangulation process for position computation

For the k -th node, the straight line equations are expressed as follows.

$$\left. \begin{aligned} y_k - y_{a1} &= \tan \theta_{k1} (x_k - x_{a1}) \\ y_k - y_{a2} &= \tan \theta_{k2} (x_k - x_{a2}) \end{aligned} \right\} \quad (4.1)$$

where, θ_{ki} and (x_{ai}, y_{ai}) denote the corresponding AoA information and the position of the anchor in the i -th beacon message ($i = 1, 2$).

However, this can also be represented in matrix form as follows.

$$B_0 x = c_0 \quad (4.2)$$

where,

$$B_0 = \begin{bmatrix} -\tan \theta_{k1} & 1 \\ -\tan \theta_{k2} & 1 \end{bmatrix}$$

$$x = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

$$c_0 = \begin{bmatrix} y_{a1} - \tan \theta_{k1} x_{a1} \\ y_{a2} - \tan \theta_{k2} x_{a2} \end{bmatrix}$$

Thus, the position of the k -th node is computed as follows.

$$x = B_0^{-1} c_0 \quad (4.3)$$

4.4.3 Design methodology

Both the anchor and the node operate in different modes such as *active*, *receive*, *idle*, and *sleep*. Thus, suitable ring counters are needed to generate trigger pulses for mode switching operation. Usually, the transition of modes for a specific action is regulated periodically (T is the period).

Algorithm 7 Pseudo-code for generating and transmitting beacon messages

Input: Maximum number of permissible anchor points (t_{max}), Angle estimated from the received signals (φ), Updated cluster size: $[nCS]$, Updated clustering node IDs: $[nID]$, Anchor reference: $(U[Ptr][3], U[Ptr][4])$, Number of antenna elements ($2N$), Beamwidth between the first nulls ($FNBW$), Sidelobe level (SLL), Depth of nulls (K), Scanning angle (θ) and Idle mode of anchor: mode state (ms) = ‘Reset’

Output: Generated beacon messages: B_m [size], Optimized beam pattern (AF_p), Active mode of the anchor: mode state (ms) = ‘Set’

```

1: Initialize:  $i, j, k, n, t = 1, AF_p = 0, \theta = -90^0, size = 4, K, N, SLL, FNBW, ms = \text{‘Reset’}, t_{max}$ 
2: while ( $t \leq t_{max}$ ) do
3:   while ( $i \leq nCS$ ) do
4:      $\theta_d \leftarrow \varphi[1]$ 
5:     while ( $j \leq size$ ) do
6:        $B_m[j] \leftarrow nID[i]; nID[i] \leftarrow \varphi[1]; \varphi[1] \leftarrow U[Ptr][3]; U[Ptr][3] \leftarrow U[Ptr][4]; j \leftarrow j + 1$ 
7:     end while
8:     while ( $\theta \leq 90^0$ ) do
9:       if ( $|\theta - \theta_d| \leq \frac{FNBW}{2}$ ) then
10:         $AF_p \leftarrow 1$ 
11:       else
12:         $AF_p \leftarrow SLL$ 
13:       end if
14:       while ( $k \leq nCS - 1$ ) do
15:         $\theta_n \leftarrow \varphi[k + 1]$ 
16:        if ( $\theta = \theta_n$ ) then
17:           $AF_p \leftarrow K$ 
18:        end if
19:         $\varphi[k] \leftarrow \varphi[k + 1]; k \leftarrow k + 1$ 
20:       end while
21:       while ( $n \leq N$ ) do
22:         $AF_p \leftarrow AF_p + W[n] * \cos \left[ \left( n - \frac{1}{2} \right) * \pi * (\sin \theta - \sin \theta_d) \right]; n \leftarrow n + 1$ 
23:       end while
24:        $\theta \leftarrow \theta + 1$ 
25:     end while
26:      $\varphi[nCS] \leftarrow \theta_d; i \leftarrow i + 1$ 
27:   end while
28:    $t \leftarrow t + 1$ 
29: end while
30:  $ms \leftarrow Set$ 
31: return  $ms, AF_p, B_m[size]$ 

```

Algorithm 8 Pseudo code to accumulate beacon messages and compute position in the node

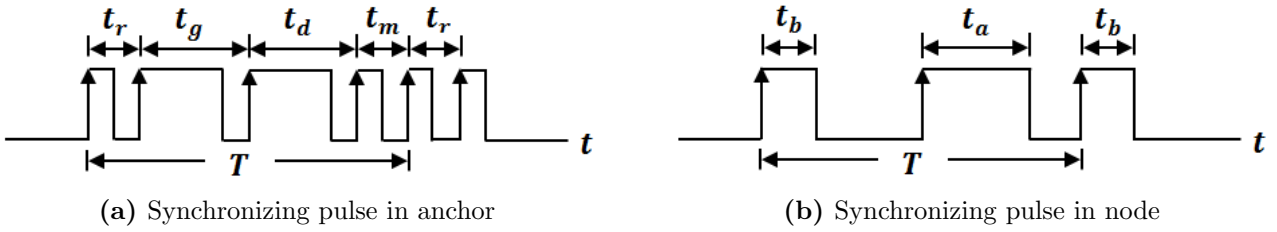
Input: Maximum number of permissible anchor points (t_{max}), Anchor generated beacon messages: B_m [size] and Receive mode of node: mode state (ms) = ‘Set’

Output: Accumulated beacon messages: U' [count] [size], Estimated position of the node: $[x]$ and Power saving mode of node (Idle/Sleep): mode state (ms) = ‘Reset’

```

1: Initialize:  $t = 1, count = 0, ms = \text{'Set'}, t_{max}$ 
2: while ( $t \leq t_{max}$ ) do
3:    $U'[count][size] \leftarrow B_m[size]; count \leftarrow count + 1$ 
4:   if ( $count = 2$ ) then
5:     break
6:   else
7:      $ms \leftarrow Set(Active)$ 
8:   end if
9:    $t \leftarrow t + 1$ 
10: end while
11:  $ms \leftarrow Reset(Idle)$ 
12:  $B_0 \leftarrow \begin{bmatrix} -\tan \theta_{k1} & 1 \\ -\tan \theta_{k2} & 1 \end{bmatrix}; c_0 \leftarrow \begin{bmatrix} y_{a1} - \tan \theta_{k1} * x_{a1} \\ y_{a2} - \tan \theta_{k2} * x_{a2} \end{bmatrix}$ 
13:  $x \leftarrow B_0^{-1} * c_0$ 
14:  $ms \leftarrow Reset(Sleep)$ 
15: return  $ms, x, U'[count][size]$ 

```

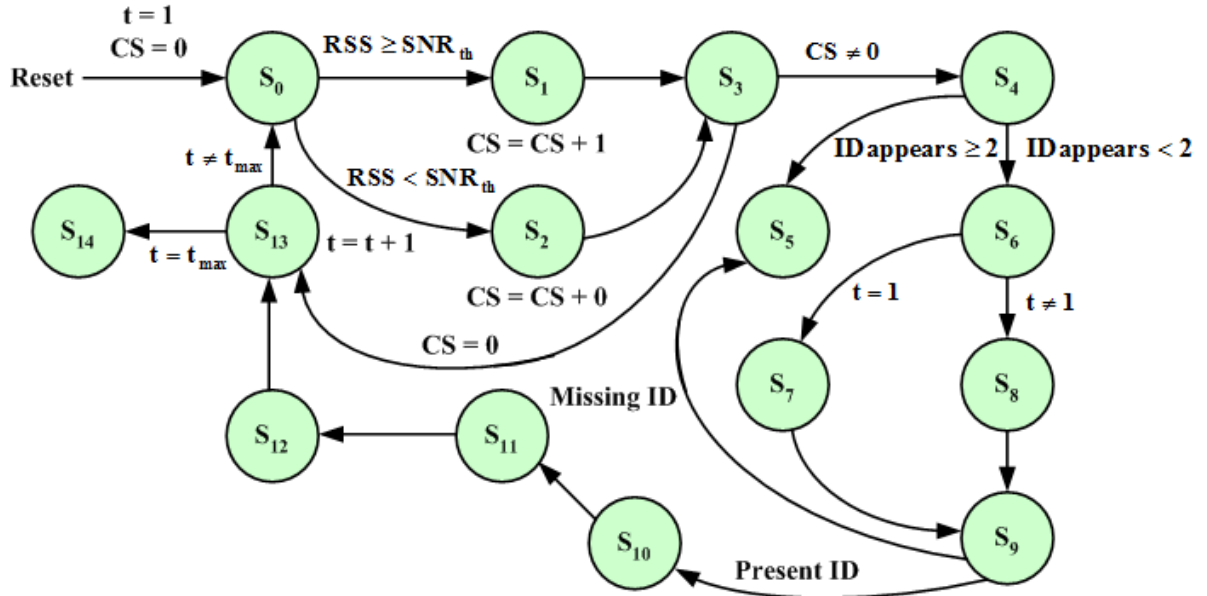

Fig. 4.9. Trigger pulses for mode switching in anchor/node

During the synchronization process, as shown in Fig. 4.9, nodes and anchor are allocated equal time slots in broadcasting (t_b) and receiving signals (t_r) from neighboring nodes, respectively. Likewise, equal time is considered for anchor and nodes in both the distribution (t_d) and accumulation (t_a) of beacon messages, respectively. Also, an allowable time is defined for generating beacon messages (t_g) and anchor movement (t_m).

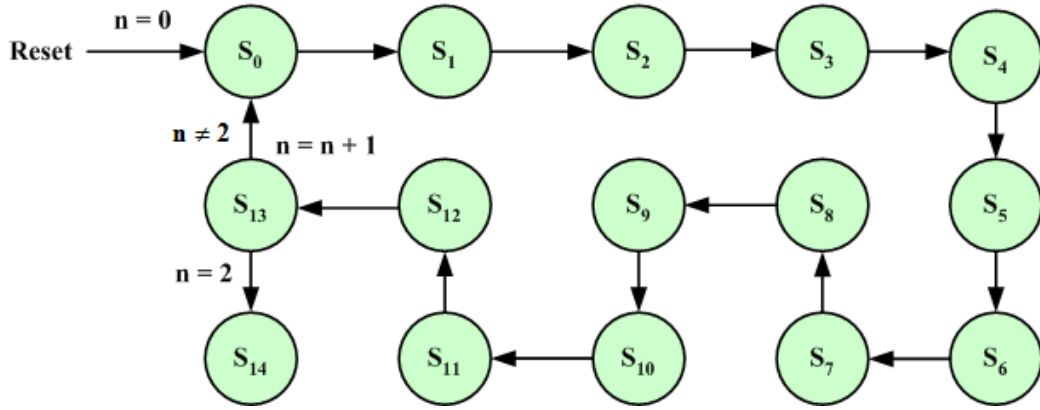
However, the design principle of such a system can be considered analogous to Finite State Machine (FSM) modeling. Thus, a separate FSM model is used to implement the controllers for anchor and node, as shown in Fig. 4.10. The functionality of each state in the anchor and node is described in Table 4.1 and 4.2, respectively.

4.5 Performance evaluation

In this work, the nodes encountered first time in any cluster are kept in the next cluster by using an intelligent control on the mobility of anchor. This strategy helps to identify suspicious nodes in a cluster. Nodes that appeared with duplicate IDs in one cluster or missing in the next cluster are considered suspicious. As a result, they are blocked and only benevolent nodes participate in the localization process. Since suspicious nodes also include some benevolent nodes, they are taken into account again in the next iteration. The proposed system dynamically reduces neighbors of malicious nodes in subsequent iterations.



(a) FSM design for anchor operation



(b) FSM design for node operation

Fig. 4.10. State diagrams illustrating the node/anchor behaviors

Table 4.1: Anchor behaviors in the FSM

State	Anchor		
	Command	Action	Mode
S_0	<i>Reset, CS = 0, t = 1, t \neq t_{max}</i>	Disable servo system, Receive signal	Receive
S_1	$RSS \geq SNR_{th}$	Start counting CS	Idle
S_2	$RSS < SNR_{th}$	Skip counting CS	Idle
S_3	Clock	Total counting CS	Idle
S_4	$CS \neq 0$	Store IDs	Idle
S_5	ID appears ≥ 2 , Missing IDs	Suspicious IDs	Idle
S_6	ID appears < 2	Update IDs and CS	Idle
S_7	$t = 1$	$fID = nID, fCS = nCS$	Idle
S_8	$t \neq 1$	$nID = nID, nCS = nCS$	Idle
S_9	Clock	Check for missing IDs	Idle
S_{10}	Present IDs	AoA estimation	Idle
S_{11}	Clock	Beacon message generation	Idle
S_{12}	Clock	Beacon message transmission	Active
S_{13}	Clock, $CS = 0$	Enable servo system	Idle
S_{14}	$t = t_{max}$	Disable servo system, Wait for new instructions	Sleep

Table 4.2: Node behaviors in the FSM

State	Node		
	Command	Action	Mode
S_0	$Reset, n = 0, n \neq 2$	Broadcast signal	Active
S_1	Clock	Wait for new instructions	Sleep
S_2	Clock	Wait for new instructions	Sleep
S_3	Clock	Wait for new instructions	Sleep
S_4	Clock	Wait for new instructions	Sleep
S_5	Clock	Wait for new instructions	Sleep
S_6	Clock	Wait for new instructions	Sleep
S_7	Clock	Wait for new instructions	Sleep
S_8	Clock	Wait for new instructions	Sleep
S_9	Clock	Wait for new instructions	Sleep
S_{10}	Clock	Wait for new instructions	Sleep
S_{11}	Clock	Wait for new instructions	Sleep
S_{12}	Clock	Beacon message accumulation	Receive
S_{13}	Clock	Update counter	Idle
S_{14}	$n = 2$	Estimate position	Idle

This, in turn, would still ensure easy detection and filtering of malicious nodes, making the localization system secure enough. Its performance is evaluated with several benchmark functions described below.

4.5.1 Performance metrics

In a hostile environment, the localization system must be as robust, secure and energy-efficient as possible. Therefore, the performance of the proposed system should be tested with the following four metrics.

- *Localization error* (ε): It measures the difference between the estimated position (x', y') and its actual position (x, y) for any node. For the k -th node, such an error is represented using equation (3.13). Also, the average localization error (ε_{av}) in the network is expressed using equation (3.14).
- *Energy consumption* (ξ): It measures the amount of energy consumed by any node during the localization process. Therefore, the total amount of energy consumption (per bit) for the k -th node, completing the localization process in the i -th iteration, is expressed by modifying equation (3.1) as follows.

$$\xi_k = 2iE_r + \sum_i t'_{ki} E_b \quad (4.4)$$

where, t'_{ki} is the number of anchor points required in the i -th iteration. Likewise, the average energy consumption (ξ_{av}) in the network is expressed using equation (3.16).

- *Localization time* (τ'): This is the time taken to complete the localization process in any node. For the k -th node, localized in the i -th iteration, this time is usually calculated by modifying equation (3.17) as follows.

$$\tau'_k = \sum_i t'_{ki} T \quad (4.5)$$

Likewise, the average localization time (τ'_{av}) in the network is expressed using equation (3.18).

Table 4.3: Simulation parameters

Parameters	Values
Network size	1000 m × 1000 m
Number of nodes	100
Number of anchor	1
Number of malicious nodes	5 to 20
Maximum communication range	90 m
Energy consumption in electronic circuits	50 nJ/bit
Energy consumption in power amplifier	10 pJ/bit
SNR threshold level	-40 dBm
Signal broadcast period	1 sec
Initial energy of nodes	27 KJ

- *Success rate (SR)*: This measures the actual detection of benevolent and malicious nodes in the network. This means successful localization, minimizing the probability of a false positive (when a benevolent node is assumed to be malicious) or false negative (when a malicious node is assumed to be a benevolent node) detection rate on every attempt. For the i -th iteration, the success rate in the detection of false positives or false negatives is expressed (in percentage) as follows.

$$SR_i = \left\{ 1 - \frac{1}{2} \left(\frac{N_{fb}}{N_{tb}} + \frac{N_{fm}}{N_{tm}} \right) \right\} \times 100\% \quad (4.6)$$

where, the false positive or false negative rate is defined as the ratio of the number of benevolent (N_{fb}) or malicious nodes (N_{fm}) wrongly detected over the total number of benevolent (N_{tb}) or malicious nodes (N_{tm}) identified in this iteration. Likewise, the success rate of benevolent node detection (also called true positive rate) is defined as the ratio of the number of benevolent nodes (N_b) correctly detected in a particular iteration to the total number of benevolent nodes ($N_s - N_m$) in the network (excluding the number of malicious nodes as they are assumed to be the dead nodes). The true positive rate (TPR) for the i -th iteration is expressed (in percentage) as follows.

$$TPR_i = \frac{N_b}{N_s - N_m} \times 100\% \quad (4.7)$$

where, N_b and N_m are respectively the numbers of benevolent and malicious nodes.

4.5.2 Simulation environments

A set of simulations is carried out on the MATLAB software package (version 14). Several offline data are generated on PC (Personal Computer), keeping an analogy with the real-time deployment scenario. A network of the size 1000 m × 1000 m is considered. A set of 100 nodes is assumed to be scattered across the field. An anchor is also supposed to move across the sensor field in specific paths determined by its controller (centroid/fuzzy-based). The attack is carried out at the level of a few nodes capturing the IDs of their neighborhood (forgery type in the Replay attack) or choosing random IDs among others (replication type in the Sybil attack). Table 4.3 summarizes the simulation parameters for this work. These are compatible with MICAz motes, using the IEEE 802.15.4 MAC protocol for narrowband transmission at a data rate of 250 kbps at 2.4 GHz.

4.5.3 Simulation results

In the network, the ratio between benevolent nodes and malicious nodes is varied to measure the robustness of the proposed system.

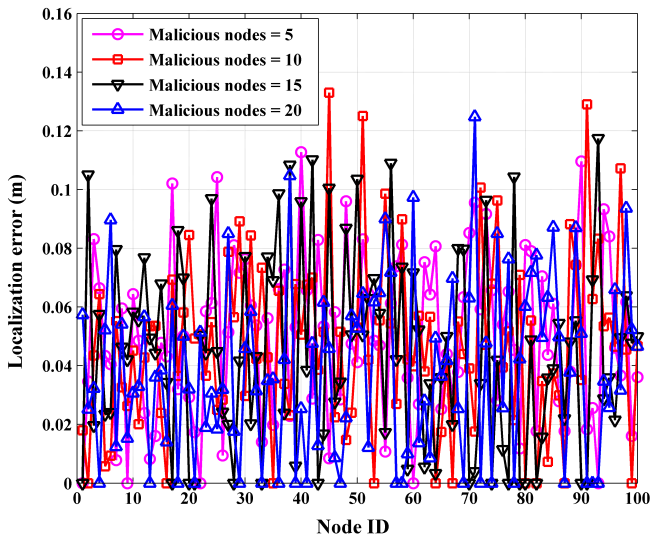


Fig. 4.11. Localization error for each node

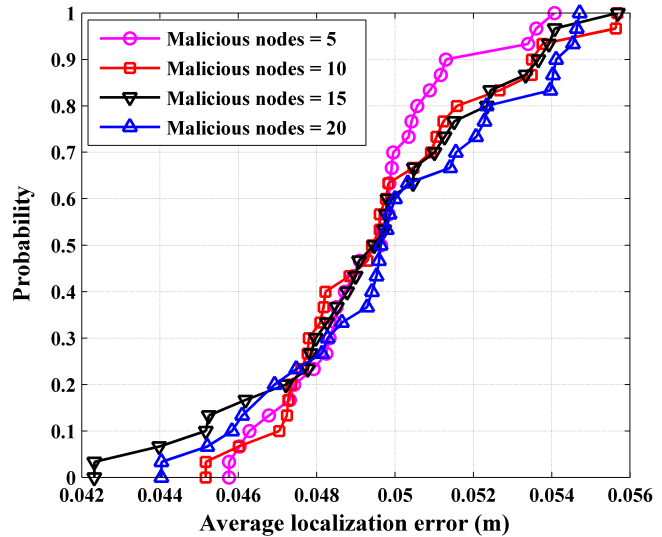


Fig. 4.12. Average localization error

A set of simulations is performed based on the metrics mentioned above. A total of 30 runs of the program are considered in each case to justify the results. Fig. 4.11 and Fig. 4.12 show the plot of the localization error in each node and the Empirical Cumulative Distribution Function (ECDF) of the average localization error for different numbers of malicious nodes, respectively. It confirms that the localization error varies proportionally to the number of malicious nodes.

Likewise, Fig. 4.13 and Fig. 4.14 show the plot of the energy consumption in each node and the Empirical Cumulative Distribution Function (ECDF) of the average energy consumption for different numbers of malicious nodes, respectively. Note that energy consumption also varies with the number of malicious nodes.

Besides, Fig. 4.15 and Fig. 4.16 show the plot of the localization time in each node and the Empirical Cumulative Distribution Function (ECDF) of the average localization time for different numbers of malicious nodes, respectively. This indicates that the localization time depends on both the density of benevolent and malicious nodes. When the ratio between benevolent and malicious nodes becomes high, the probability of suspicious nodes remains low. Thus, the anchor must travel much longer to explore them and relay the beacon messages.

Similarly, Fig. 4.17 and Fig. 4.18 show the success rate in detecting malicious nodes and benevolent nodes, respectively. It specifies that the success rate improves over the iterations when the ratio of benevolent nodes to malicious nodes becomes high, leading to a decrease in the number of suspicious/malicious nodes in the network. However, the optimal results obtained for localization error and success rate (assuming $N_b = 95$ and $N_m = 5$) are compared with other well-known methods and given in Tables 4.4 and 4.5, respectively.

As these results often vary depending on the number of malicious nodes in the network, a comparative study of the average localization error and the success rate in detecting malicious nodes is presented in Fig. 4.19 and Fig. 4.20, respectively.

4.5.4 Performance analysis

Mobility control strategies preserve the privacy of the anchor location (only a negligible amount of GPS error exists). Thus, the localization error is due to an attack on the nodes. This usually varies with the number of malicious nodes in the network. Likewise, when the probability of malicious nodes remains high, it affects more nodes.

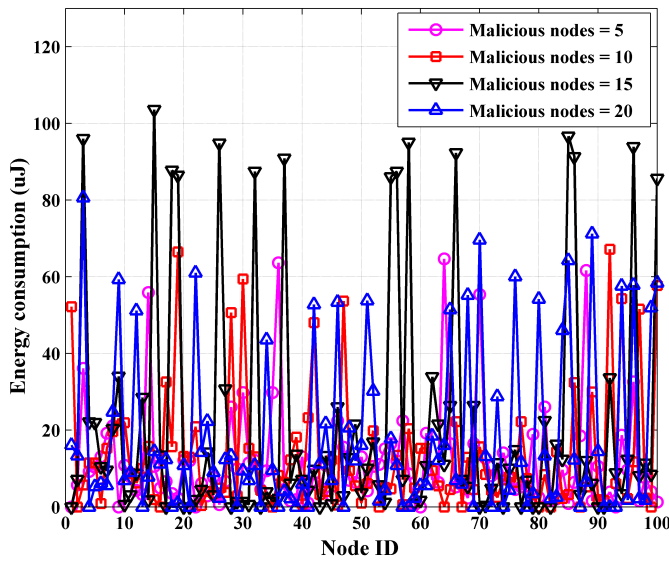


Fig. 4.13. Energy consumption for each node

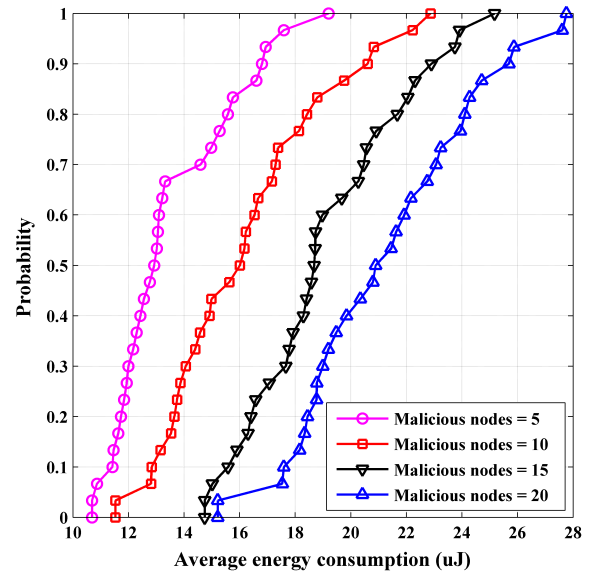


Fig. 4.14. Average energy consumption

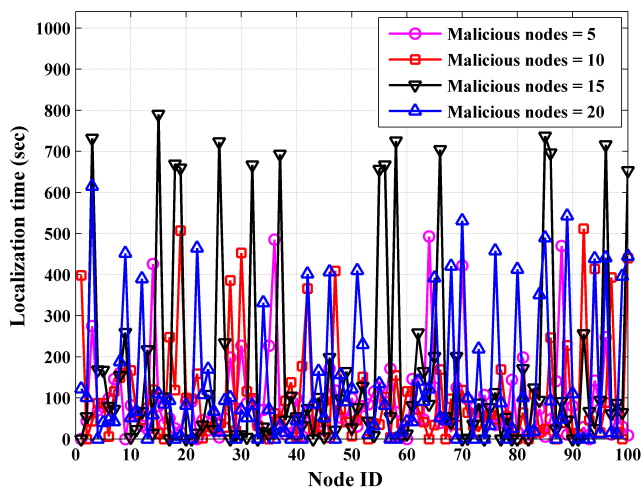


Fig. 4.15. Localization time for each node

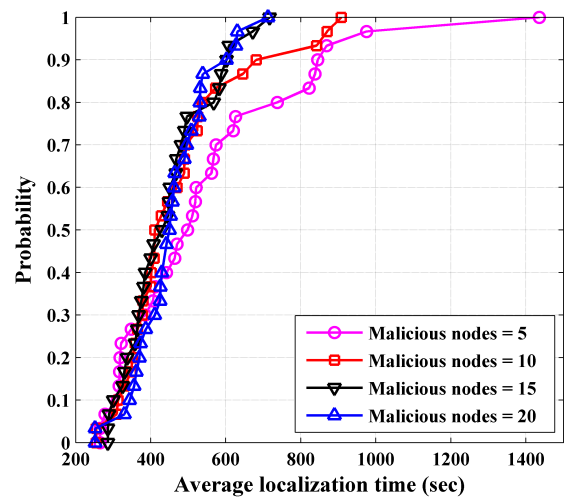


Fig. 4.16. Average localization time

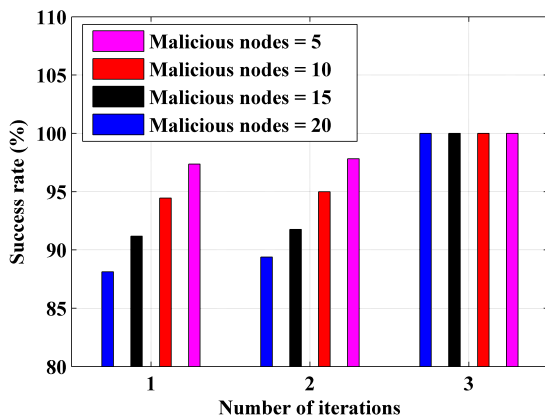


Fig. 4.17. Malicious node discovery rate

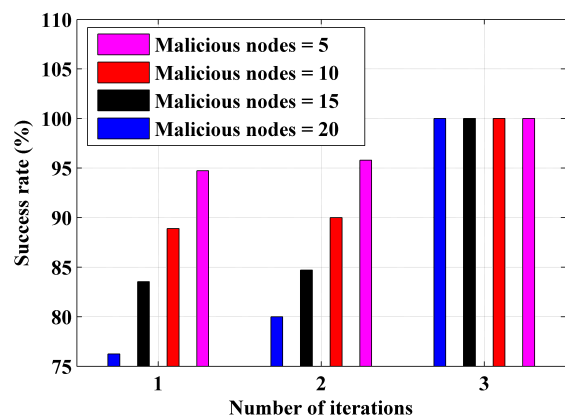


Fig. 4.18. Benevolent node discovery rate

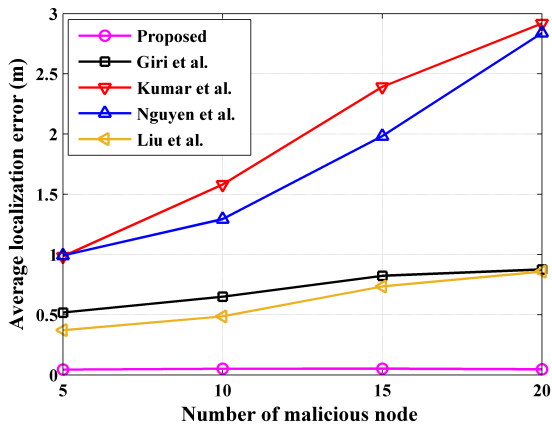


Fig. 4.19. Average localization error

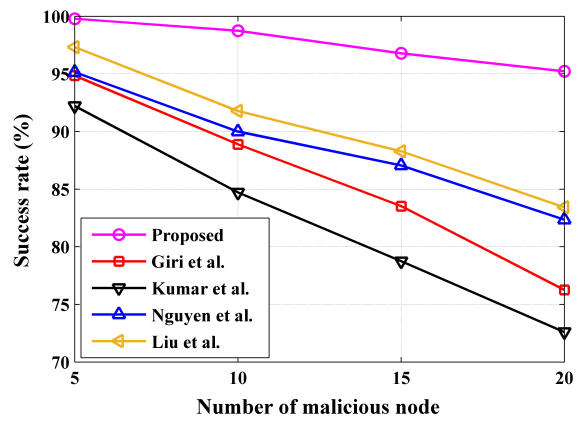


Fig. 4.20. Malicious node discovery rate

Table 4.4: Comparison of localization error and simulation time

Localization systems	Average localization error (m)	Simulation time (ms)
Fuzzy based anchor mobility control	0.056	4.3
Attack-resistant localization algorithm [72]	0.992	5.3
Information-theoretic approach [85]	0.518	10.6
Mutual authentication method [86]	0.985	12.4
Malicious node detection algorithm [88]	0.371	20.8

Table 4.5: Comparison of detection rate and time complexity

Localization systems	Success rate (%)	Computation overhead
Fuzzy based anchor mobility control	99.79	$O(\log n)$
Attack-resistant localization algorithm [72]	95.16	$O(n)$
Information-theoretic approach [85]	94.85	$O(n^2)$
Mutual authentication method [86]	92.22	$O(n^2)$
Malicious node detection algorithm [88]	97.32	$O(n \log n)$

It puts more nodes into active mode and reconsidering their localization results in higher power consumption. However, when the number of malicious nodes remains low, the result is fewer affected nodes. But localization often takes more time because the anchor has to travel more steps to explore these nodes on a low-density network. The iterative process converges very quickly and is independent of the number of deployed nodes. This system also imposes a small overhead for several reasons. Since the anchor is only equipped with GPS and a smart antenna as additional hardware, deployment costs remain the same regardless of the size of the network. Besides, a node should broadcast a hello message in each interval and only collect two beacon messages from the anchor until it is localized. It keeps power consumption to a minimum in each node by minimizing the number of messages needed and switching operations to different modes. Also, a node requires a small amount of memory to load these beacon messages and processed localization data. It ensures reduced communication and memory overhead. Since the number of suspicious nodes decreases significantly with each attempt of the iterative process, the computational overhead is also low.

4.6 Summary

In this chapter, we have proposed a secure and robust localization system based on the two anchor mobility control strategies. In this system, the nodes compute their positions in a distributed manner using the AoA information of two consecutive anchor points.

Here, the mobility controller (centroid/fuzzy-based) keeps the anchor in a path which must include identical neighboring nodes on two consecutive anchor points. In the event of an attack, nodes appearing with duplicate identities at any anchor point or missing at the next anchor point are considered suspicious. On the contrary, persistent nodes on both anchor points are considered benevolent. The position of any benevolent node is estimated with two separate beacon messages received during this period. Suspicious nodes that remain consistent in the next two iterations are malicious. Thus, they are filtered out of the localization process by detecting them through iterative checking. It is a more flexible system and can also be adopted in RSS or ToA-based systems. Its performance is evaluated with several metrics and obtained a satisfactory result. Simulation results with higher localization accuracy and greater success rate in malicious detection validate its effectiveness compared to existing systems. However, this system uses only one anchor, which leads to higher localization time and excessive power consumption in the nodes. Additionally, it assumes two simple attack models. Thus, it is still possible to evaluate its robustness with several anchors and under more complex attack scenarios in the future.

Chapter 5

Secure localization under hostile radio environments via a consistency check

5.1 Introduction

As discussed earlier, the base station must be aware of the source of the accumulated data to recognize an event in the network. And, this is now accomplished by incorporating a localization system where nodes autonomously determine their locations and relay this information as part of data packets. However, data communications over radio are unavoidable in WSNs and data packets remain fragile to attacks in the physical layer. Thus, a simple attack tampering with only the location information in data packets can lead to a significant deterioration in overall system performance. In this case, adversaries can introduce errors into data packets in three possible ways, such as

- capture relayed data packets and directly tampering with them,
- replay relayed data packets to spoof neighboring nodes and disrupt the functionality of one or more components of the localization systems, and
- modify the propagation characteristics of the radio environment over a specific region and hamper distance/angle estimation in nodes within it.

The beamforming attribute and direction finding capability of the smart antenna provide significant advantages for implementing a secure and robust localization system. Data packet integrity and confidentiality are ensured by establishing a more stable and private point-to-point link between anchors and relay nodes. By exploiting spatial diversity to counteract multipath interference, throughput and Quality of Service (QoS) are improved under a variable and unpredictable wireless medium. This must confirm the least probability of errors in the distance/angle estimation process.

In this chapter, we have proposed a new secure localization system using a smart antenna. It performs the selective segregation of the computed positions of a node iteratively in a corrupted radio environment. The consistent variants among the computed positions are considered to estimate the mean position of a node in the compromised environment (treated as malicious). To improve the convergence speed and energy efficiency in the localization process, the entire network area is divided into several sectors. And, for each individual sector, a particular anchor with a smart antenna and a GPS receiver is assumed to move on a random path. It would also avoid collisions between anchors. Fig. 5.1 shows a network with four sectors and a few region of corrupted radios. Nodes compute their positions based on beacon messages (AoA information and anchor reference) obtained from a few mobile anchors. After receiving at least three sets of beacon messages, nodes can initiate selective segregation to remove inconsistent position estimations in a hostile radio environment.

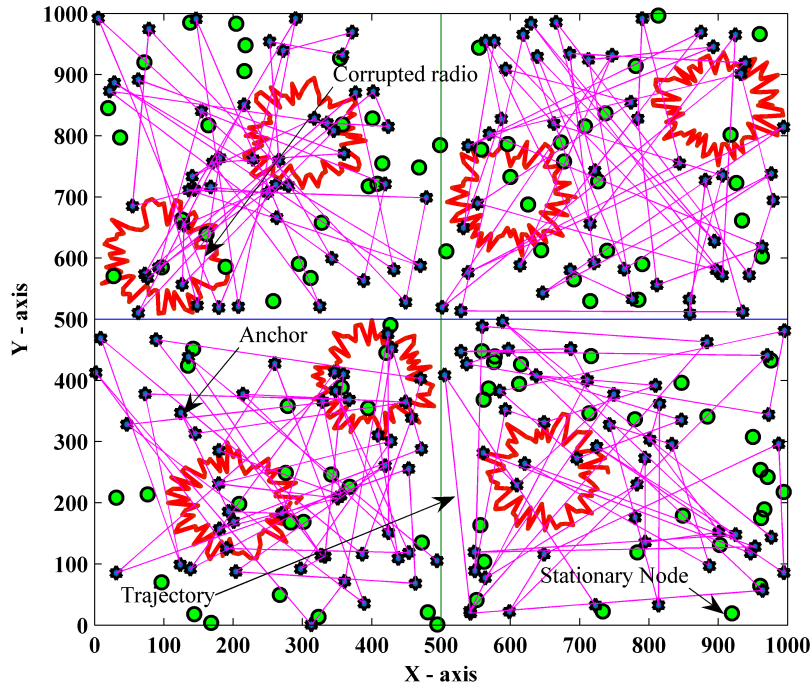


Fig. 5.1. Network architecture with attack scenarios

The rest of this chapter is organized as follows. In Section 5.2, an overview of attack scenarios is described. In Section 5.3, the strategy of Consistent Variant Assortment (CVA) is discussed. In Section 5.4, the methodology for implementing the proposed secure localization system is explained in detail. In Section 5.5, the simulation results are presented. Finally, Section 5.6 discusses shortcomings of the proposed system and possible modifications in the future.

5.2 Attack scenarios

The attack is primarily seen as impairing the normal functioning of a localization system. As a result, the adversary is expected to perturb the RSS/AoA measurement (distance/angle estimation) in some nodes. The network ground is assumed to have many obstacles (acting as radio frequency reflectors) or free space in a particular region is assumed to be covered with several clouds of smoke (changing the dielectric properties of the wireless medium). Thus, nodes located in such a physical/radio environment must always endure a considerable amount of variation in their signal strengths, directions and propagation time. Because the signal is absorbed at obstacles by several reflections/diffraction (causing multipath fading) or its speed is slowed down due to a sudden change in the dielectric properties of the corrupted wireless medium. Thus, the broadcast signals from neighboring nodes appear to be far from their actual positions. This would create misleading information at the receiving anchors and lead to wrong distance estimation in the RSS/ToA based localization system. Also, the directions of broadcast signals from neighboring nodes are changed by multiple reflections/diffraction at obstacles (changing the direct and dominant path of the signals) or by a rapid change in the dielectric properties of the corrupted wireless medium (causing refraction). This would produce an incorrect angle estimation in the AoA based localization system. The speed (v_r) and the wavelength (λ) of a radio wave always vary inversely proportional to the square root of the permittivity (ϵ_r) of the medium. When the signal propagates in a corrupted radio environment, the distance estimation error (ϵ_d) can be expressed using the Friis transmission equation in free space.

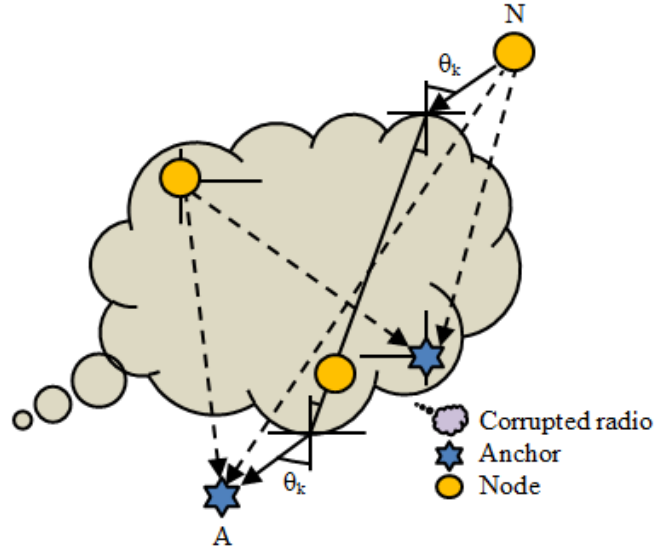


Fig. 5.2. RSS/AoA estimation in corrupted radio

For the RSS-based localization system, it is written as follows.

$$\varepsilon_d = \left(1 - \frac{1}{\sqrt{\epsilon_r}}\right) \frac{\lambda}{4\pi} \sqrt{\frac{P_t G_t G_r}{P_r}} \quad (5.1)$$

where, P_t, P_r and G_t, G_r are the power and the gain of transmitting/receiving antenna, respectively.

Likewise, for the ToA/TDoA based localization system, it is written using the relationship between velocity and propagation time as follows.

$$\varepsilon_d = \left(1 - \frac{1}{\sqrt{\epsilon_r}}\right) \tau_p v_r \quad (5.2)$$

where, τ_p and v_r denote the time and the velocity of wave propagation, respectively. On the other hand, an oblique wave incidence is seen at the interface of corrupted radios (as they often have irregular geometry) and hence a significant deviation in the angle of incidence would be noticed due to refraction in the corrupted environment. The angle estimation error (ε_θ) can be expressed using Snell's law. For the AoA-based localization system, it can be written as follows.

$$\varepsilon_\theta = \left| \theta_k - \sin^{-1} \left(\frac{1}{\sqrt{\epsilon_r}} \sin \theta_k \right) \right| \quad (5.3)$$

where, θ_k represents the direction of the signal propagating in a tidy radio environment.

The RSS/AoA measurement (distance/angle estimation) under a corrupted radio is illustrated in Fig. 5.2. The distance estimation would be erroneous whenever an anchor receives broadcast signals from nodes that cross the corrupted medium. This is valid for all cases, irrespective of the position of the node and the anchor being outside or inside the corrupted radio (as indicated by the dotted lines). Likewise, the error in the angle estimation only becomes significant if the mobile anchor (A) and the stationary node (N) remain in a different medium (as indicated by the solid line).

5.3 Consistent variant assortment

Estimated positions in a corrupted radio environment often remain unreliable. Therefore, the trustworthiness of these estimates should be improved by applying a suitable verification process.

For this purpose, the position of each node is computed several times by taking any two distinct combinations of beacon messages using equation (4.3). Further, all the possible variants among these estimated positions are generated. However, the precision of the estimation is ensured in the evaluation of the optimal position of a node by considering a selective segregation of the variants and by choosing only those which are consistent. Thus, an optimal position is taken as the mean of all the estimated positions yielding the consistent variants (the variance between the two estimated positions is lower than the maximum permissible limit). This process is shown in Fig. 5.3 and explained in Algorithm 9. Consider that a possible combination of p distinct positions is generated in each node, where $p = {}^n C_2 = \frac{n(n-1)}{2}$. The variants are made by differentiating each position from the others in the form of $\delta_{ij} = \sqrt{(x_i - x_{i+j})^2 + (y_i - y_{i+j})^2}$ where, $i = 1, 2, \dots, (p-1)$ and $j = 1, 2, \dots, (p-i)$. Therefore, a sum of q variants is produced in total, where $q = {}^p C_2 = \frac{p(p-1)}{2}$.

In this work, a maximum admissible limit (δ_{max}) is set to measure the consistency (Ω_c) of the variants so obtained. A node is considered benevolent when it has at least 50% (percentage) of variants below this limit, otherwise it is treated as malicious. Malicious nodes are again taken into account in the localization process in an iterative manner. The parameters, SNR_{th} and δ_{max} are updated in the following iterations. Malicious nodes reconfigure themselves in active mode and refresh their memory at all iterations. The process continues in the same way until all nodes become benevolent. Each time a node becomes a benevolent; its position is finally computed.

Algorithm 9 Pseudo-code for consistent variant assortment process in node position estimation

Input: Estimated positions for the specific node: $[X]$, Total number of estimated positions (X_{total}), Maximum permissible variations that signifies the consistency (δ_{max}), Threshold level in signal-to-noise ratio (SNR_{th}) and Power saving mode of node (Idle/Sleep): mode state (ms) = ‘Reset’
Output: Final estimated position after the consistency verification: $[X_{mean}]$ and Active mode of node (Idle/Sleep): mode state (ms) = ‘Set’

```

1: Initialize:  $i, j = 1, count, sum = 0, \eta_0, \gamma_0, \delta_{max}, X_{total}, SNR_{th}, ms = \text{'Set'}$ 
2: while ( $i \leq X_{total} - 1$ ) do
3:   while ( $j \leq X_{total} - i$ ) do
4:      $\delta_{ij} \leftarrow \sqrt{(x_i - x_{i+j})^2 + (y_i - y_{i+j})^2}$ 
5:     if ( $\delta_{ij} \leq \delta_{max}$ ) then
6:        $count \leftarrow count + 1; X \leftarrow \frac{X_i + X_{i+j}}{2}; sum \leftarrow sum + X$ 
7:     else
8:       continue
9:     end if
10:     $j \leftarrow j + 1$ 
11:  end while
12:   $i \leftarrow i + 1$ 
13: end while
14:  $\Omega_c \leftarrow \frac{count}{X_{total}}$ 
15: if ( $\Omega_c \geq 0.5$ ) then
16:    $X_{mean} \leftarrow \frac{sum}{count}$ 
17: else
18:    $SNR_{th} \leftarrow \eta_0 * SNR_{th}; \delta_{max} \leftarrow \gamma_0 * \delta_{max}$ 
19: end if
20:  $ms \leftarrow Set$ 
21: return  $ms, X_{mean}$ 

```

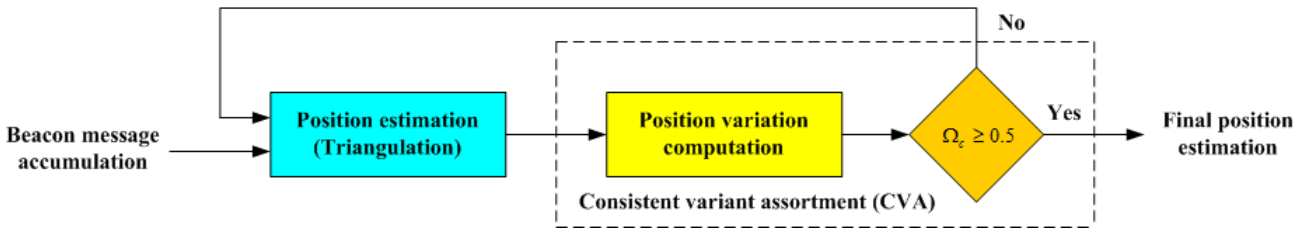


Fig. 5.3. CVA for position estimation in corrupted radio

5.4 Localization system

Accurately locating the sources of relayed data packets is essential for various operations such as security surveillance, enemy detection and tracking on the battlefield. From this point of view, the localization system must be made as robust and secure as possible in a hostile radio environment. In this work, we have proposed such a system based on a few mobile anchors. Using smart antennas, anchors receive signals from neighboring nodes and estimate their AoA information. This AoA information and the current reference of an anchor are used to generate a beacon message, and transfer it to the corresponding nodes. After collecting the prescribed number of such beacon messages, each node formulates straight-line equations and computes its position independently as the point of intersection between two straight-lines, each framed by a unique beacon message. The security (or accuracy) of the localization process is preserved by measuring and selecting only the consistent estimates among several estimated positions, in an iterative manner.

Node ID (bits)	AoA (rad)	Anchor reference (meter)
k	θ_k	$[x_a \ y_a]$

Fig. 5.4. Format of a beacon message

5.4.1 Beacon message generation/transmission

In the network, the nodes are supposed to periodically broadcast signals and the anchors only receive them from neighboring nodes having an RSS greater than SNR_{th} . The anchors are equipped with both GPS receivers and smart antennas. An anchor determines the AoA information of all received signals using the ESPRIT algorithm. It also generates and transmits beacon messages for these nodes. A beacon message contains important data for locating a node. An anchor accomplishes this by storing the AoA data in its memory along with the respective node IDs. Moreover, it amalgamates its current reference retrieved from the GPS receiver to the same memory locations. The format of a beacon message is given in Fig. 5.4.

However, the GPS receiver expresses the anchor reference in latitude (γ') and longitude (ψ') in geodetic coordinates. Thus, it must be converted to Cartesian coordinates (x_a, y_a) for position computation using the geometric method (triangulation). Such coordinate transformation is performed using equation (3.4). To relay beacon messages to nodes, an anchor uses the adaptive beamforming capabilities of smart antennas and a proper scheduling mechanism. However, a stochastic process is often desirable to create such a pattern with a narrower beamwidth and lower sidelobe level. This can ensure the security of beacon transmission by producing a more stable radio link. For each time slot, the anchor is expected to receive signals from a few neighboring nodes with different arrival time or received signal strength (because a signal with lower RSS is expected to travel a long time/distance).

Thus, an anchor can always establish a direct link using the optimal pattern to send a beacon message to a particular node, steering the deep nulls in the directions of other neighboring nodes. Algorithm 10 explains the process of generating/transmitting beacon messages.

Algorithm 10 Pseudo-code for generating and transmitting beacon messages

Input: Maximum number of permissible anchor points (t_{max}), Angle estimated from the received signals (φ), Distance estimated from the received signals (d), Cluster size: $[CS]$, Clustering node IDs: $[ID]$, Anchor reference (x_a, y_a), Number of antenna elements ($2N$), Beamwidth between the first nulls ($FNBW$), Sidelobe level (SLL), Depth of nulls (K), Scanning angle (θ) and Idle mode of anchor: mode state (ms) = ‘Reset’

Output: Generated beacon messages: B_m [size], Optimized beam pattern (AF_p), Active mode of the anchor: mode state (ms) = ‘Set’

```

1: Initialize:  $i, j, k, n, t = 1, AF_p = 0, \theta = -90^0, size = 4, K, N, SLL, FNBW, ms = \text{'Reset'}, t_{max}$ 
2: while ( $t \leq t_{max}$ ) do
3:   while ( $i \leq CS$ ) do
4:      $\theta_d \leftarrow \varphi[1]$ 
5:     while ( $j \leq size$ ) do
6:        $B_m[j] \leftarrow ID[i]; ID[i] \leftarrow d[i]; d[i] \leftarrow \varphi[1]; \varphi[1] \leftarrow x_a; x_a \leftarrow y_a; j \leftarrow j + 1$ 
7:     end while
8:     while ( $\theta \leq 90^0$ ) do
9:       if ( $|\theta - \theta_d| \leq \frac{FNBW}{2}$ ) then
10:         $AF_p \leftarrow 1$ 
11:       else
12:         $AF_p \leftarrow SLL$ 
13:       end if
14:       while ( $k \leq CS - 1$ ) do
15:         $\theta_n \leftarrow \varphi[k + 1]$ 
16:        if ( $\theta = \theta_n$ ) then
17:           $AF_p \leftarrow K$ 
18:        end if
19:         $\varphi[k] \leftarrow \varphi[k + 1]; k \leftarrow k + 1$ 
20:       end while
21:       while ( $n \leq N$ ) do
22:         $AF_p \leftarrow AF_p + W[n] * \cos \left[ \left( n - \frac{1}{2} \right) * \pi * (\sin \theta - \sin \theta_d) \right]; n \leftarrow n + 1$ 
23:       end while
24:        $\theta \leftarrow \theta + 1$ 
25:     end while
26:      $\varphi[CS] \leftarrow \theta_d; i \leftarrow i + 1$ 
27:   end while
28:    $t \leftarrow t + 1$ 
29: end while
30:  $ms \leftarrow Set$ 
31: return  $ms, AF_p, B_m[size]$ 

```

5.4.2 Position computation

For an AoA-based localization system, only two beacon messages are required to estimate the position of any node. But, the verification of the estimated position is essentially needed in a hostile radio environment. Thus, each node is allowed to store a prescribed number of beacon messages (n) within an acceptable time interval (t_{max}). Then they can toggle themselves to sleep mode. Each node is periodically switched to active/receive mode during the permissible time interval.

Therefore, a counter is enabled to verify the prescribed number of beacon messages in its memory. Algorithm 11 explains the process of accumulating beacon messages. In this work, the position of a node is computed using the triangulation method. This requires formulating a pair of straight-line equations using a distinct pair of stored beacon messages. Thus, it is possible to represent a set of n straight-lines passing through each node as shown in Fig. 5.5.

For the k -th node, it is expressed as follows.

$$\left. \begin{aligned} y_k - y_{a1} &= \tan \theta_{k1} (x_k - x_{a1}) \\ y_k - y_{a2} &= \tan \theta_{k2} (x_k - x_{a2}) \\ &\dots \\ y_k - y_{an} &= \tan \theta_{kn} (x_k - x_{an}) \end{aligned} \right\} \quad (5.4)$$

where, θ_{kn} and (x_{an}, y_{an}) denote the corresponding AoA and anchor reference in the n -th beacon message. Thus, the positions of the k -th node are computed using equation (4.3), solving any two of the straight-line equations each time, yielding $n(n-1)/2$ positions in total for n beacon messages. However, considering only the AoA estimation error in a hostile radio environment, the estimated positions of a node always remain different. And, the final position of a node is estimated by taking the mean of all positions producing consistent variants. Algorithm 12 explains the triangulation method of position computation.

Algorithm 11 Pseudo code for accumulating beacon messages in each node

Input: Maximum number of permissible anchor points (t_{max}), Anchor generated beacon messages: B_m [size], Maximum number of permissible beacon messages (n), Received signal strength (RSS), Threshold level in signal-to-noise ratio (SNR_{th}) and Receive mode of node: mode state (ms) = ‘Set’

Output: Accumulated beacon messages: U' [count] [size] and Power saving mode of node (Idle/Sleep): mode state (ms) = ‘Reset’

```

1: Initialize:  $t = 1, count = n, ms = \text{'Set'}, t_{max}, SNR_{th}$ 
2: while ( $t \leq t_{max}$ ) do
3:   if ( $RSS \geq SNR_{th}$ ) then
4:      $U'[count][size] \leftarrow B_m[size]; count \leftarrow count - 1$ 
5:   else
6:     continue
7:   end if
8:   if ( $count = 0$ ) then
9:     break
10:  else
11:    continue
12:  end if
13:   $t \leftarrow t + 1$ 
14: end while
15:  $ms \leftarrow \text{Reset}$ 
16: return  $ms, U'[count][size]$ 

```

5.4.3 Design methodology

In this work, the First Come, First Served (FCFS) scheduling method is used for the generation and transmission of beacon messages. A suitable ring counter is designed to generate trigger pulses to switch anchors/nodes to operate in different modes. On the synchronization sequence illustrated in Fig. 5.6, nodes are assigned an equal time slot to broadcast signals (t_b) and receive beacon messages (t_r) from an anchor, as indicated by $t_b = t_r = \frac{d_{max}}{v_r} + \tau_c$.

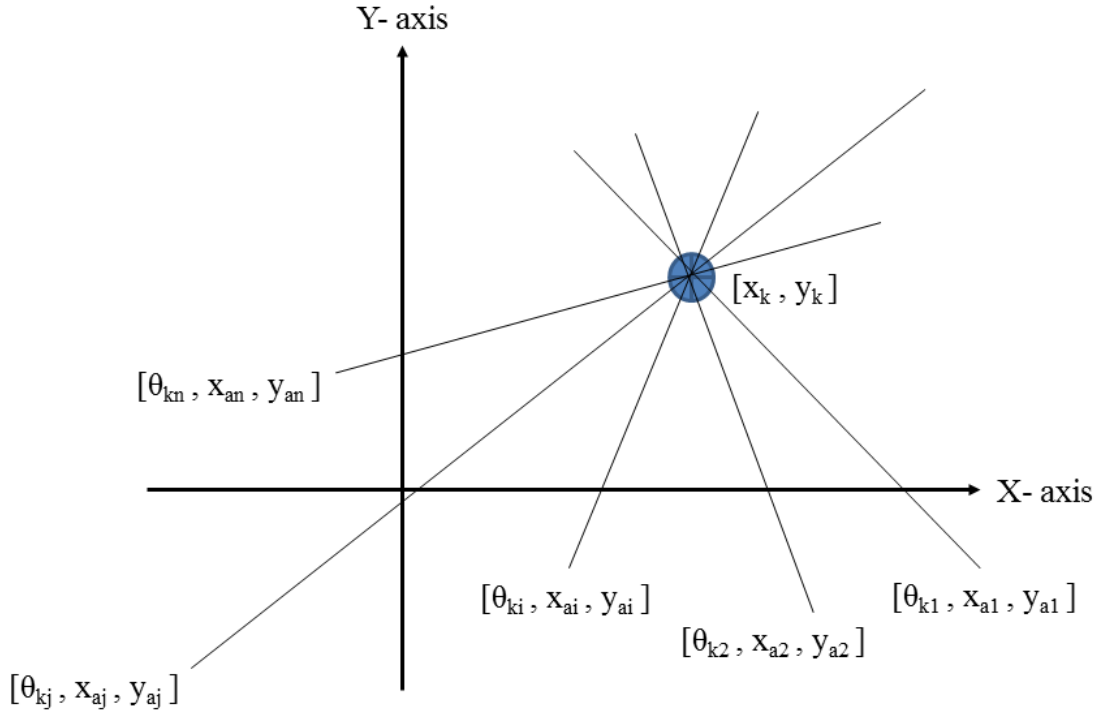


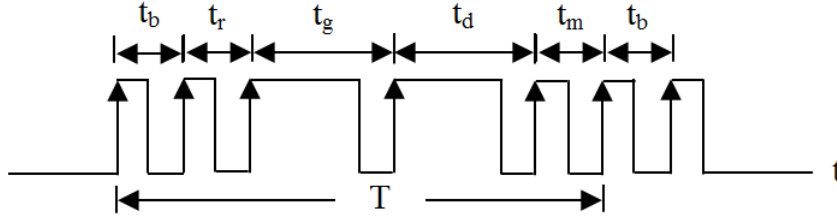
Fig. 5.5. Triangulation method for position computation

Algorithm 12 Pseudo-code for estimating positions in each node

Input: Accumulated beacon messages: U' [count] [size], Maximum number of permissible beacon messages (n) and Idle mode of node: mode state (ms) = 'Reset'

Output: Estimated positions for the specific node: $[X]$, Total number of positions estimated (X_{total}) and Power saving mode of node (Idle/Sleep): mode state (ms) = 'Reset'

- 1: **Initialize:** $i, j = 1, count = 0, n, ms = \text{'Reset'}$
 - 2: **while** ($i \leq n - 1$) **do**
 - 3: **while** ($j \leq n - i$) **do**
 - 4: $B_0[i][j] \leftarrow \begin{bmatrix} -\tan \theta[i] & 1 \\ -\tan \theta[i+j] & 1 \end{bmatrix}; c_0[i][j] \leftarrow \begin{bmatrix} y_a[i] - \tan \theta[i] * x_a[i] \\ y_a[i+j] - \tan \theta[i+j] * x_a[i+j] \end{bmatrix}$
 - 5: $X[i][j] \leftarrow B_0[i][j]^{-1} * c_0[i][j]; count \leftarrow count + 1$
 - 6: $j \leftarrow j + 1$
 - 7: **end while**
 - 8: $i \leftarrow i + 1$
 - 9: **end while**
 - 10: $X_{total} \leftarrow count$
 - 11: $ms \leftarrow \text{Reset}$
 - 12: **return** ms, X, X_{total}
-


Fig. 5.6. Trigger pulses for mode switching

Here, d_{max} and τ_c represents the maximum communication range and signal processing time in nodes/anchors, respectively. Also, the time of generation (t_g) and distribution (t_d) of beacon messages to nodes is assumed equal such that $t_g = t_d = CS(\tau_c + \tau_d)$. Here, CS and τ_d are the number of neighboring nodes chosen at a given time and the set up time of amalgamation/link, respectively. A permissible time interval as given by $t_m = T - 2(t_b + t_d)$ is set for the anchor to move to the next reference. T is the time period. With a real-time implementation strategy, the controller circuits for an anchor and a node are both depicted with a separate Finite State Machine (FSM) shown in Fig. 5.7. Also, their functionality in each state is defined in Table 5.1 and Table 5.2.

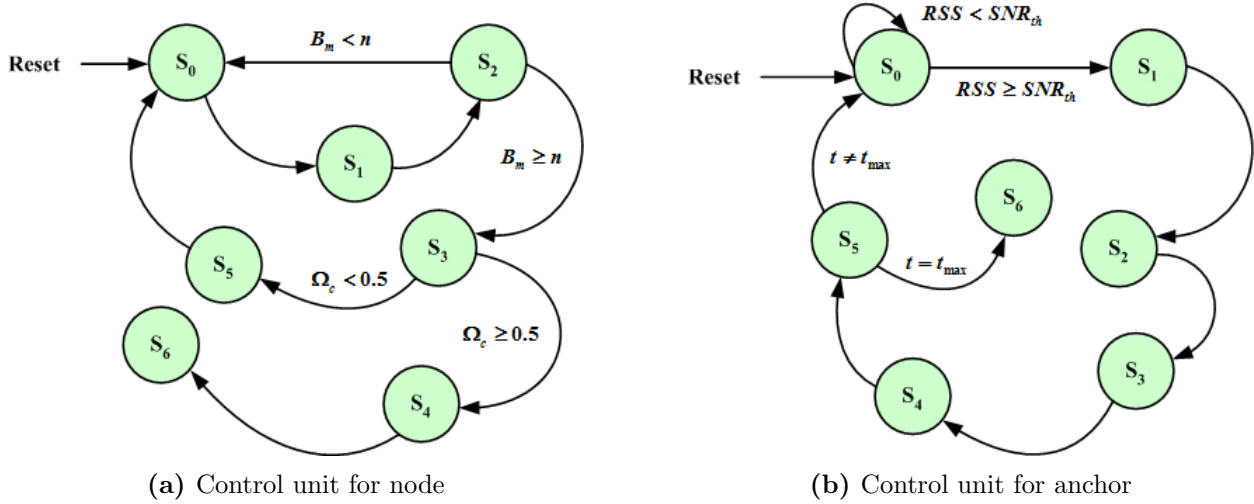

Fig. 5.7. FSM for anchor and node

Table 5.1: Anchor behaviors in FSM

State	Command	Action	Mode
S_0	Reset, $t \neq t_{max}$	Disable servo system, Receive signal	Receive
S_1	$RSS \geq SNR_{th}$	Estimate AoA, Generate beacon	Idle
S_2	Clock	Distribute beacon	Active
S_3	Clock	Wait for next broadcast signal	Idle
S_4	Clock	Wait for next broadcast signal	Idle
S_5	Clock	Actuate servo system	Idle
S_6	$t = t_{max}$	Disable servo system, Wait for new instructions	Sleep

5.5 Performance evaluation

In this work, the AoA estimations in the anchors are considered to be erroneous when the signals received from the nodes pass through a medium with different dielectric properties (causing the refraction).

Table 5.2: Node behaviors in FSM

State	Command	Action	Mode
S_0	Reset, Clock, $B_m < n$	Broadcast signal	Active
S_1	Clock	Wait for beacon	Idle
S_2	Clock	Receive beacon	Receive
S_3	$B_m \geq n$	Calculate Ω_c	Idle
S_4	$\Omega_c \geq 0.5$	Estimate position	Idle
S_5	$\Omega_c < 0.5$	Update SNR_{th} and δ_{max}	Idle
S_6	Clock	Wait for new instructions	Sleep

Thus, we classified a few nodes as malicious in WSN, these producing inconsistent variants in their computed positions. By dynamically adjusting the SNR_{th} , the proposed system often ensures signal propagation over an identical radio environment, reducing the AoA estimation error in malicious nodes. Likewise, by iteratively increasing δ_{max} , the localization process tends to end with a higher convergence speed.

5.5.1 Performance metrics

The proposed system must be as robust and energy-efficient as possible to perform well in a hostile radio environment. Thus, its performance is verified under three benchmark functions as follows.

- *Localization error* (ε): It measures the difference between the estimated position (x', y') and its actual position (x, y) for any node. For the k -th node, such an error is represented using equation (3.13). Also, the average localization error (ε_{av}) in the network is expressed using equation (3.14).
- *Energy consumption* (ξ): It measures the amount of energy consumed by any node during the localization process. Therefore, the total amount of energy consumption (per bit) for the k -th node, completing the localization process in the i -th iteration, is expressed by modifying equation (4.4) as follows.

$$\xi_k = niE_r + \sum_i t'_{ki} E_b \quad (5.5)$$

where, n indicates the number of beacon messages used and t'_{ki} denotes the anchor points needed for the i -th iteration. Likewise, the average energy consumption (ξ_{av}) in the network is expressed using equation (3.16).

- *Localization time* (τ'): This is the time taken to complete the localization process in any node. For the k -th node, localized in the i -th iteration, this time is usually calculated using equation (4.5). Likewise, the average localization time (τ'_{av}) in the network is expressed using equation (3.18).

5.5.2 Simulation environments

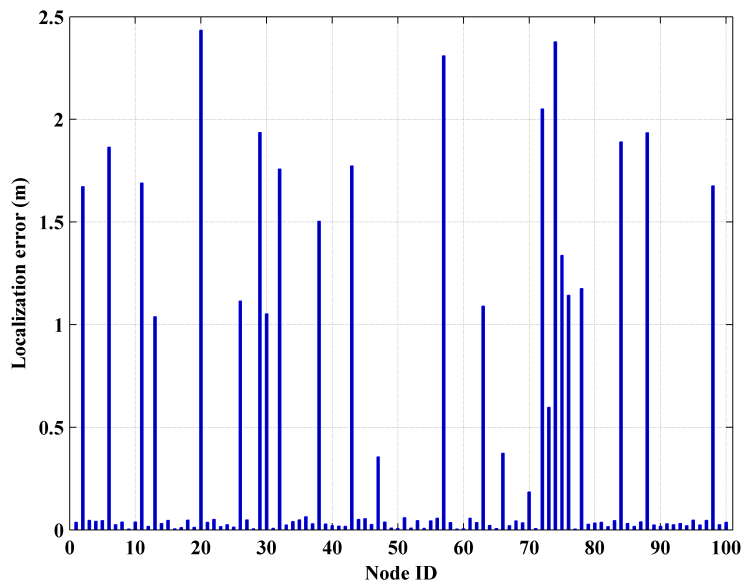
A set of simulations is carried out on the MATLAB software package (version 14). To keep an analogy with the real-time deployment scenario, several off-line data are generated on PC (Personal Computer). As a network architecture, a total of 100 stationary nodes are considered to be scattered over an area of 1000 m \times 1000 m. Each anchor is assumed to move along random trajectories over a distinct sector of the sensor field. The attack is performed at a few nodes to disrupt AoA estimations by spreading clouds of smoke over specific regions (since smoke can improve air permittivity about 10 - 15 times). The simulation parameters are summarized in Table 5.3. They are compatible with MICAz motes, using the IEEE 802.15.4 MAC protocol for narrowband transmission at a data rate of 250 kbps in 2.4 GHz.

Table 5.3: Simulation parameters

Parameters	Values
Network size	1000 m \times 1000 m
Number of nodes	100
Number of anchor	1 to 5
Number of malicious nodes	5 to 25
Maximum communication range	100 m
Energy consumption in electronic circuits	50 nJ/bit
Energy consumption in power amplifier	10 pJ/bit
SNR threshold level	-40 dBm to -10 dBm
Maximum permissible variation	1 to 3

5.5.3 Simulation results

A set of simulations are performed in terms of performance metrics as mentioned earlier. In WSN, the density of the anchor and malicious node are varied to measure the robustness of the proposed system. To justify the results, 50 runs of the program are considered in each case. The localization error obtained at each node (considering five anchors and twenty five malicious nodes) is shown in Fig. 5.8. Using different numbers of anchors and malicious nodes, the average

**Fig. 5.8.** Localization error for each node

localization error is also plotted in Fig. 5.9 (a), Fig. 5.9 (b) and Fig. 5.9 (c). It is evident that the localization error varies proportionally to the density of malicious nodes. However, no significant change in result is observed for varying the density of the anchors.

Similarly, the energy consumption obtained at each node (considering five anchors and twenty five malicious nodes) is shown in Fig. 5.10. Choosing different numbers of anchors and malicious nodes, the average energy consumption is also shown in Fig. 5.11(a), Fig. 5.11(b) and Fig. 5.11(c). We see that the energy consumption varies with the density of malicious nodes. However, a small effect is noticed for varying the anchor density.

Likewise, the localization time obtained at each node (considering five anchors and twenty five malicious nodes) is shown in Fig. 5.12. By setting different numbers of anchors and malicious nodes, the average localization time is also plotted in Fig. 5.13 (a), Fig. 5.13 (b) and Fig. 5.13 (c).

It should be noted that the localization time depends on both the anchor and the density of malicious nodes. Since the anchors divide the entire network into several sectors, the nodes can receive the necessary beacon messages in the shortest possible time.

However, the localization process at the nodes can begin with the acquisition of a minimum of three beacon messages. But, more variants are often desirable to achieve greater accuracy. On the other hand, the accumulation of more beacon messages requires more localization time, always violating the energy saving policy. Thus, we have provided a comparative study of various metrics for different numbers of maximum permissible beacon messages, in Table 5.4. Moreover, the simulation results are compared with other well-known methods and given in Table 5.5.

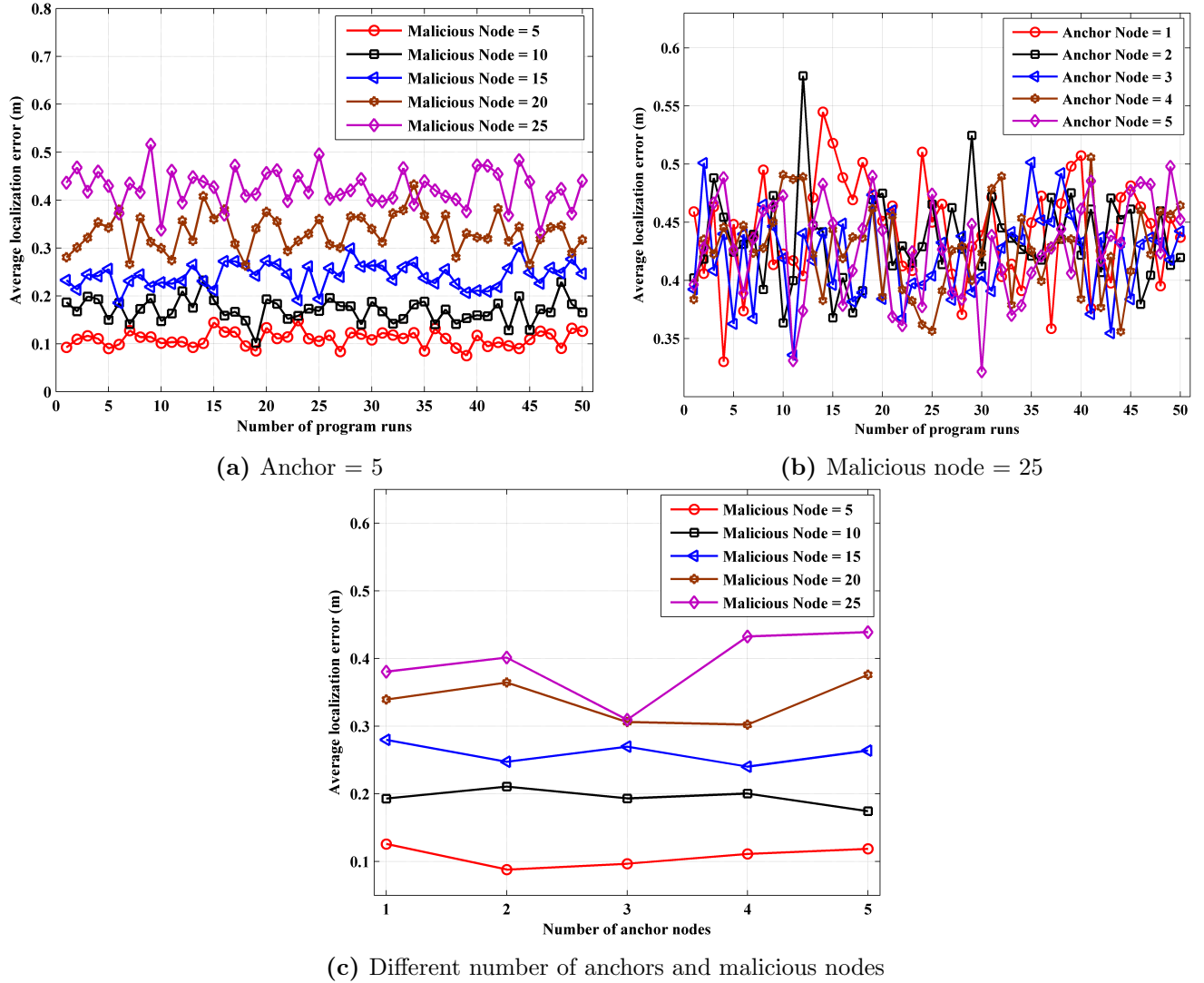


Fig. 5.9. Average localization error in WSN

Table 5.4: Results obtained with different beacons

Performance metrics	Number of permissible beacons			
	$n = 3$	$n = 4$	$n = 5$	$n = 6$
Average localization error (m)	0.4518	0.4479	0.4232	0.4080
Average energy consumption (nJ)	8.8067	9.3118	9.5458	10.122
Average localization time (sec)	168.67	177.75	180.31	192.49

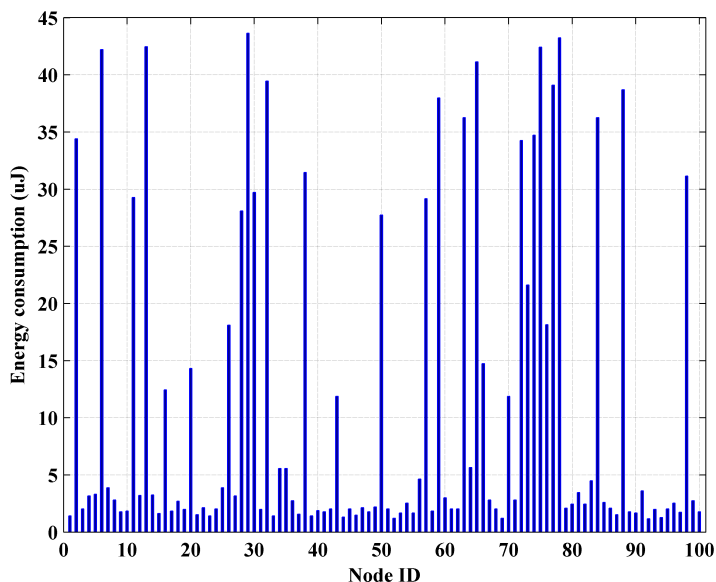
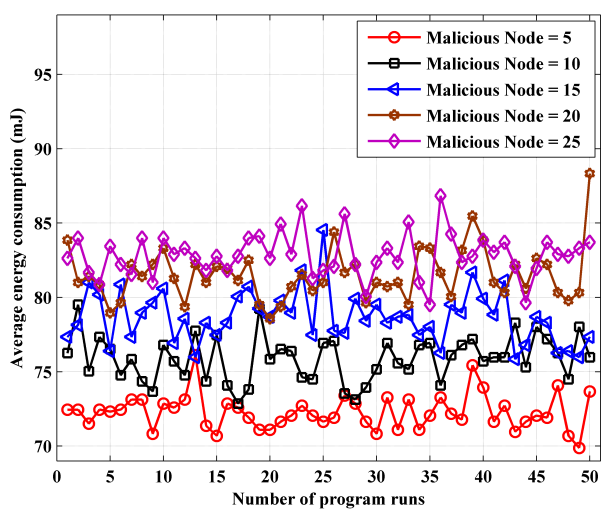
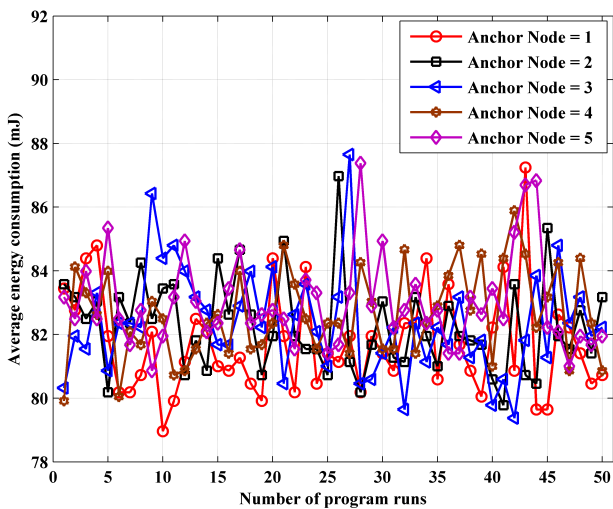


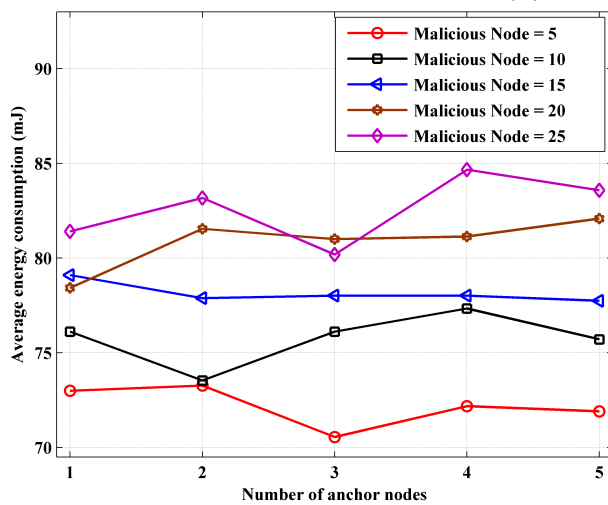
Fig. 5.10. Energy consumption for each node



(a) Anchor = 5



(b) Malicious node = 25



(c) Different number of anchors and malicious nodes

Fig. 5.11. Average energy consumption in WSN

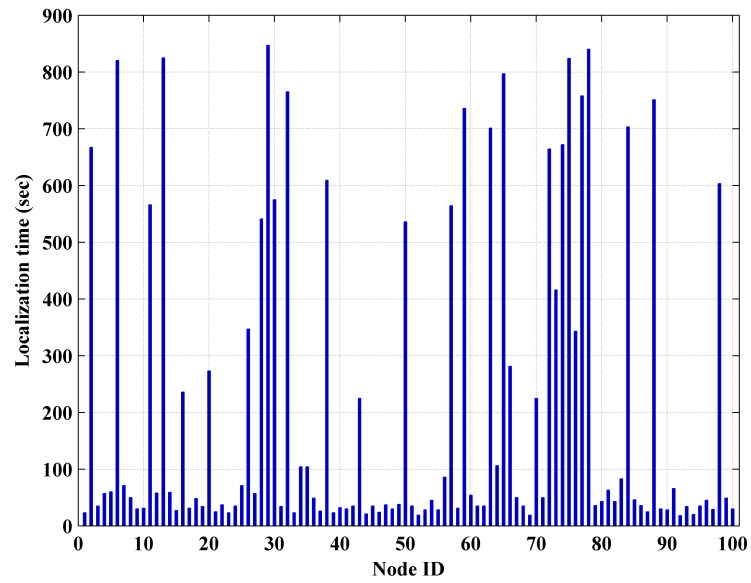
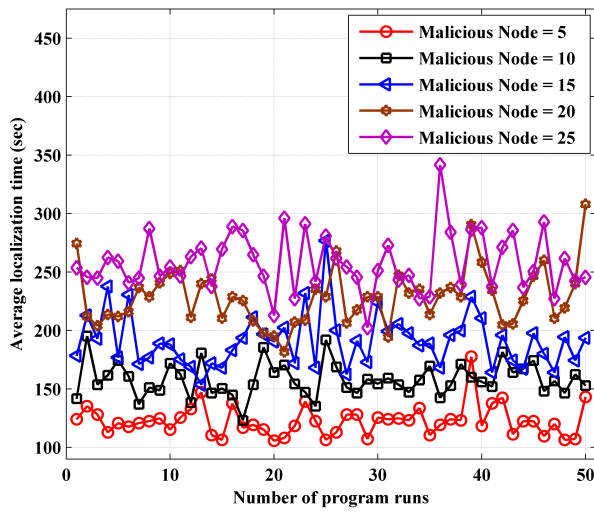
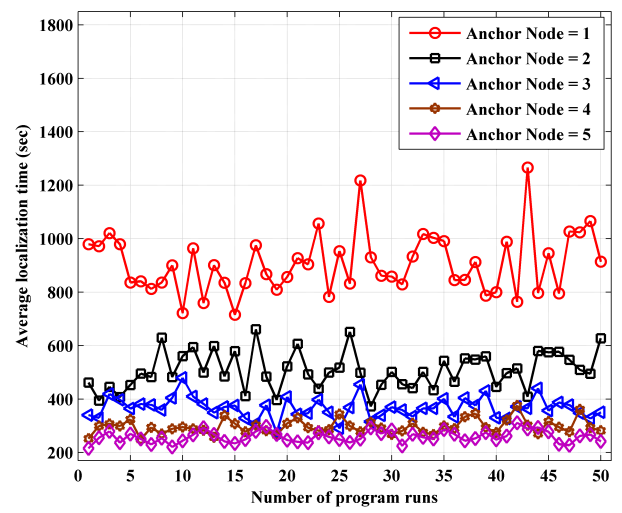


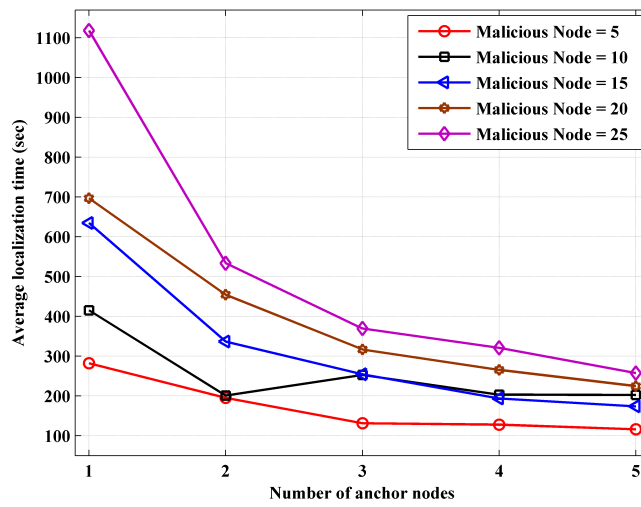
Fig. 5.12. Localization time for each node



(a) Anchor = 5



(b) Malicious node = 25



(c) Different number of anchors and malicious nodes

Fig. 5.13. Average localization time in WSN

Table 5.5: Comparison of results

Localization systems	Average localization error (m)	Simulation time (ms)
Consistent variant assortment	0.57	7.9
Practical and secure localization [68]	0.72	8.4
Two step secure localization [74]	0.83	9.2
Secure distance-based localization [83]	0.73	9.5
Gradient descent approach [91]	1.49	3.7
Least median square data fusion [106]	1.34	24.8

5.6 Summary

In this chapter, we have proposed a new secure localization system using multiple mobile anchors. The computation of the position of the nodes is based on the AoA information estimated with the ESPRIT algorithm. In a tampered radio environment, the system must verify the consistency of the positions computed in each node. A node is considered benevolent to comply with at least 50% of the consistency verification, otherwise malicious. In this system, the position of a node is computed, only when it becomes benevolent in WSN, by taking the mean of all its estimated positions producing the consistent variants. Moreover, malicious nodes are upgraded to benevolent nodes by adjusting the threshold value of the signal-to-noise ratio and the maximum permissible variation, iteratively. It is a flexible system that can also be adopted in RSS/ToA-based systems. Its performance is tested under several metrics. Simulation results with higher accuracy validate its effectiveness comparable to existing systems. However, the system is realized with a simple attack scenario. Thus, it is still possible to test its robustness in more complex attack scenarios in the future.

Chapter 6

Anchor data security under PHY layer attacks via antenna pattern control

6.1 Introduction

As mentioned earlier, localization systems involve anchor reference data in the computation of node positions. Thus, an attack aimed at distorting the anchor reference data can cause an erroneous position estimation. For this reason, such reference data should always be kept secure and intact for a robust localization system. An adversarial attack often becomes more severe in the physical layer (PHY) as all nodes/anchors relay data packets sharing a common wireless medium. Thus, adversaries eavesdropping on some specific regions or compromising a few benevolent nodes can easily capture data packets relayed between anchors and/or nodes. An anchor can also be detected and tracked when attackers perform sufficient RSS/AoA measurements of its broadcast messages. Thus, the anchor reference data always remains insecure and can easily be corrupted by estimating it using trilateration/triangulation. This can also disrupt localization in some other nodes by relaying these tampered data throughout the network. Moreover, attackers can introduce errors into GPS data, producing jamming signals or noise. Although some research proposals have focused on mitigating this issue by obscuring the actual location or changing pseudonyms at appropriate intervals, these may not be as effective as anchors may be captured whenever they broadcast signals in the radio. Therefore, keeping an anchor secure on sensor fields remains a difficult task and attackers must always be kept beyond their radio coverage.

In this chapter, we have proposed a new security strategy to avoid attacks on the anchor in the physical layer. An anchor remains aware of location in the network with a GPS receiver. It is also able to find the directions of neighboring nodes with a smart antenna. The proposed strategy is based on minimizing the probabilities of successful attacks on an anchor by keeping adversaries beyond its beam coverage. Thus, each time, an optimal beam pattern is produced with the estimated AoA information to relay the data packets between the anchor and the nodes via private links. To generate the optimal pattern, the excitation coefficients of the array elements are computed using a Particle Swarm Optimization (PSO) algorithm. The security of an anchor is ensured by pattern irregularities that deteriorate the chances of data recovery for attackers, even if adequate RSS/AoA measures take place. Moreover, the anchor is assumed to send a pseudo reference to the nodes which always preserves data integrity for a localization system.

The rest of this chapter is organized as follows. In Section 6.2, an overview of attack scenarios in the physical layer is discussed. In Section 6.3, the security criterion of an anchor keeping its references confidential is explained. In Section 6.4, the proposed security strategy using antenna pattern control and pseudo references is described in detail. In Section 6.5, the results of the simulation are presented. Finally, Section 6.6 discusses the limitations and possible modifications of this strategy in the future.

6.2 Attack scenarios

The primary objective of an attack is to mislead the plans/decisions taken based on the accumulated data. As mentioned earlier, a localization system plays an important role in detecting/tracking an event in the network. Thus, an attack aims to cause any component to malfunction which eventually leads to a breakdown of overall localization systems. For this work, we considered all possible attacks on a localization system in the physical layer. Usually, such an attack can happen in two ways, such as

- Direct attack, or
- Indirect attack.

6.2.1 Direct attacks

Adversaries can establish passive links to access and control the stored data of a few benevolent nodes/anchor. Otherwise, they can alter the physical/radio environments in some specific areas and deteriorate the propagation characteristics of a few benevolent nodes/anchor therein. Such attacks (also called insider/internal attacks) can make the localization process very tricky in many ways, as follows.

- *Through compromised nodes/anchors*

For systems based on measuring RSS/ToA/TDoA, distance estimations can be erroneous by varying the transmission power or by delaying the transmission of data packets in compromised nodes [107]. Likewise, for systems based on AoA measurement, angle estimations would be incorrect when compromised nodes send signals with reduced SNR (tampering with noise). Besides, position computations can be incorrect by modifying anchor reference data.

- *Through compromised environments*

For systems based on RSS/ToA/TDoA measurement, distance estimations can also be erroneous by spreading obstacles on ground, or smoke/noise to change the physical medium of the network and modify signal propagation characteristics. Likewise, for systems based on AoA measurement, angle estimations can be erroneous by deploying magnets on the sensor fields. Also, position computations can be hampered by jamming GPS signals to cause erroneous anchor reference data [108].

6.2.2 Indirect attacks

Adversaries can install huge advanced electronic devices and circuits to eavesdrop on a few benevolent nodes/anchor and capture their IDs from messages broadcast on the network. They can now appear with such IDs and participate in the process of localization claiming legitimate nodes/anchors. During data communications, they can easily inject wrong/misleading information into nodes/anchor, causing misrouting of data throughout the network. Such attacks are often referred to as external attacks [109].

6.3 Security criterion

For precise localization of nodes, accurate anchor reference information is always necessary. Thus, the design of a suitable security framework for the mobile anchor is the main concern of this work. It keeps the anchor reference data intact against adversarial attacks in the physical layer. To achieve this, all neighboring nodes are considered suspicious.

Because there can always be a chance of existing compromised or externally implanted nodes. Thus, the anchor must control its transmission power by setting up a private link with the desired node. This is accomplished by maintaining the beam coverage area for a node according to the signal-to-noise ratio threshold (SNR_{th}) value when receiving a beacon message. Since SNR_{th} is the minimum requirement to ensure reliable data transfer, this will cause SNR to deteriorate at all adjacent nodes, keeping them always below this threshold. Moreover, a second layer of protection is provided by steering deep nulls to all adjacent nodes. Since accurate measurement of RSS/AoA data or retrieval of GPS data is quite impossible at such a low SNR level, the integrity of relayed data packets is assured. Thus, the conditions for preserving the security of an anchor are expressed (in dBm) as follows.

- For the desired node,

$$P_0 |AF_d(\theta_d)|^2 \geq SNR_{th} \quad (6.1)$$

- For all other neighboring nodes,

$$P_0 |AF_d(\theta_k)|^2 < SNR_{th} \quad (6.2)$$

where, P_0 is the effective isotropic radiated power (EIRP) at a reference distance from the anchor and AF_d is the array factor (beamforming function) of the smart antenna.

In the network, SNR gives a quantitative measure of signal strength relative to noise power. Considering an AWGN (Additive White Gaussian Noise) distribution of noise power, the SNR only varies due to variation in signal power (inverse of the square of the distance to the transmitter) in free space ($X_{\sigma\eta} = 0$ dB, $\alpha = 2$). Thus, the SNR level for the k -th node with received signal power (P_k) and distance (d_k) is defined as follows.

$$SNR_k = P_k(d_k) \quad (6.3)$$

where, P_k and d_k are described using equation (1.3) and equation (1.4). However, an attacker can locate the anchor if it eavesdrops on two or more neighboring nodes to collect RSS/AoA data. Thus, the formulation of the anchor security conditions must always consider the position computation method used. Also, these conditions may fail in various cases as defined below.

Definition 1: if the two or more neighboring nodes stay on and above the SNR_{th} in a beam pattern (triangulation).

Definition 2: if the three or more neighboring nodes stay on and above the SNR_{th} in a beam pattern (trilateration).

Definition 3: if the four or more neighboring nodes stay on and above the SNR_{th} in a beam pattern (multilateration).

Likewise, transmitting data packets with a pseudo reference for the anchor can be useful to camouflage its actual positions from attackers. However, the desired node can estimate its position, as the point of intersection of two straight-lines originating from two distinct anchor points with such pseudo reference data.

6.4 Security framework

The proposed security system is based on antenna pattern control and pseudo reference generation for the anchor. An anchor is considered to be equipped with a smart antenna. The optimal pattern must steer the main lobe to the desired node and also steer several deep nulls to neighboring nodes during data communications.

Thus, anchor security is ensured by reducing the probability of successful attacks and keeping adversaries beyond exposure to this pattern. Precise measurement of RSS/AoA data always depends on the radio propagation characteristics and physical environments of a network. Thus, the irregularities of the pattern impose each time an erroneous estimation of data. Moreover, the system provides an extra layer of protection for data integrity allowing pseudo-references for the anchor along its trajectories. Thus, this system leads to the development of robust localization for wireless sensor networks. In the proposed system, we endeavored to combine the two objectives, such as

- use an optimal pattern to block adversaries in the localization process, and
- use pseudo references for the anchor in the computation of the positions of the nodes.

6.4.1 Antenna pattern controls

For locating nodes in WSNs, it is always important to have precise information regarding distance/angle and anchor reference. The adaptive beamforming features of smart antennas can be useful for relaying them over private and point-to-point links. Adaptive beamforming means shaping an optimal pattern by steering the main lobe and nulls in the appropriate directions based on Angle of Arrival (AoA) information [110]. But, it depends on the accuracy of the AoA information. Adversarial attacks altering the transmission characteristics over a wireless medium often produce erroneous localization data (distance/angle). As a result, conventional beamforming techniques that focus on beam steering (directs the main lobe to the desired node) or null steering (places deep nulls in the interfering directions of neighboring nodes) cannot be as effective in hostile environments. To achieve a complete solution, beamforming must also ensure a lower SLL with beam steering and null steering. However, by integrating all these criteria, pattern synthesis becomes a multi-objective problem, and a powerful heuristic search engine is often needed to solve it [111]. Optimizing any of the electrical/physical parameters (e.g., element excitation, element position, or progressive phase shift) in a suitable array configuration can produce such a shaped beam pattern [112]. In this work, the PSO is used to calculate the optimal value of the excitation coefficients of the array elements in order to produce an optimal pattern.

1. Particle swarm optimization

Particle Swarm Optimization (PSO) is a population-based, stochastic, evolutionary technique introduced by Kennedy and Eberhart in 1995. It is based on a socio-behavioral model of swarm intelligence. It has a simple structure and a higher degree of convergence. It is, therefore, now a better choice for optimizing the objective of multidimensional, discontinuous, and complex problems [113, 114]. In this method, each ‘particle’ refers to an individual agent within the swarm. And the position of each particle represents a probable solution to the problem. It usually formulates a problem-specific fitness function, $f(X_0)$ (also called cost/error/objective function) which defines the level of precision of such a solution. Therefore, each particle should find its position with higher fitness value on the search space. However, without any prior knowledge, it initializes the particles with random location $[X_0]_{M \times N}$ and random velocity $[V_0]_{M \times N}$ for their motions (M and N are the numbers of particles and the dimensions of the problem, respectively). It uses the social interaction between particles during the search process. Thus, the particles must update their velocity and position based on their personal best position $[X_{pbest}]_{M \times N}$ and global best position $[X_{gbest}]_{1 \times N}$ iteratively, using the equations as follows.

$$V_0 = \kappa V_0 + c_1 rand_1() (X_{pbest} - X_0) + c_2 rand_2() (X_{gbest} - X_0) \quad (6.4)$$

and

$$X_0 = X_0 + V_0 \quad (6.5)$$

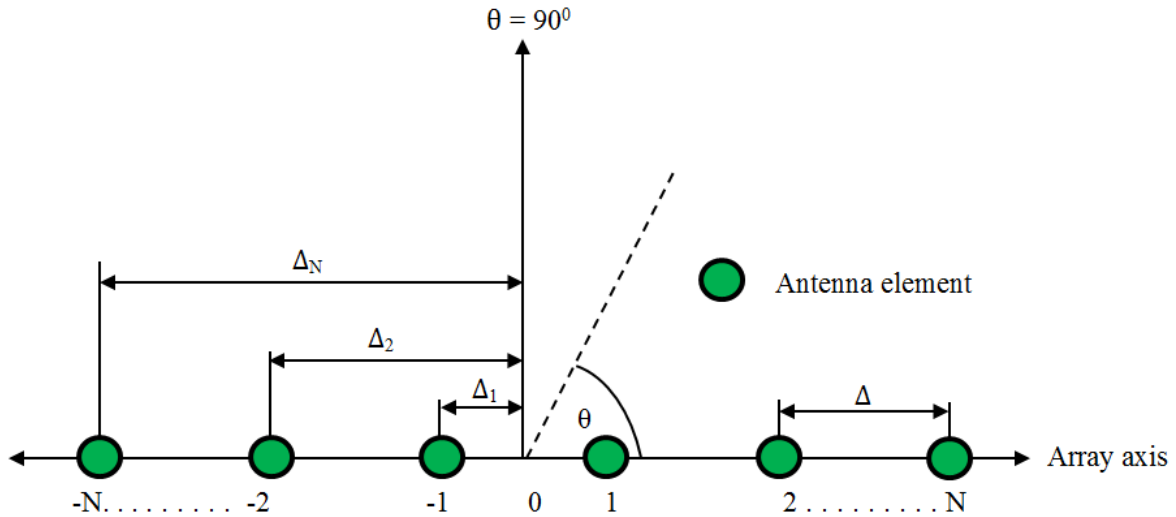


Fig. 6.1. A linear and symmetric array

Here, $rand_1()$ and $rand_2()$ are two uniform random variables in the range $\{0, 1\}$ to introduce some natural randomness. In addition, the parameters c_1 and c_2 specify the relative weights of X_{pbest} and X_{gbest} respectively. It often uses an inertia weight (κ) to improve its convergence performance by linearly reducing the particle velocity within a limit $\{0.9, 0.4\}$. It gradually emphasizes local exploitation rather than global exploration in the search process. It also applies boundary conditions to enhance and confine particle movement to the desired domain of interest $\{X_{min}, X_{max}\}$. However, the process continues until a predefined termination criterion (t_{max}) is reached. Algorithm 13 explains the standard PSO as below. Input parameters specify the set of data needed to start the process. Likewise, the desired data set obtained at the end of program execution represents output parameters.

2. PSO-based beamforming

An antenna pattern with higher directivity, narrower beamwidth and lower sidelobe level are always desirable for secure data transmission. However, a single antenna is unable to generate such a pattern and an array configuration is often used [115]. Conventional beamforming in smart antennas cannot tradeoff among these pattern attributes (e.g., directive gain, beamwidth, and SLL). On the other hand, this is achieved by optimizing one of the electrical/physical parameters of an antenna array [116]. But such a pattern alone cannot always provide security against malicious attacks. And, the flexibility of placing deep nulls in the direction of interference is crucial to extend the second layer of protection [117]. Incorporating this strategy makes the design problem very complicated. There is always a chance of deteriorating tradeoff issues. For this reason, the PSO is considered to evaluate the excitation coefficients (W) of array elements. Assuming the geometry of a linear and symmetric array (as shown in Fig. 6.1) with uniform spacing (Δ) between elements ($2N$), the normalized form of the array factor (AF) is written as follows.

$$AF_p(\theta) = \sum_{n=1}^N W_n \cos \left[\left(n - \frac{1}{2} \right) \beta \Delta (\sin \theta - \sin \theta_d) \right] \quad (6.6)$$

where, W_n denotes the excitation coefficient of the n -th element and θ_d is the direction of the main lobe.

Due to the symmetric structure, the optimization problem reduces to only half (N) of the actual/physical dimensions of the array.

Algorithm 13 Pseudo-code for particle swarm optimization

Input: Number of particles (M), Number of dimensions (N), Position vectors for particles (X_0), Velocity vectors for particles (V_0), Dynamic ranges for position vectors: $\{X_{min}, X_{max}\}$, Dynamic ranges for velocity vectors: $\{V_{min}, V_{max}\}$, Inertia weight (κ) and Maximum permissible search time (t_{max})

Output: Particle with the optimum position vector: X_{gbest}

```

1: Initialize:  $M, N, V_{min} = X_{min}, V_{max} = X_{max}, f(X_{gbest}) = \infty, \kappa = 0.9, \kappa_0 = \frac{0.5}{t_{max}}, m, n, t = 1, t_{max}$ 
2: for all particles  $m$  do
3:    $f(X_{pbest}) \leftarrow \infty$ 
4:   for all dimensions  $n$  do
5:      $X_0 \leftarrow rand() * \{X_{min}, X_{max}\}; V_0 \leftarrow rand() * \{V_{min}, V_{max}\}$ 
6:   end for
7: end for
8: while ( $t \leq t_{max}$ ) do
9:   for all particles  $m$  do
10:    if ( $f(X_0) < f(X_{pbest})$ ) then
11:       $X_{pbest} \leftarrow X_0; f(X_{pbest}) \leftarrow f(X_0)$ 
12:    end if
13:    if ( $f(X_0) < f(X_{gbest})$ ) then
14:       $X_{gbest} \leftarrow X_0; f(X_{gbest}) \leftarrow f(X_0)$ 
15:    end if
16:    for all dimensions  $n$  do
17:       $V_0 \leftarrow \kappa * V_0 + c_1 * rand_1() * (X_{pbest} - X_0) + c_2 * rand_2() * (X_{gbest} - X_0); X_0 \leftarrow X_0 + V_0$ 
18:      if ( $X_{max} < X_0 < X_{min}$ ) then
19:         $X_0 \leftarrow rand() * X_{max}$ 
20:      end if
21:      if ( $V_{max} < V_0 < V_{min}$ ) then
22:         $V_0 \leftarrow rand() * V_{max}$ 
23:      end if
24:    end for
25:  end for
26:   $\kappa \leftarrow \kappa - \kappa_0; t \leftarrow t + 1$ 
27: end while
28: return  $X_{gbest}$ 

```

Moreover, the boundary is set to a value in the range $\{0, 1\}$ for each dimension, simply giving a dynamic range of 1 in the optimization process. Now, all the necessary attributes of the desired pattern are stipulated on a reference template defined with a simple time-scaled cosine function as follows.

$$AF_d(\theta) = \begin{cases} \cos \left[(\theta - \theta_d) \frac{\pi}{FNBW} \right] & \text{if } |\theta - \theta_d| \leq \frac{FNBW}{2} \\ K & \text{if } \theta = \theta_n \\ SLL & \text{otherwise} \end{cases} \quad (6.7)$$

where, the beamwidth between the first nulls (FNBW), the sidelobe level (SLL), and the depth of the nulls (K) are three user-defined variables to synthesize the desired pattern (shown in Fig. 6.2). In order to maintain lower computational complexity, the fitness function (F)

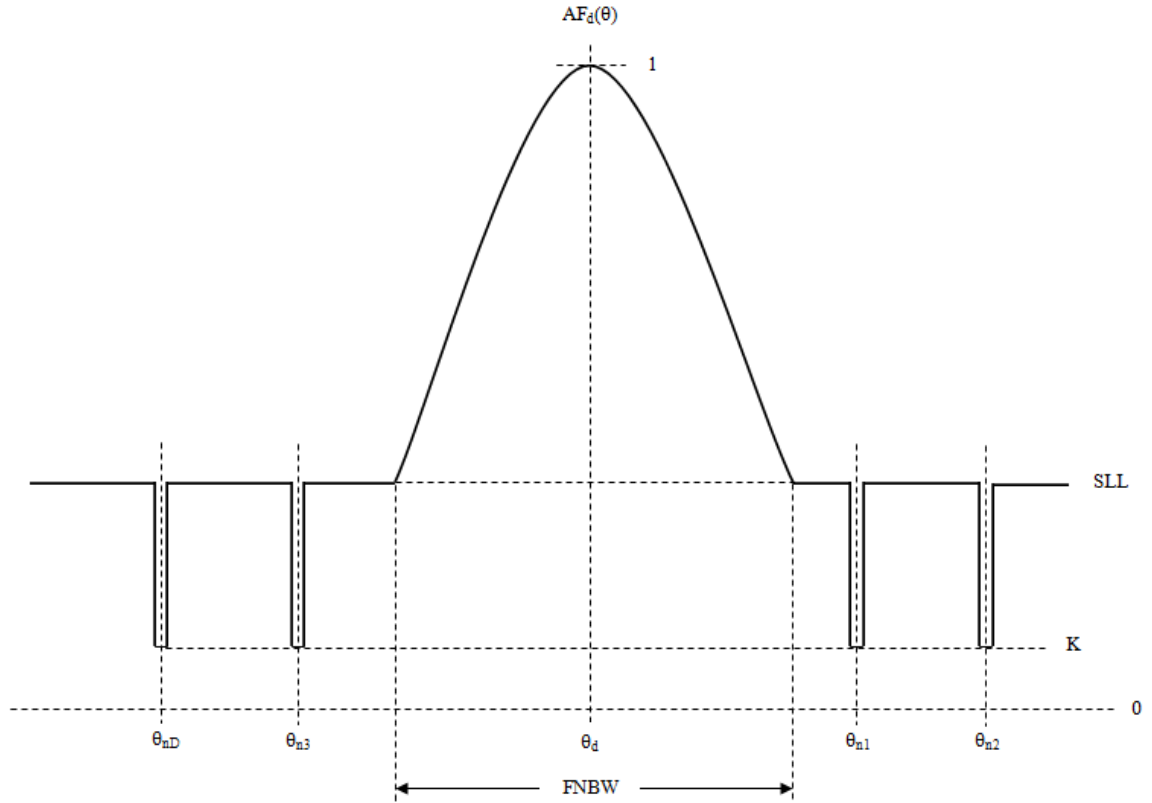


Fig. 6.2. A reference template

for the optimization process is formulated as the sum of all deviations, $\delta(\theta)$ obtained between the desired pattern, $AF_d(\theta)$ and the produced pattern, $AF_p(\theta)$. The fitness value is evaluated for any sample angle (θ) in the range $\{-90^\circ, 90^\circ\}$ with an interval of 1° and expressed as follows.

$$F = \min \left\{ \sum_{\theta=-90^\circ}^{90^\circ} \delta(\theta) \right\} \quad (6.8)$$

Each deviation is usually assigned to a penalty value of 1 whenever it violates the design goal such that $[AF_p(\theta) \leq AF_d(\theta)]$. It is also defined as follows.

$$\delta(\theta) = \begin{cases} 1 & \text{if } AF_d(\theta) - AF_p(\theta) \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.9)$$

Thus, the process always converges towards a better design (having a minimal penalty). It keeps record of the best solution ever achieved.

Algorithm 14 explains the optimal pattern generation using PSO. Here, the PSO produces an optimal pattern using the template as a reference. Accordingly, it considers them as the input and output variables. In this case, each particle has N dimensions in the solution space. Considering M particles, the PSO yields the position vector X_0 and the velocity vector V_0 as matrices of order $M \times N$. These matrices are initialized randomly for faster convergence. The fitness function is evaluated for each particle to find its best value as $f(X_{pbest}) = [f(X_{pbest1}), f(X_{pbest2}), \dots, f(X_{pbestM})]^T$ and the global best value as $f(X_{gbest}) = \min\{f(X_{pbest})\}$ in an iterative way. The $f(X_{pbest})$ and $f(X_{gbest})$ values and their respective X_{pbest} and X_{gbest} positions are recorded and used to update the velocity and position matrices. The termination condition for this process is set to a maximum iteration of 1000.

6.4.2 Pseudo reference generation

In the proposed system, the security of an anchor is preserved against direct attacks by sending a beacon message with pseudo reference information instead of its actual location on the trajectories. This can spoof attackers while benevolent nodes can autonomously estimate their locations by obtaining two such pseudo references from the anchor. As stated earlier, an anchor is aware of its own locations through a GPS receiver. It estimates distance and angle information of signals received from a few neighboring nodes with RSS exceeding SNR_{th} . These neighboring nodes (also called clustering nodes) are selected to transfer beacon messages serially from the current anchor position. The anchor reference data is very important for estimating the position of a clustering node. Therefore, the confidentiality of this data must be ensured so that the integrity of the data is maintained in the relayed beacon messages. To achieve this, pseudo forms of such reference data are prepared along the corresponding directions of each clustering node. For the k -th clustering node, the anchor pseudo reference data (x_{pk}, y_{pk}) is generated as follows.

$$\left. \begin{aligned} x_{pk} &= x_a + rand()d_k \cos \theta_k \\ y_{pk} &= y_a + rand()d_k \sin \theta_k \end{aligned} \right\} \quad (6.10)$$

where, (x_a, y_a) denotes actual position of the anchor in the current anchor point.

6.5 Performance evaluation

In this work, the security of the anchor is preserved through the control of the antenna pattern and the generation of pseudo-references. Each time pattern control aims to provide a stable link only with the desired node keeping all other neighboring nodes away from its radio coverage. On the other hand, the purpose of generating a pseudo reference is to protect the anchor, thus reducing the chances of being captured directly by attackers. The performances of this system are evaluated through a set of simulations under several benchmark functions.

6.5.1 Performance metrics

The performance of the proposed security system depends on the ability to generate the optimal pattern according to the desired attributes defined in the reference template. Thus, the accuracy and the success rate of the beamforming process are considered as the two metrics to verify its performance as follows.

- *Beamforming efficiency* (χ): It is the ability to reproduce the patterns according to the desired attributes defined in the reference template. In fact, it measures all the deviations of an optimal pattern from the desired specifications and expressed (in percentage) as follows.

$$\chi = (1 - \varepsilon_b) \times 100\% \quad (6.11)$$

Algorithm 14 Pseudo-code for generating the optimal pattern

Input: Maximum permissible search time (t_{max}), Array weight vector (W), Weight update vector (V_0), Number of vector set (M), Number of vector elements (N), Dynamic ranges of the vector elements: $\{0,1\}$, Array steering vector (\bar{a}), Dynamic ranges of the scanning angle: $\{-90^0, 90^0\}$, Desired beam pattern (AF_d) and Inertia weight (κ)

Output: Beam pattern (AF_p) with the optimum weight vector (W_{gbest})

```

1: Initialize:  $m, n, t = 1, \theta = -90^0, \kappa = 0.9, \kappa_0 = \frac{0.5}{t_{max}}, f(W_{gbest}) = \infty, M, N, t_{max}$ 
2: for all vector set  $m$  do
3:    $f(W_{pbest}) \leftarrow \infty$ 
4:   for all vector elements  $n$  do
5:      $W \leftarrow rand(); V_0 \leftarrow rand()$ 
6:   end for
7: end for
8: while ( $t \leq t_{max}$ ) do
9:   for all vector set  $m$  do
10:     $F \leftarrow 0$ 
11:    for all scanning angles  $\theta$  do
12:       $AF_p \leftarrow 0$ 
13:      for all vector elements  $n$  do
14:         $AF_p \leftarrow AF_p + W * \bar{a}$ 
15:      end for
16:      if ( $AF_p > AF_d$ ) then
17:         $\delta \leftarrow 1; F \leftarrow F + \delta$ 
18:      end if
19:    end for
20:    if ( $F < f(W_{pbest})$ ) then
21:       $W_{pbest} \leftarrow W; f(W_{pbest}) \leftarrow F$ 
22:    end if
23:    if ( $F < f(W_{gbest})$ ) then
24:       $W_{gbest} \leftarrow W; f(W_{gbest}) \leftarrow F$ 
25:    end if
26:    for all vector elements  $n$  do
27:       $V_0 \leftarrow \kappa * V_0 + c_1 * rand_1() * (W_{pbest} - W) + c_2 * rand_2() * (W_{gbest} - W)$ 
28:      if ( $1 < V_0 < 0$ ) then
29:         $V_0 \leftarrow rand()$ 
30:      end if
31:       $W \leftarrow W + V_0$ 
32:      if ( $1 < W < 0$ ) then
33:         $W \leftarrow rand()$ 
34:      end if
35:    end for
36:  end for
37:   $\kappa \leftarrow \kappa - \kappa_0; t \leftarrow t + 1$ 
38: end while
39: for all scanning angles  $\theta$  do
40:    $AF_p \leftarrow 0$ 
41:   for all vector elements  $n$  do
42:      $AF_p \leftarrow AF_p + W_{gbest} * \bar{a}$ 
43:   end for
44: end for
45: return  $AF_p$ 

```

Table 6.1: Simulation parameters

Parameters	Values
Network size	1000 m × 1000 m
Number of nodes	100, 200, 300
Number of anchor	1
Maximum communication range	100 m
Transmit power of an anchor	30 dBm
SNR threshold level	-40 dBm
Path loss exponent	2
Shadowing noise variance	1

where, ε_b denotes the error associated with the beamforming process. It is defined as a ratio of the aggregated value of all deviation weights obtained to the total number of sample points (Ω) and expressed as follows.

$$\varepsilon_b = \frac{\sum_{i=1}^{\Omega} \delta(\theta_i)}{\Omega} \quad (6.12)$$

where, θ_i is the i -th sample angle over the range $[-90^0, 90^0]$.

- *Success rate (SR)*: It measures the degree of effective attempts to preserve the security of an anchor on a particular trajectory in WSNs. Therefore, it is also defined as the ratio between the number of successful attempts (Λ_s) and the total number of attempts (Λ) made by an anchor over a specific time interval. It is expressed (in percentage) as follows.

$$SR = \frac{\Lambda_s}{\Lambda} \times 100\% \quad (6.13)$$

where, a successful attempt means not permitting more than one neighboring node (worst case) into the optimal pattern at a time.

6.5.2 Simulation environments

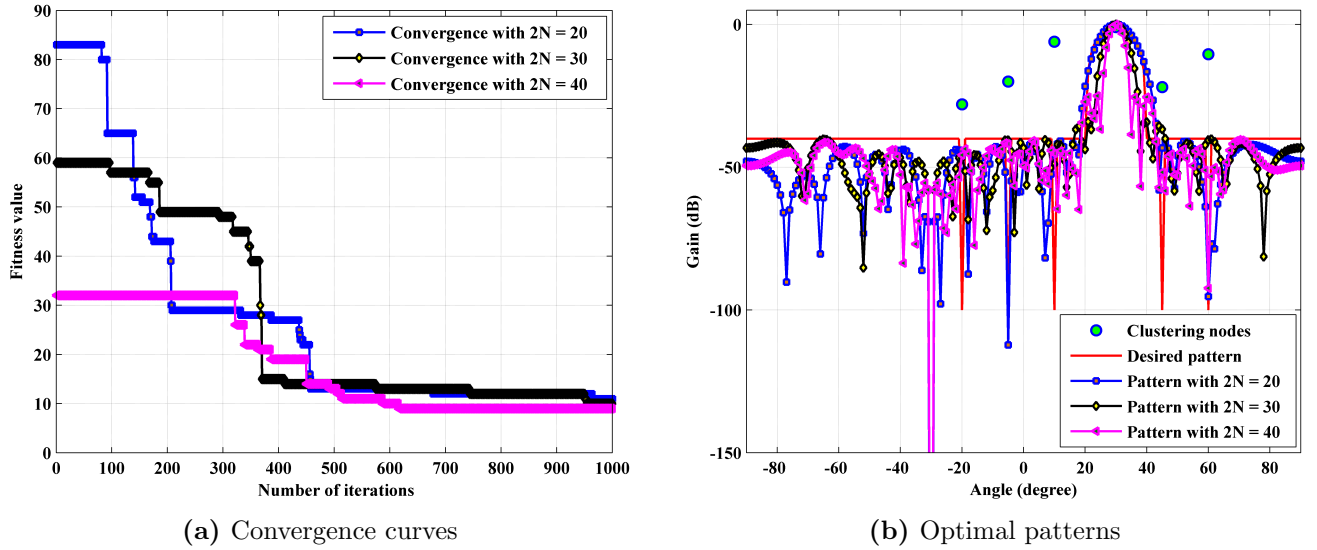
A set of simulations is carried out on the MATLAB software package (version 14). To keep an analogy with the real-time deployment scenario, several off-line data are generated on PC (Personal Computer). As a network architecture, a random deployment of 100, 200 and 300 nodes (making different node densities) over a 2-D field of 1000 m × 1000 m is considered. Nodes are supposed to be scattered with reasonable space to increase network coverage, avoiding interference between them. The anchor is also considered as moving along random trajectories on the sensor field and relaying data packets to nodes via private links. Adversaries are assumed to act either by capturing a few benevolent nodes or by directly participating in data communications in the network. As radio propagation in free space, a log-normal shadowing condition is assumed. The simulation parameters are given in Table 6.1. They are compatible with MICAz motes, using the IEEE 802.15.4 MAC protocol for common wireless narrowband transmission systems. Also, the design parameters and specifications of the desired patterns for an antenna array are given in Table 6.2. A maximum of 1000 iterations are chosen for the PSO algorithm where the variables are kept in the range of $\{0,1\}$.

6.5.3 Simulation results

The security system is tested with a set of simulations for performance metrics as mentioned above. However, the efficiency of beamforming depends on the length of the array.

Table 6.2: Antenna design parameters and pattern attributes

Parameters	Values
Array size	20, 30, 40
Frequency of operation	2.4 GHz
Spacing between elements	0.5λ
SLL	0.01 (-40 dB)
FNBW	20°
K	0.00001 (-100 dB)
θ_d	30°
θ_n	$-20^\circ, -5^\circ, 10^\circ, 45^\circ, 60^\circ$

**Fig. 6.3.** PSO-based pattern controls

And the success rate varies with the length of the array and the density of nodes. Thus, the results are presented as Empirical Cumulative Distribution Functions (ECDF) taking 30 runs of the program in all cases. Also, considering 100 nodes in the network, convergence curves and optimal patterns for different array lengths are shown in Fig. 6.3(a) and Fig. 6.3(b), respectively. It is observed that the beamforming process converges faster for a larger array length. This happens because an optimal pattern often requires a narrower beamwidth which is ensured with a larger number of array elements. The optimized values for the array element excitation coefficients are given in Table 6.3. Likewise, considering 100 nodes in the network, the beamforming efficiency for different array lengths is illustrated in Fig. 6.4. The success rate (considering only 25 anchor points in each run) under variable array lengths and node density is also shown in Fig. 6.5(a) and Fig. 6.5(b), respectively. Note that the beamforming efficiency and success rate remain higher for larger array length. The success rate also becomes higher when the density of nodes is lower in the network. This happens because a low node density always guarantees fewer neighboring nodes for an anchor. This, in turn, can ensure the least probability of compromised nodes in its radio coverage.

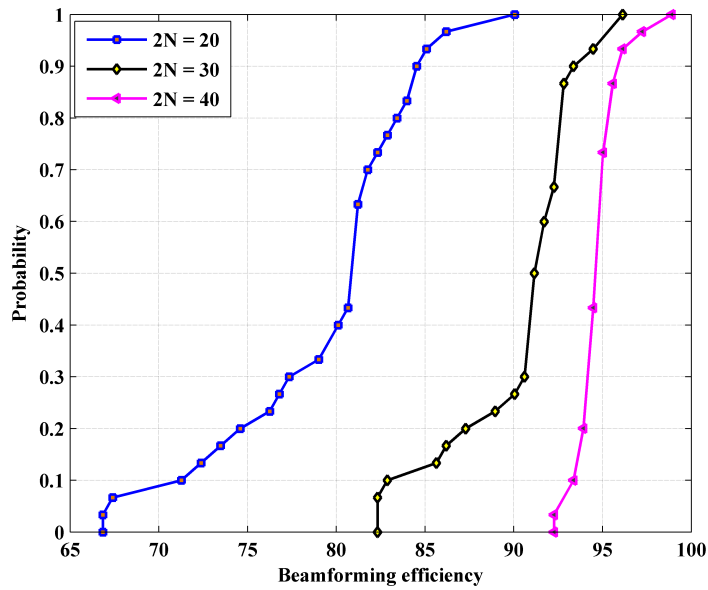
6.5.4 Performance analysis

In WSNs, the performance of a security system mainly depends on two factors, such as

- node deployment scenarios, and
- attack scenarios.

Table 6.3: Optimized element excitation coefficients ($W_{g_{best}}$)

Index (n)	Number of array elements ($2N$)		
	$N = 10$	$N = 15$	$N = 20$
1	1.0000	1.0000	1.0000
2	0.9340	0.9693	0.9873
3	0.8505	0.9307	0.9297
4	0.7358	0.8813	0.8912
5	0.6098	0.7999	0.8971
6	0.4618	0.7324	0.8106
7	0.3389	0.6745	0.7440
8	0.2083	0.5832	0.7555
9	0.1252	0.4907	0.7060
10	0.0636	0.4307	0.6783
11	...	0.3050	0.6354
12	...	0.2282	0.6038
13	...	0.1689	0.5556
14	...	0.1298	0.4760
15	...	0.0615	0.4152
16	0.3317
17	0.2565
18	0.1964
19	0.1335
20	0.0661

**Fig. 6.4.** Efficiency in pattern generation

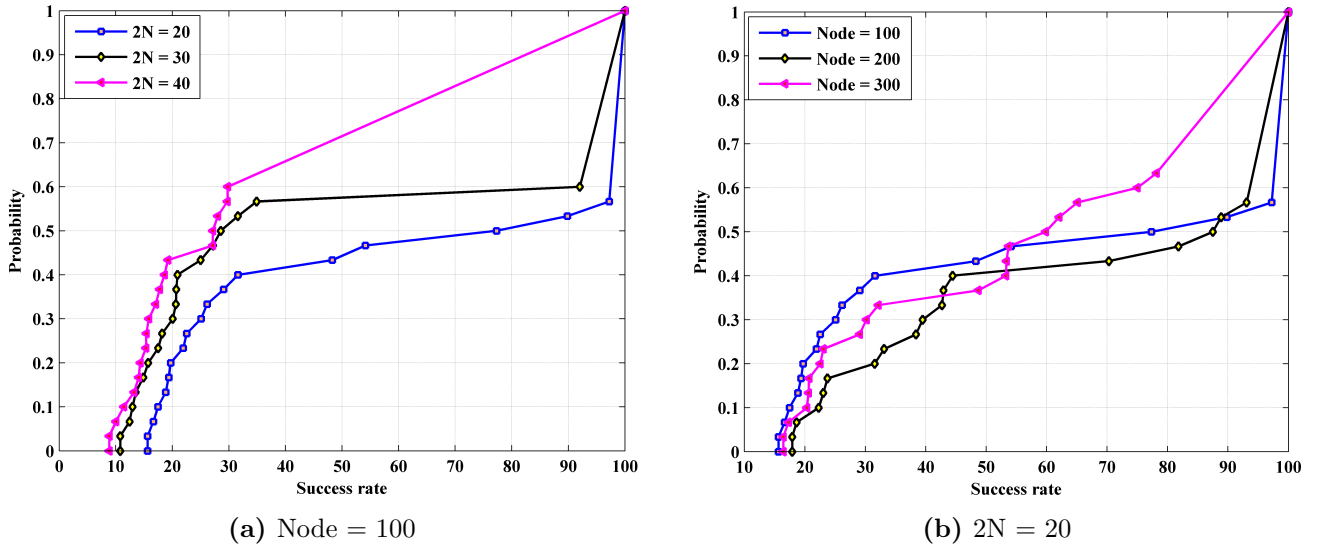


Fig. 6.5. Success rate to secure anchor data

Thus, attacks would be greater for a network with higher node density. Also, the complexity of the attack scenario must have a significant impact on the performance of a security system. An ordinary security system often overlooks to include the necessary preventive measures against such a complex attack scenario and does not perform well. The proposed security system is very effective in mitigating physical layer attacks because it considers all adjacent nodes as compromised. However, it can become fragile in the face of replay type of attacks, especially when the ratio of compromised nodes to benevolent nodes becomes higher in the network or for a very dense network. Combining a cryptographic system with it and using pairwise key distribution (when transmitting data with an anchor) may not be useful in most cases. It also remains vulnerable when there is no prior knowledge of the behaviors of a compromised node in the network. And the anchor can share its secret keys with a compromised node via a direct link. On the other hand, a compromised node prevailing on the patterns of another link can capture data packets generated by the anchor. Thus, attackers could tamper with data packets and replay them on all adjacent nodes. This would surely mislead the localization process on those particular nodes. Therefore, some alternative measures should also be taken to identify the compromised nodes and eliminate their effects on the localization process.

6.6 Summary

In this chapter, we have proposed a new security strategy for the anchor via antenna pattern control and pseudo reference generation. It can protect the privacy of the mobile anchor against physical layer attacks. Since an anchor is used as a reference node in the localization process, it is always essential to preserve its privacy in the network. In the proposed system, a data packet is securely transmitted by establishing a stable link with an optimal pattern. In most cases, such a pattern can reduce the chances of attacks on an anchor by keeping adversaries beyond its radio coverage. Moreover, pattern irregularities can complicate the data recovery process for attackers, although some remain in the links. By relaying a pseudo reference for an anchor, the confidentiality of the actual data is also preserved against direct attacks. However, this system is realized with a simple attack scenario. In fact, adversaries can often install huge sophisticated equipment/codes to decrypt a relayed data packet and gain access to its contents. Thus, it is still possible to verify the operational feasibility of this system in more complex attack scenarios in the future.

Chapter 7

Implementation of a smart antenna beamformer architecture on FPGA

7.1 Introduction

As discussed earlier, adaptive beamforming in a smart antenna can be effective for precise localization of nodes in wireless sensor networks. In an erratic channel, a link with possibly a narrower beamwidth is often suitable to send data for the nodes. It is also possible to cancel interference from other nearby nodes by steering deep nulls and keeping the level of the sidelobes lower in the pattern. However, the beamformer architecture must be reconfigurable to generate the pattern coping with the radio irregularities [118]. To implement such a signal processor, a digital platform is often preferred as it avoids the requirement of analog phase shifters [119, 120]. Although several platforms exist in the literature, the design technology based on FPGA (Field Programmable Gate Array) is now the best suited for a wireless sensor network infrastructure [121]. Since a smart antenna typically operates at a higher frequency (on the order of a few GHz), it needs a high-speed parallel processor to compute the array weight vector for such applications [122]. FPGA families can use their reconfigurable architectures to generate beams of different shapes depending on the specific needs of the data routing operation [123]. Moreover, it can address resource constraints requiring relatively minimal hardware. However, only a few research proposals have been found so far that implement beamformers in smart antennas based on conventional techniques [124, 125, 126]. They cannot suppress noise/interference when transmitting data on an erratic channel and incorrect distance/angle estimation extends coverage to attackers/interferers.

In this chapter, we have proposed the custom design of a beamformer architecture for the smart antenna using the PSO. A linear symmetric array with non-uniform element excitations is considered. Using the PSO, the weight vector containing the excitation coefficients of the array elements is iteratively optimized. It always generates a pattern according to the desired attributes stipulated on a reference template, described by the parabolic function. Using the FSM (Finite State Machine with Datapath) concept and the appropriate CORDIC (COordinate Rotation DIgital Computer) blocks, the beamformer architecture is implemented on an FPGA chip. It is robust to maintain data security, having the flexibility to control beamwidth, null positioning, and SLL in the pattern. It also includes benefits such as less hardware, less power, and minimal computational overheads needed for WSN infrastructures.

The rest of this chapter is organized as follows. In Section 7.2, an overview of the FPGA architecture and the CORDIC method is explained. In Section 7.3, the design principle of the proposed beamformer architecture is discussed. In Section 7.4, the hardware implementation on FPGA is described. In Section 7.5, the results of the simulation are presented. Finally, Section 7.6 discusses the limitations and possible modifications of this system in the future.

7.2 Design preliminaries

In the real-time WSN scenario, processors using traditional Von Neumann architectures with several Multiplier-Accumulator (MAC) stages may not be suitable for processing huge accumulated data. They also consume enormous power because their hardware is not customized for specific applications. Considering all the constraints of nodes like low power and volume, custom processors/Application Specific Integrated Circuits (ASICs) would be a good choice. The recent growth of microelectronics and digital technology makes FPGA families as flexible platforms for the development of digital signal processors.

7.2.1 Field programmable gate array

An FPGA architecture consists of an array of Configurable Logic Blocks (CLBs) interconnected by programmable routing resources [127]. CLBs generally act as basic building blocks, providing the basic logic gates and storage capacity to implement various user-defined logic functions. They usually consist of Basic Logic Elements (BLEs) having Flip-flops (FFs) and Lookup Tables (LUTs) along with an input/output Multiplexer (MUX) unit as general routing resources. At each periphery, they are also surrounded by Input/Output Blocks (IOBs) that connect LUTs or FFs via programmable interconnect networks to initiate interfacing with external devices. The k -input LUTs are generally able to implement any k -input Boolean function by taking 2^k number of configuration bits stored in any block of static memory (SRAM) arranged on both sides of an FPGA architecture. Typically, four-input LUTs constitute a slice, and each CLB contains two slices that implement logic gates or small memories that control the connectivity of internal resource with internal MUX units. Routing interconnects involve wires and programmable switches to establish a connection. Internal MUX units frequently control connectivity between internal resources so that they can be reconfigured infinite times to implement new logical functions. However, the re-configurability in a pre-fabricated FPGA structure regarding the optimized circuit area, higher computational speed and lower dynamic power consumption, mainly depends on the nature of the underlying programming technology associated with it [128, 129]. Thus, an effective programming technology must contain all the attributes such as re-programmability, non-volatility and use of the standard CMOS process. In fact, one of the popular technologies based on static memory (SRAM), flash memory (EEPROM) or anti-fuse, does not only have all these features. Using the standard CMOS process, the first one appears to dominate the other two technologies. However, the topology of routing resources and logic blocks in the FPGA architecture, known as floor planning, categorizes it in two types, such as hierarchical/tree based and island-style/mesh based [130]. The latter one is more common for academic applications. In this work, Xilinx Virtex-5 XC5VFX30T with ISE Design Suite 12.3 was used as the target FPGA device and hence, its architecture is shown in Fig. 7.1 for a vivid understanding of designing FPGA-based digital beamformers.

7.2.2 Coordinate rotation digital computer

CORDIC is a computationally efficient technique introduced by J. E. Volder in 1959. It can compute many mathematical functions such as real/complex, trigonometric, and hyperbolic using binary arithmetic. It is based on rotating a vector in 2-D coordinates (x, y) for a desired angle (θ) , as shown in Fig. 7.2. For different coordinates, it can have a unified form to add only one parameter (μ_0) . It is also possible to use identical hardware for various signal processing applications by changing only the initial conditions [131]. Thus, it has now become an ideal solution for FPGA based flexible design technology. In this method, the rotation vector rotates around the desired angle (θ) in a specific coordinate system. Thus, the final coordinates (x_n, y_n) are computed by multiplying the initial coordinates (x_0, y_0) by a rotation matrix.

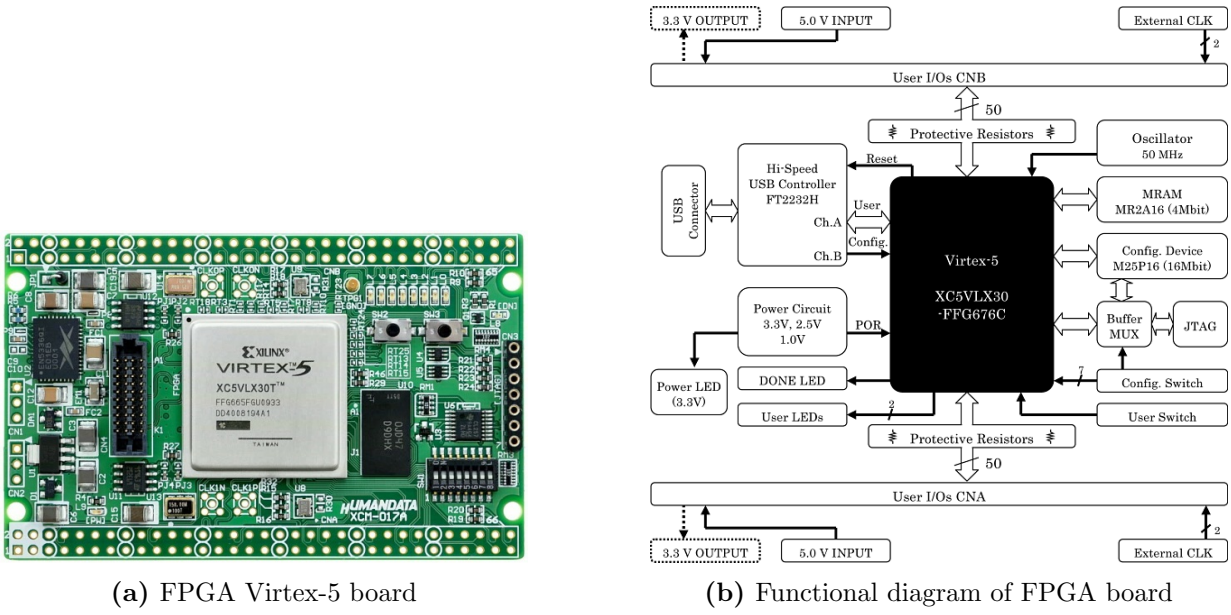


Fig. 7.1. FPGA Virtex-5 board and its functionality

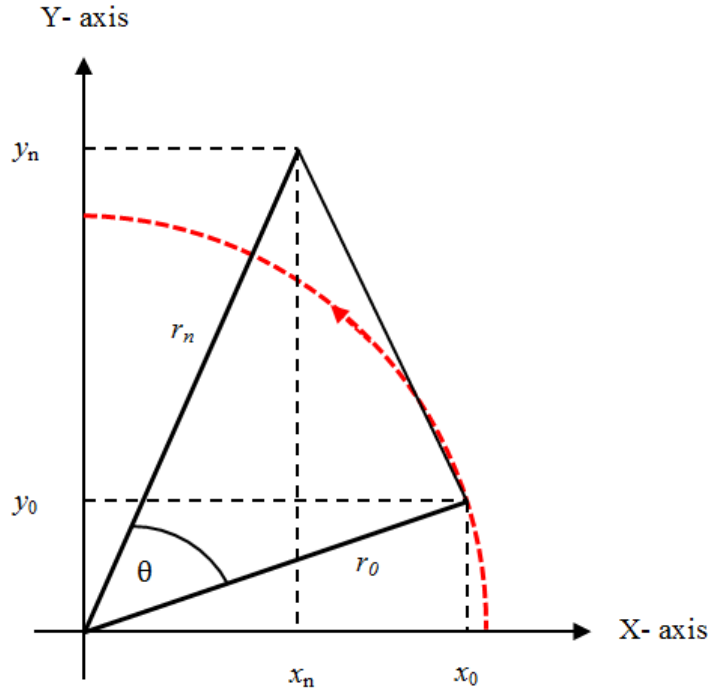


Fig. 7.2. Rotating a vector in a 2-D plane

It is usually expressed (in matrix form) as follows.

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (7.1)$$

where, $R' = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$ denotes the rotation matrix.

Now, the rotation matrix can also be represented as follows.

$$R' = (\cos \theta) \begin{bmatrix} 1 & \tan \theta \\ -\tan \theta & 1 \end{bmatrix} \quad (7.2)$$

The angle of rotation (θ) is decomposed into a set of elementary angles (θ_i) that can be obtained by pseudo-micro-rotations. Thus, the angle is computed as the sum of a few constituent angles iteratively, as follows.

$$\theta = \sum_{i=0}^{p'-1} \sigma_i \theta_i \quad (7.3)$$

where, σ_i denotes the direction of the micro-rotations.

It turns into a series of simple binary arithmetic with *shift* and *add* operations to assign the constituent angles to the power of 2. Considering the elementary angles as $\theta_i = \tan^{-1}(2^{-i})$ for the i -th micro-rotations, the rotation matrix can be represented as follows.

$$R'_i = \zeta_i \begin{bmatrix} 1 & 2^{-i} \\ -2^{-i} & 1 \end{bmatrix} \quad (7.4)$$

Due to the cosine term ($\cos \theta$), the rotation vector is scaled by a factor (ζ) through the iterations, which takes form as follows.

$$\zeta = \prod_{i=0}^{p'-1} \cos \theta_i = \prod_{i=0}^{p'-1} \frac{1}{\sqrt{1 + 2^{-2i}}} \quad (7.5)$$

The scale factor converges to 1.6467605 and 0.8281 in a circular and hyperbolic coordinate system, respectively. This can be avoided by initializing the rotation vector with a gain factor ($A' = \frac{1}{\zeta}$).

Thus, cosine/sine function is computed simply initializing the rotation vector respectively as $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} A' \\ 0 \end{bmatrix}$ or $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ A' \end{bmatrix}$ in circular coordinates.

The CORDIC usually operates in two different modes, such as

- Rotation Mode (RM), and
- Vectoring Mode (VM).

The mode of operation is characterized by directions of the micro-rotations (σ_i) and initial coordinates of the rotation vector (x_0, y_0) given as follows.

$$\sigma_i = \begin{cases} \text{sign}(z_i) & \text{for rotation mode} \\ -\text{sign}(y_i) & \text{for vectoring mode} \end{cases} \quad (7.6)$$

where, $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1 \\ y_0 \end{bmatrix}$ is set for rotation mode and vectoring mode, respectively.

Also, z or y acts as an accumulator for storing the angle of the micro-rotations in these two modes. And, direction of micro-rotations depends on the sign of z_i or y_i which is represented as follows.

$$\sigma_i = \begin{cases} +1 & \text{if } z_i \geq 0 \text{ or } y_i < 0 \\ -1 & \text{otherwise} \end{cases} \quad (7.7)$$

The CORDIC is used for a circular, linear, or hyperbolic coordinate system by choosing $\mu_0 = 1, 0$ or -1 and $\theta_i = \tan^{-1}(2^{-i}), 2^{-i}$ or $\tanh^{-1}(2^{-i})$, respectively.

The generalized forms of CORDIC equations are summarized as follows.

$$x_{i+1} = x_i + \mu_0 \sigma_i 2^{-i} y_i \quad (7.8)$$

$$y_{i+1} = y_i + \sigma_i 2^{-i} x_i \quad (7.9)$$

$$z_{i+1} = z_i + \sigma_i \theta_i \quad (7.10)$$

The iterative process continues until a predefined termination criterion (i_{max}) is reached. Algorithm 15 explains the standard CORDIC in a circular coordinate system to compute the sine/cosine function below. It avoids the scale factor by initializing the rotation vector to $x_0 = 0.61, y_0 = 0$. It uses coordinates, rotation angles, and iterations to obtain an acceptable value for the sine and cosine functions. Therefore, they are considered as the input and output variables, respectively.

Algorithm 15 Pseudo-code for realizing the sine/cosine function in CORDIC

Input: X-coordinate of rotation vector (x), Y-coordinate of rotation vector (y), Accumulator for storing angle (z), Angle of rotation (θ) and Maximum permissible iterations (i_{max})

Output: Sine function ($\sin \theta$): $-y$; Cosine function ($\cos \theta$): x

```

1: Initialize:  $i = 0, x = 0.61, y = 0, z = \theta, i_{max}$ 
2: for all iterations  $i$  do
3:   if ( $z \geq 0$ ) then
4:      $\sigma \leftarrow 1$ 
5:   else
6:      $\sigma \leftarrow -1$ 
7:   end if
8:    $x \leftarrow x + \sigma * 2^{-i} * y; y \leftarrow y - \sigma * 2^{-i} * x; z \leftarrow z - \sigma * \tan^{-1}(2^{-i})$ 
9: end for
10: return  $-y; x$ 

```

7.3 Design methodology

As mentioned in the previous chapter, antenna pattern control is very useful in mitigating malicious attacks at the physical layer. Thus, suitable hardware for beamforming in a smart antenna is also required to implement a robust localization system. We considered the same network architecture and the same protocols where the smart antennas are integrated with a few mobile anchors. Smart antennas are used to estimate Angle of Arrival (AoA) information from signals (s_j) received from neighboring nodes and transmit beacon messages serially over point-to-point links to those nodes. For secure data transmission, the antenna pattern should be highly directional with a narrower beamwidth and lower sidelobe level. In hostile radio environments, this is the primary beamforming requirement. However, the flexibility of placing deep nulls in the interference direction can extend the second layer of protection against malicious attacks. Also, the beamforming architecture must be reconfigurable, generating a new pattern from time to time as needed on the same antenna configuration.

In this work, a reconfigurable architecture for a smart antenna beamformer is realized with CORDIC and PSO. For each node in a cluster, it operates in three successive phases, such as

- the formation of an appropriate reference template based on the AoA information,
- creating an array steering vector using the CORDIC block, and
- generation of an optimal pattern using the PSO module which optimizes the excitation coefficients of the antenna elements in a linear array.

Fig. 7.3 illustrates such a smart antenna beamforming architecture. The detailed description of the proposed design methodology is given below.

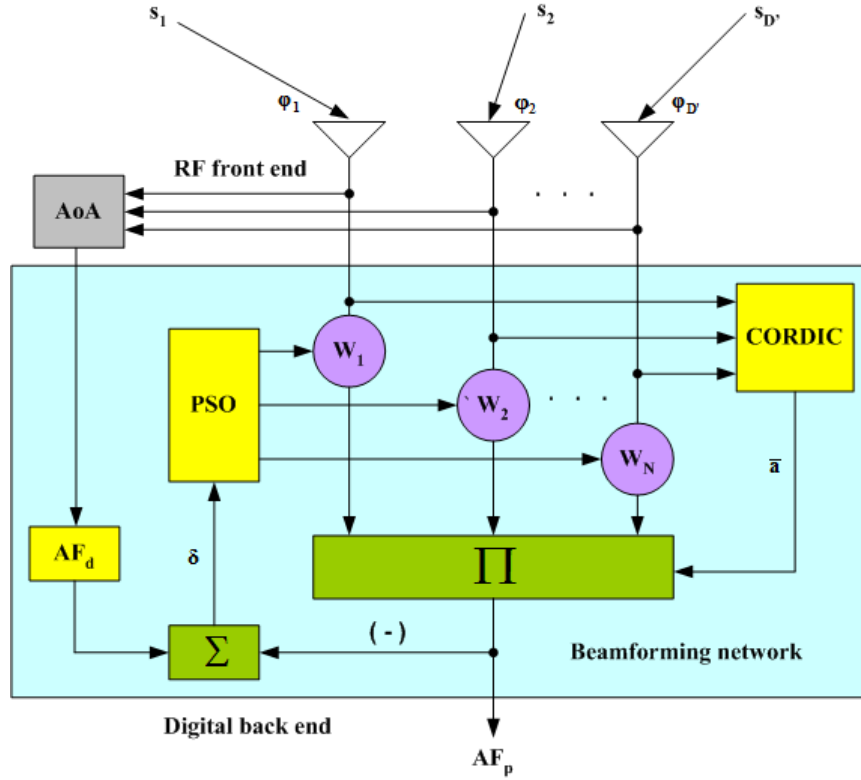


Fig. 7.3. Smart antenna beamforming system

7.3.1 Reference template

In WSNs, it is always possible to keep all adjacent nodes out of radio links by placing deep nulls in their directions. Otherwise, all nearby malicious/compromised node can eavesdrop to access localization data. Likewise, maintaining a lower SLL and narrower beamwidth would be effective, while precise estimation of localization data remains difficult in a compromised radio environment. This can improve gain and directivity to avoid packet drop (data loss) issues. A reference template must include these desired pattern attributes. Considering all neighboring nodes, each time a link is established with one, assuming the others are the interferers. A reference template contains its AoA data as the desired direction (θ_d) for beam steering. And the rest of the AoA data is selected as directions (θ_n) for the null positioning. Algorithm 16 explains the construction of a reference template based on the AoA information of the nodes. Since the generation of a reference template requires several data of the received signal and specifications for the design of the antenna pattern, they are considered as input variables. Likewise, the desired attributes of an antenna pattern are considered as output variables. In this work, a parabolic function defines the reference template (AF_d) which is expressed as follows.

$$AF_d(\theta) = \begin{cases} 1 - \left\{ \frac{2(\theta - \theta_d)}{FNBW} \right\}^2 & \text{if } |\theta - \theta_d| \leq \frac{FNBW}{2} \\ K & \text{if } \theta = \theta_n \\ SLL & \text{otherwise} \end{cases} \quad (7.11)$$

Where, the beamwidth between the first nulls ($FNBW$), the level of the sidelobes (SLL), and the depth of the nulls (K) are three user-defined variables to synthesize the desired pattern.

7.3.2 Array steering vector

Considering the geometry of a linear and symmetrical array with a uniform spacing (Δ) between the elements ($2N$), the beamforming function (AF_p) can be written by modifying equation (6.6).

It is expressed as follows.

$$AF_p = \sum_{n=1}^N W_n \cos \left[\left(n - \frac{1}{2} \right) \psi \right] \quad (7.12)$$

where, $\psi = \beta\Delta(\sin\theta - \sin\theta_d)$ is the progressive phase shift component in a phased array that produces the pattern in the direction, θ_d . Also, β is the wave number. W_n denotes the excitation coefficient of the n -th element. However, in matrix form, this equation can also be represented as follows.

$$AF_p = W^T \bar{a}(\theta) \quad (7.13)$$

where, $[W]_{N \times 1}$ and $[\bar{a}(\theta)]_{N \times 1}$ are respectively the weight vector and the array steering vector of the smart antenna. The steering vector consists of sine and cosine functions. Therefore, a suitable CORDIC block can be used to evaluate it. Algorithm 17 explains the making of an array steering vector for this beamforming system. Since this involves CORDIC and antenna design parameters to generate an array steering vector, they are considered the input and output variables.

Algorithm 16 Pseudo-code to generate the reference template

Input: AoA of received signals (φ), Number of signals received (D'), Desired signal direction (θ_d), Directions of interfering signals (θ_n), Dynamic ranges of scanning angle (θ): $\{-90^0, 90^0\}$, Beamwidth between the first nulls ($FNBW$), Sidelobe level (SLL), Depth of nulls (K)

Output: Beam pattern with desired attributes (AF_d)

```

1: Initialize:  $j, k = 1, \theta = -90^0, K, D', SLL, FNBW$ 
2: for all received signals  $j$  do
3:    $\theta_d \leftarrow \varphi[1]$ 
4:   for all scanning angles  $\theta$  do
5:     if  $(|\theta - \theta_d| \leq \frac{FNBW}{2})$  then
6:        $AF_d \leftarrow 1 - \left\{ 2 * \frac{(\theta - \theta_d)}{FNBW} \right\}^2$ 
7:     else
8:        $AF_d \leftarrow SLL$ 
9:     end if
10:    for all interfering signals  $k$  do
11:      if  $(k \leq D' - 1)$  then
12:         $\theta_n \leftarrow \varphi[k + 1]$ 
13:      end if
14:      if  $(\theta = \theta_n)$  then
15:         $AF_d \leftarrow K$ 
16:      end if
17:    end for
18:  end for
19: return  $AF_d$ 
20:  for all interfering signals  $k$  do
21:    if  $(k \leq D' - 1)$  then
22:       $\varphi[k] \leftarrow \varphi[k + 1]$ 
23:    end if
24:  end for
25:   $\varphi[D'] \leftarrow \theta_d$ 
26: end for

```

Algorithm 17 Pseudo-code for constructing an array steering vector

Input: Maximum permissible iterations (i_{max}), Desired signal direction (θ_d), Number of symmetric array elements (N), Spacing between elements (Δ), Wave number (β), Dynamic ranges of scanning angle (θ): $\{-90^0, 90^0\}$

Output: Array steering vector (\bar{a})

```

1: Initialize:  $\beta = \frac{2\pi}{\lambda}, \Delta = \frac{\lambda}{2}, n = 1, i = 0, \theta = -90^0, N, i_{max}$ 
2:  $z \leftarrow \theta_d; Flag \leftarrow 0; x \leftarrow 0.61; y \leftarrow 0$ 
3: for all permissible iterations  $i$  do
4:   if ( $z \geq 0$ ) then
5:      $\sigma \leftarrow 1$ 
6:   else
7:      $\sigma \leftarrow -1$ 
8:   end if
9:    $x \leftarrow x + \sigma * 2^{-i} * y; y \leftarrow y - \sigma * 2^{-i} * x; z \leftarrow z - \sigma * \tan^{-1}(2^{-i})$ 
10: end for
11:  $\sin \theta_d \leftarrow -y; \psi_0 \leftarrow \beta * \Delta * \sin \theta_d; Flag \leftarrow 1$ 
12: for all scanning angles  $\theta$  do
13:    $z \leftarrow \theta; \sin \theta \leftarrow -y; \psi_1 \leftarrow \beta * \Delta * \sin \theta; \psi \leftarrow \psi_1 - \psi_0; Flag \leftarrow 2$ 
14:   for all vector elements  $n$  do
15:      $z \leftarrow (n - \frac{1}{2}) * \psi; \cos [(n - \frac{1}{2}) * \psi] \leftarrow x; \bar{a} \leftarrow \cos [(n - \frac{1}{2}) * \psi]$ 
16:   end for
17:    $Flag \leftarrow 1$ 
18: end for
19: return  $\bar{a}$ 

```

7.3.3 Optimal pattern

For pattern optimization, a well-known stochastic process called PSO is used in the beamforming system. The array element excitation coefficients (N) are iteratively updated by the PSO. By choosing a symmetric structure, the optimization problem is reduced to half of the actual/physical dimensions of the array ($2N$). Also, the boundary limit is set to a value in the range $\{0,1\}$ for each vector element, simply giving a dynamic range of 1 in the optimization process. The fitness function (F) is formulated as the sum of all deviations (δ) obtained between the desired pattern (AF_d) and the produced pattern (AF_p). It is expressed in equation (6.8) which is re-written as follows.

$$F = \min \left\{ \sum_{\forall \theta} \delta(\theta) \right\} \quad (7.14)$$

Thus, the fitness value is evaluated for scanning angles ranging from -90^0 to 90^0 with an interval of 1^0 . Each deviation is usually assigned a penalty value of 1 each time it violates the design goal ($AF_p \leq AF_d$). This is also defined in the equation (6.9) which is re-written as follows.

$$\delta = \begin{cases} 1 & \text{if } AF_d - AF_p < 0 \\ 0 & \text{otherwise} \end{cases} \quad (7.15)$$

Thus, the process always converges towards a better design (having a minimal penalty). It keeps record of the best solution ever achieved. Algorithm 18 explains the optimal pattern generation of this beamforming system. Here, the PSO is used to produce an optimal pattern by taking the template as a reference. Accordingly, the input and output variables are considered with their attributes.

Algorithm 18 Pseudo-code for generating the optimal pattern

Input: Maximum permissible search time (t_{max}), Array weight vector (W), Weight update vector (V_0), Number of vector set (M), Number of vector elements (N), Dynamic ranges of the vector elements: $\{0,1\}$, Array steering vector (\bar{a}), Dynamic ranges of the scanning angle: $\{-90^0, 90^0\}$, Desired beam pattern (AF_d) and Inertia weight (κ)

Output: Beam pattern (AF_p) with the optimum weight vector (W_{gbest})

```

1: Initialize:  $m, n, t = 1, \theta = -90^0, \kappa = 0.9, \kappa_0 = \frac{0.5}{t_{max}}, f(W_{gbest}) = \infty, M, N, t_{max}$ 
2: for all vector set  $m$  do
3:    $f(W_{pbest}) \leftarrow \infty$ 
4:   for all vector elements  $n$  do
5:      $W \leftarrow rand(); V_0 \leftarrow rand()$ 
6:   end for
7: end for
8: while ( $t \leq t_{max}$ ) do
9:   for all vector set  $m$  do
10:     $F \leftarrow 0$ 
11:    for all scanning angles  $\theta$  do
12:       $AF_p \leftarrow 0$ 
13:      for all vector elements  $n$  do
14:         $AF_p \leftarrow AF_p + W * \bar{a}$ 
15:      end for
16:      if ( $AF_p > AF_d$ ) then
17:         $\delta \leftarrow 1; F \leftarrow F + \delta$ 
18:      end if
19:    end for
20:    if ( $F < f(W_{pbest})$ ) then
21:       $W_{pbest} \leftarrow W; f(W_{pbest}) \leftarrow F$ 
22:    end if
23:    if ( $F < f(W_{gbest})$ ) then
24:       $W_{gbest} \leftarrow W; f(W_{gbest}) \leftarrow F$ 
25:    end if
26:    for all vector elements  $n$  do
27:       $V_0 \leftarrow \kappa * V_0 + c_1 * rand_1() * (W_{pbest} - W) + c_2 * rand_2() * (W_{gbest} - W)$ 
28:      if ( $1 < V_0 < 0$ ) then
29:         $V_0 \leftarrow rand()$ 
30:      end if
31:       $W \leftarrow W + V_0$ 
32:      if ( $1 < W < 0$ ) then
33:         $W \leftarrow rand()$ 
34:      end if
35:    end for
36:  end for
37:   $\kappa \leftarrow \kappa - \kappa_0; t \leftarrow t + 1$ 
38: end while
39: for all scanning angles  $\theta$  do
40:    $AF_p \leftarrow 0$ 
41:   for all vector elements  $n$  do
42:      $AF_p \leftarrow AF_p + W_{gbest} * \bar{a}$ 
43:   end for
44: end for
45: return  $AF_p$ 

```

7.4 System implementation

The beamforming system is implemented on an FPGA device. The FPGA enables parallel processing in practice. Unlike conventional processors, it always develops a reconfigurable and distributed arithmetic structure using the internal logic blocks. It avoids instruction fetch and data load/store bottlenecks at runtime. Thus, it is now the most popular and flexible design solution in the digital platform [132]. The digital beamforming architecture consists of two functional blocks, such as

- control unit, and
- data unit.

A digital beamforming system can be developed using Finite State Machine with Datapath (FSMD) design methodology. The Finite State Machine (FSM) acts as a controller, and the datapath performs digital signal (data) processing operations in this system. It translates the sequence of operations (statements) of the beamforming algorithm into states and represents them by state diagrams. A state diagram includes several states and arcs that describe the operational flow through a series of commands and actions specified by arithmetic expressions. Such expressions involve external inputs, outputs, and other variables used in the algorithm. In this work, we realized the functionality of the beamforming processor with three separate modules corresponding to the reference template, CORDIC, and PSO, as in Fig. 7.3. According to the AoA information, the template module first produces a suitable pattern, AF_d to use as a reference. Then, based on the antenna design parameters (N, β, Δ , and θ_d), the CORDIC block forms the array steering vector, \bar{a} . Finally, by comparing each candidate design with the reference, the PSO module evaluates the fitness function, F and generates the optimal pattern, AF_p with the optimized weight vector, $W_{g_{best}}$ after the prescribed iterations, t_{max} . A detailed description of the FSMD design method and beamforming architecture is given below.

7.4.1 Finite state machine with datapath

The datapath is a collection of registers (store interim data), data processing units (compute arithmetic expressions), and routing networks (transfer data via buses). It performs a specific operation based on the command signal from the FSM and produces the internal status signal (flag). The FSM has a state register, input and output logic for a state transition. It uses the external commands and the datapath status signal as input. Thus, it generates the control signals to specify the datapath operation. Also, it can generate an external status signal (reset/halt) which indicates the operational status of the system.

Here, we used three distinct FSMDs to describe the behavioral model of the proposed beamforming algorithm. Fig. 7.4 illustrates the state diagram for generating a reference template based on AoA information. The first data, $\phi[1]$ in the register containing the AoA information is selected as the direction of the orientation of the main lobe (θ_d) and the others as the directions of placement of the deep nulls (θ_n). The register, ϕ is shifted left by one data position in each case of a new template generation. The process repeats until the last data comes first. Therefore, the number of reference templates produced is equal to the number of data (D') available in the register at any given time. The template is also defined by a specific SLL , null depth (K), and $FNBW$. A parabolic function characterizes the main lobe in this work.

Fig. 7.5 shows the state diagram for creating an array steering vector. Three status signals (flag) are used to control the computation of the sine and cosine functions on the CORDIC block. Thus, the trigonometric sine functions, $\sin \theta_d$ and $\sin \theta$, are computed by setting the register, $Flag$ to a value of 0 and 1, respectively. The array steering vector described by the trigonometric cosine function, $\cos \left[\left(n - \frac{1}{2} \right) \psi \right]$ is further computed using the array design parameters (β, Δ , and N) when the contents of the register, $Flag$ becomes 2.

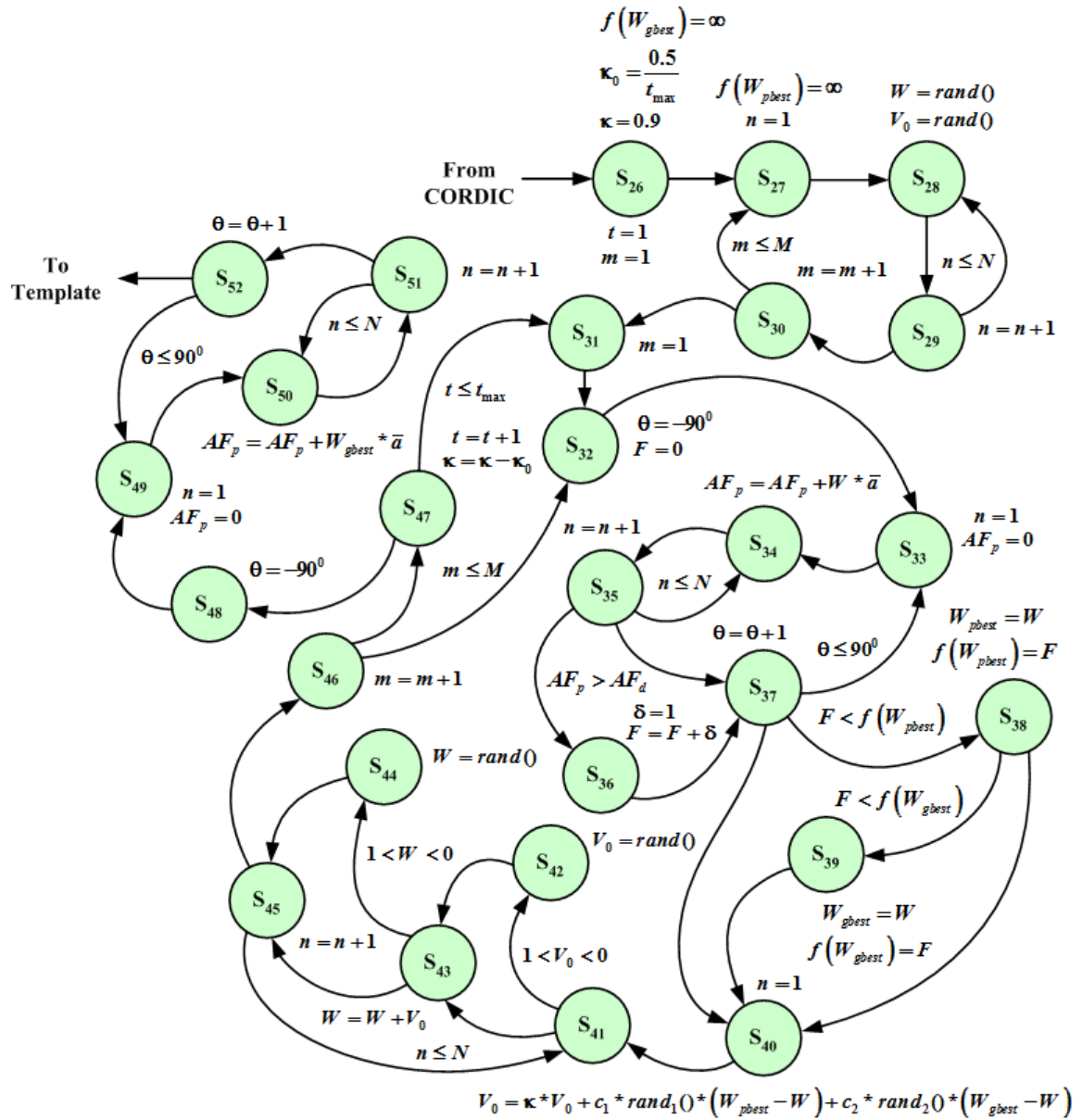


Fig. 7.6. State diagram of the PSO module

Table 7.1: Beamforming processor behaviors in the FSM

State	Command	Action
S_0	Reset/Start	Activate counter j
S_1	Clock, $j \leq D'$	Activate counter θ ; Move first content of register φ to register θ_d
S_2	Clock, $\theta \leq 90^\circ$	Subtract register θ_d from register θ ; Compare difference with $\frac{FNBW}{2}$
S_3	$ \theta - \theta_d \leq \frac{FNBW}{2}$	Set register AF_d to $1 - \left\{ \frac{2(\theta - \theta_d)}{FNBW} \right\}^2$
S_4	$ \theta - \theta_d > \frac{FNBW}{2}$	Set register AF_d to SLL
S_5	Clock	Activate counter k
S_6	Clock, $k \leq D' - 1$	Move rest contents of register φ to register θ_n one by one
S_7	$\theta = \theta_n$	Set register AF_d to K
S_8	$\theta \neq \theta_n$	Update counter k

Table 7.1: Beamforming processor behaviors in the FSM

State	Command	Action
S_9	Clock, $k > D' - 1$	Update counter θ
S_{10}	$\theta > 90^0$	Set <i>Flag</i> register to 0; Move register θ_d to accumulator z
S_{11}	Clock	Activate counter i ; Set register x to 0.61; Set register y to 0
S_{12}	$z \geq 0$	Set register σ to 1
S_{13}	$z < 0$	Set register σ to -1
S_{14}	Clock, $i \leq i_{max}$	Update register x, y and z
S_{15}	Clock	Update counter i
S_{16}	$i > i_{max}$	Check <i>Flag</i> register content
S_{17}	<i>Flag</i> = 0	Load register ψ_0 ; Set <i>Flag</i> register to 1
S_{18}	Clock	Activate counter θ
S_{19}	Clock, $\theta \leq 90^0$	Move register θ to accumulator z
S_{20}	<i>Flag</i> = 1	Load register ψ_1 ; Subtract register ψ_0 from ψ_1 ; Set <i>Flag</i> register to 2
S_{21}	Clock	Activate counter n
S_{22}	Clock, $n \leq N$	Set accumulator z to $(n - \frac{1}{2})\psi$
S_{23}	Clock	Move register x to register \bar{a}
S_{24}	Clock	Update counter n
S_{25}	Clock, $n > N$	Set <i>Flag</i> register to 1; Update counter θ
S_{26}	$\theta > 90^0$	Set register κ to 0.9 and $f(W_{gbest})$ to ∞ ; Activate counter m and t
S_{27}	Clock, $m \leq M$	Set register $f(W_{pbest})$ to ∞ ; Activate counter n
S_{28}	Clock, $n \leq N$	Move register <i>LFSR</i> to register W and V_0
S_{29}	Clock	Update counter n
S_{30}	$n > N$	Update counter m
S_{31}	$m > M, t \leq t_{max}$	Activate counter m
S_{32}	Clock, $m \leq M$	Set register F to 0; Activate counter θ
S_{33}	Clock, $\theta \leq 90^0$	Set register AF_p to 0; Activate counter n
S_{34}	Clock, $n \leq N$	Update register AF_p
S_{35}	Clock	Update counter n
S_{36}	$n > N, AF_p > AF_d$	Set register δ to 1; Update register F
S_{37}	Clock, $AF_p \leq AF_d$	Update counter θ
S_{38}	$\theta > 90^0, F < f(W_{pbest})$	Move register W to register W_{pbest} ; Update register $f(W_{pbest})$
S_{39}	$F < f(W_{gbest})$	Move register W to register W_{gbest} ; Update register $f(W_{gbest})$
S_{40}	Clock, $F \geq f(W_{pbest}), F \geq f(W_{gbest})$	Activate counter n
S_{41}	Clock, $n \leq N$	Update register V_0
S_{42}	$1 < V_0 < 0$	Move register <i>LFSR</i> to register V_0
S_{43}	Clock, $1 \geq V_0 \geq 0$	Update register W
S_{44}	$1 < W < 0$	Move register <i>LFSR</i> to register W
S_{45}	Clock, $1 \geq W \geq 0$	Update counter n
S_{46}	$n > N$	Update counter m
S_{47}	$m > M$	Update counter t ; Update register κ
S_{48}	$t > t_{max}$	Activate counter θ
S_{49}	Clock, $\theta \leq 90^0$	Set register AF_p to 0; Activate counter n
S_{50}	Clock, $n \leq N$	Update register AF_p

Table 7.1: Beamforming processor behaviors in the FSM

State	Command	Action
S_{51}	Clock	Update counter n
S_{52}	$n > N$	Update counter θ
S_{53}	$\theta > 90^0$	Activate counter k
S_{54}	Clock, $k \leq D' - 1$	Shift contents of register φ left
S_{55}	Clock	Update counter k
S_{56}	$k > D' - 1$	Move register θ_d to last content of register φ ; Update counter j
S_{57}	$j > D'$	Halt the process

7.4.2 Proposed architecture

FSMD design technology often implements a digital system as a Register Transfer Level (RTL) circuit [133]. The RTL models a synchronous digital circuit that describes the flow of signals (data) between registers and performs logical operations on these signals/data. It is very convenient to represent the circuit at the register level by the Hardware Description Language (HDL). Also, it may be easier to derive the gate-level representation and the actual wired connectivity of the circuit. For the algorithm, an HDL (Verilog/VHDL) declares the variables with the registers and describes the statements (*if-then-else*, arithmetic expressions) with the combinational logic circuits. The system architecture is developed here as a combination of three modules, such as

- Template,
- CORDIC, and
- PSO.

It uses an embedded clock to synchronize the operation of each module. Fig. 7.7, Fig. 7.8, and Fig. 7.9 respectively illustrate the internal circuits of each module in RTL. In RTL, a Boolean expression (binary arithmetic) composed of variables, constants, and the content of registers, defines the data processing operations. It can be implemented with a combinational logic circuit. It always operates clock-by-clock and performs a state transition of the state register. Thus, a data transfer occurs to a specific sequential logic circuit/register at the next rising edge of the clock. The state register output is used to control the function of the multiplexer and selects the data desired in a particular operation. Closed feedback loops preserve the current contents of sequential logic circuits (registers) when they are not changed. Likewise, a data routing operation is performed with a multiplexing circuit (multiplexer, tri-state buffers) which correctly transfers the desired data to a specific register in each state. A conditional data assignment statement (*if-then-else*) is implemented with a 2×1 multiplexer whose select input contains a Boolean (relational/logical) expression. When the condition is satisfied, it passes the data from the true (logic 1) input port to the output port. Otherwise, it transfers data from the false (logic 0) input port. Also, a loop statement (*for/while*) in the algorithm is executed through three sequential data operations: *preliminary data assignment*, *conditional data assignment*, and *increment/decrement data assignment*. Thus, it realizes a loop counter composed of three-level hardware architectures, such as

- a register (corresponding to the loop variable) which is loaded with the initial data and updates it at each count,
- a 2×1 multiplexer (having a relational expression as select input) to check the status of an update or keep the register contents unchanged, and
- a multiplexer (containing the output of the state register as a select input) which transfers the data to the sequential logic circuit/register in a suitable clock.

While the same loop counters are used in different modules, it resets them to separate clocks. An FPGA chip can store all constants and parametric values of user-defined variables in Block Random Access Memory (BRAM). Also, the random function, $rand()$ is realized with a 16-bit Linear Feedback Shift Register (LFSR). All other data storage/transfer operations are performed with the 12-bit registers for this work.

Thus, the template module contains four data registers ($AF_d, \phi, \theta_d, \theta_n$) and three loop counters (j, k, θ). Also, the CORDIC module includes eight data registers ($Flag, x, y, z, \bar{a}, \sigma, \psi_0, \psi_1$) and three loop counters (i, n, θ). The PSO module involves four loop counters (t, m, n, θ) and eleven data registers ($AF_p, \kappa, W, V_0, F, \delta, rand(), f(W_{pbest}), f(W_{gbest}), W_{pbest}, W_{gbest}$).

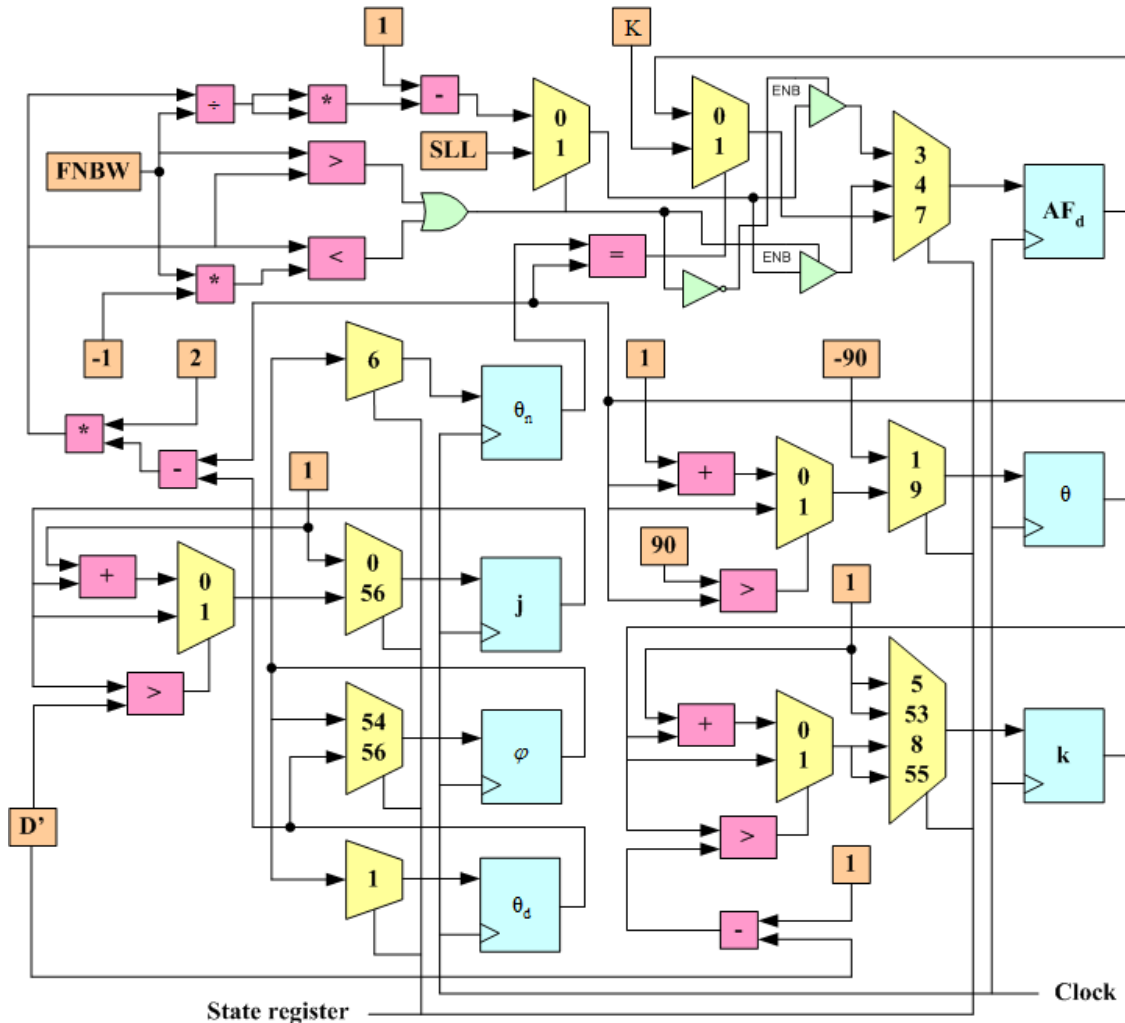


Fig. 7.7. RTL schematic of the Template module

7.4.3 FPGA design flow

The FPGA-based design methodology is now considered a versatile platform for SoC (System-on-Chip) development. It has a structural design that can integrate high-speed programmable processors and resources into a single chip. Implementation performance is also greatly enhanced by the introduction of efficient CAD (Computer Aided Design) tools to map the system design to its architecture [127]. Generic CAD/software design flow has the ability to initiate the hardware description of complex design features with a high-level language (VHDL/Verilog) known as HDL (Hardware Description Language) which eventually converts them into bit-stream for programming/loading to the FPGA [133].

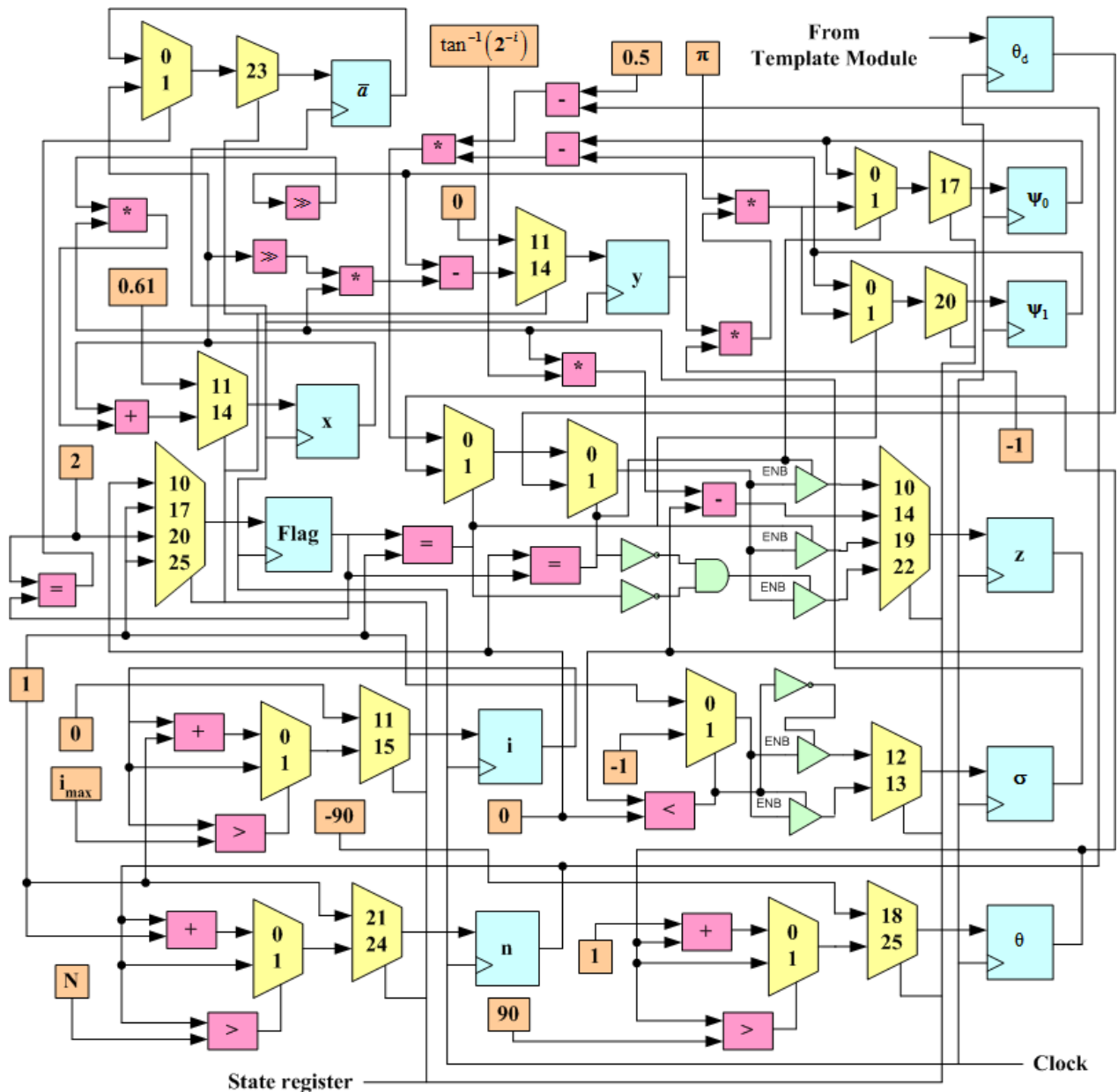


Fig. 7.8. RTL schematic of the CORDIC module

The entire implementation process is depicted by arranging its different modules in chronological order as in Fig. 7.10. At design entry, the structural/behavioral modeling of the design is prepared with a hardware description language for parallel execution. Then, a functional simulation of the design is performed with given sets of test inputs for a certain period to ensure perfect logical functionality. Subsequently, the hardware compilation phase involves several steps that are summarized in synthesis, mapping, clustering/packing, placement and routing. At the synthesis stage, the HDL description of the design at the Register Transfer Level (RTL) is translated into an intermediate logical structure, namely the netlist, which is usually stored in a standard format called EDIF (Electronic Design Interchange Format). The mapping algorithm transforms the netlist into actual macrocells from the given design library, composed of k -input LUTs and flip-flops. The clustering phase forms several clusters each having a group of k logic blocks (CLBs) which are then mapped directly to the Basic Logic Elements (BLEs) on FPGA. The placement algorithm recognizes the logic blocks required in the FPGA for a particular design implementation and optimizes their configuration by packing them close enough to minimize required wiring or balance wiring density or maximize speed. The routing scheme allocates nets to dedicated routing resources.

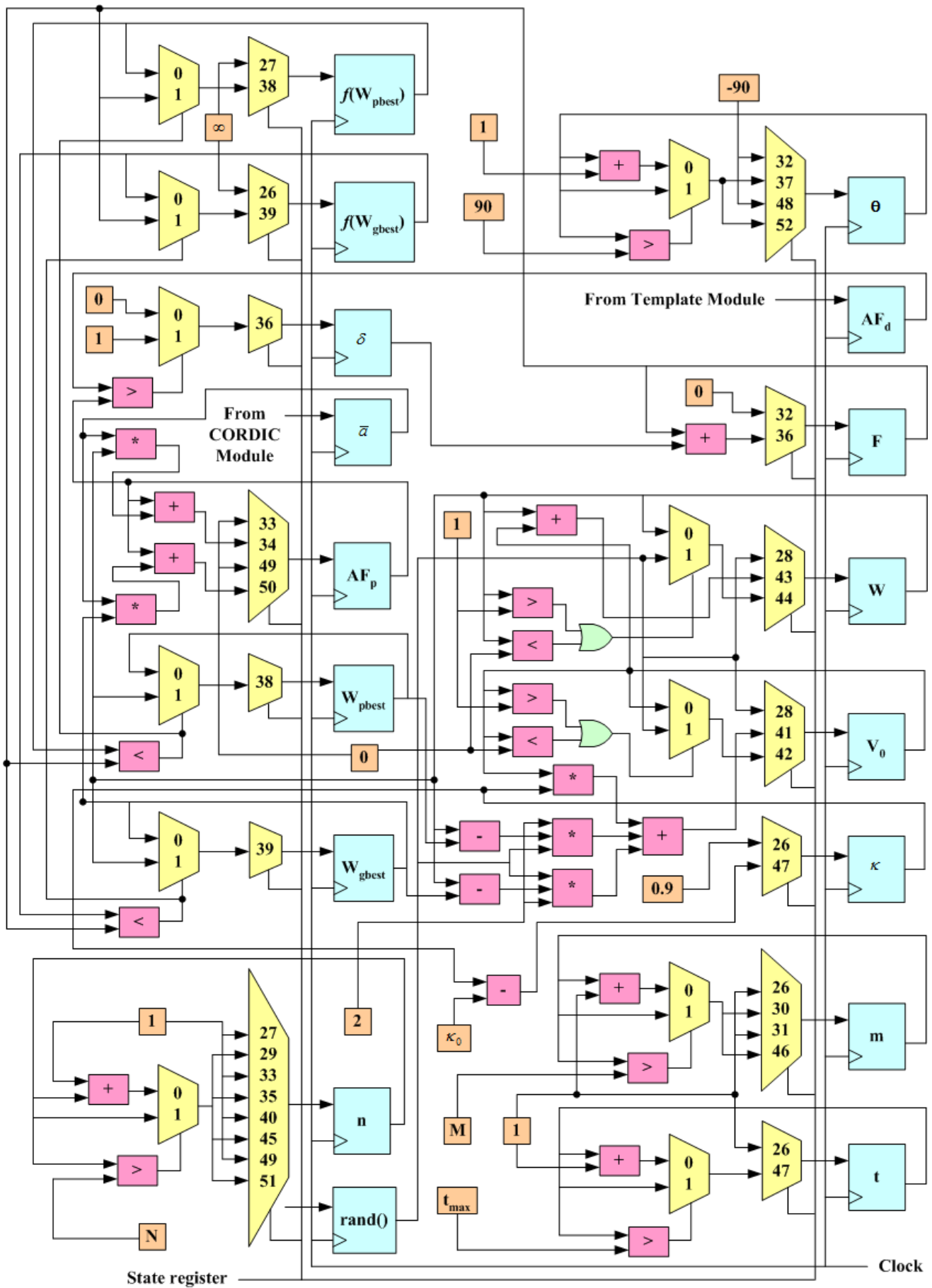


Fig. 7.9. RTL schematic of the PSO module

It can be expressed (in percentage) as follows.

$$\Lambda_b = \left(1 - \frac{\sum_k \delta_k}{K_0} \right) \times 100\% \quad (7.16)$$

where, K_0 is the total number of sample points taken to measure deviations.

- *Hardware resource utilization (v)*: It is a quantitative measure that indicates the number of resources used out of the total resources available in the FPGA device and is usually expressed (in percentage) as a ratio between them as follows.

$$v = \frac{n_u}{n_t} \times 100\% \quad (7.17)$$

where, n_u and n_t represent the number of resources used and the total resources available in the system, respectively.

- *Computation time (τ_b)*: It measures the total time it takes for the system to execute the weight vector computation and produce the optimal pattern. Therefore, it depends on the total number of clock cycles needed to achieve convergence in the search process and expressed (in milliseconds) as follows.

$$\tau_b = t_{con} N_c t_c \quad (7.18)$$

where, N_c , t_{con} , and t_c indicate the number of clock cycles per search, convergence time, and clock period, respectively.

- *Power consumption (ρ_t)*: It is the measurement of total power dissipation in internal circuits which usually arises in two forms: static power (caused by leakage currents) and dynamic power (caused by capacitive loads). So, it is expressed (in milliwatts) as the sum of these two power components as follows.

$$\rho_t = \rho_s + \rho_d \quad (7.19)$$

where, ρ_s and ρ_d denote the static and dynamic power in the system, respectively.

Table 7.2: Simulation parameters

Parameters	Values
Number of antenna elements	20
Spacing between the elements	0.5λ
Sidelobe level	0.0316 (-30 dB)
Beamwidth between the first nulls	20^0
Depth of the nulls	0.00001 (-100 dB)
Angle of arrivals	$-45^0, -30^0, -10^0, 20^0, 40^0$
FPGA board frequency	122.64 MHz
Number of particles	20
Maximum permissible search time in PSO	1000
Maximum permissible iterations in CORDIC	10

7.5.2 Simulation environments

A set of simulations are performed on the Xilinx Virtex-5 (device: XC5VFX30T, frequency: 122.64 MHz) FPGA board for beamforming accuracy, FPGA resource utilization, power consumption, and computation time. Simulation parameters are chosen keeping compatibility with small wireless measurement systems such as the Crossbow MICAz mote which operates with a 2.4 GHz RF transceiver and uses IEEE 802.15.4 MAC protocols. Table 7.2 shows all the simulation parameters for this work. The Verilog Hardware Description Language (HDL) maps the proposed design directly to the FPGA board [132].

7.5.3 Simulation results

The performance of the beamforming system is tested by taking a set of simulations for the metrics mentioned above. Fig. 7.11(a), Fig. 7.11 (b), Fig. 7.11 (c), Fig. 7.11 (d), and Fig. 7.11 (e) show the optimal patterns for specific AoA data respectively. These figures explain that reducing SLL can improve the protection layer in node localization if the null steering fails due to poor AoA estimation. Table 7.3 provides the corresponding optimized values for the elements of the weight vector. Also, Fig. 7.12 illustrates the convergence behaviors of the beamforming process in each case. However, the accuracy and convergence in a beamforming process can vary for different beam steering angles. Therefore, they are presented as the Empirical Cumulative Distribution Function (ECDF) considering 30 runs of the program and plotted in Fig. 7.13 (a) and Fig. 7.13 (b), respectively. Table 7.4 also lists the minimum and maximum levels of accuracy and convergence achieved for these particular cases. It is observed that the accuracy always decreases, and more iterations are required to complete the process when the magnitude of the beam steering angle becomes larger.

The data obtained from the experiments on the Xilinx virtex 5 (version: ISE 14.7) FPGA board verify the complexities of the system (both space and time). Table 7.5 and Table 7.6 provide these data respectively. It only uses fewer resources which remain almost consistent in all cases. Here, we have computed the execution time for a minimum number of iterations needed, and they are also within an acceptable limit. Table 7.7 gives the power consumption statistics. It requires a low power which remains constant in all cases.

Simulation results for resource usage, computation time, and power consumption are compared to some well-known methods and presented in Table 7.8 and Table 7.9, respectively. They clarify that the proposed system often requires less power and can be realized with a minimum of hardware resources. Also, its execution time is comparable to others.

Table 7.3: Optimal values of the weight vector (W_n)

Index n	Directions of beam orientations (θ_d)				
	-45^0	-30^0	-10^0	20^0	40^0
1	0.9623	0.9110	0.9072	0.8864	0.9619
2	0.9190	0.8338	0.8897	0.8716	0.8959
3	0.8844	0.8030	0.7712	0.8391	0.8303
4	0.7528	0.7345	0.7075	0.7293	0.6914
5	0.6722	0.6203	0.5717	0.5760	0.6530
6	0.5950	0.4495	0.4968	0.5288	0.5393
7	0.4030	0.4417	0.3102	0.3836	0.4167
8	0.3415	0.2359	0.2028	0.3029	0.2521
9	0.2692	0.1952	0.1842	0.1789	0.2304
10	0.1860	0.0983	0.0680	0.1040	0.1422

Table 7.4: Accuracy and convergence of the beamforming process

Directions of beam orientations (θ_d)	Accuracy (%)		Iterations	
	Minimum	Maximum	Minimum	Maximum
-45^0	93.37	93.92	9	901
-30^0	97.24	97.79	1	919
-10^0	97.79	98.34	1	968
20^0	97.79	98.34	2	899
40^0	95.03	95.58	1	932

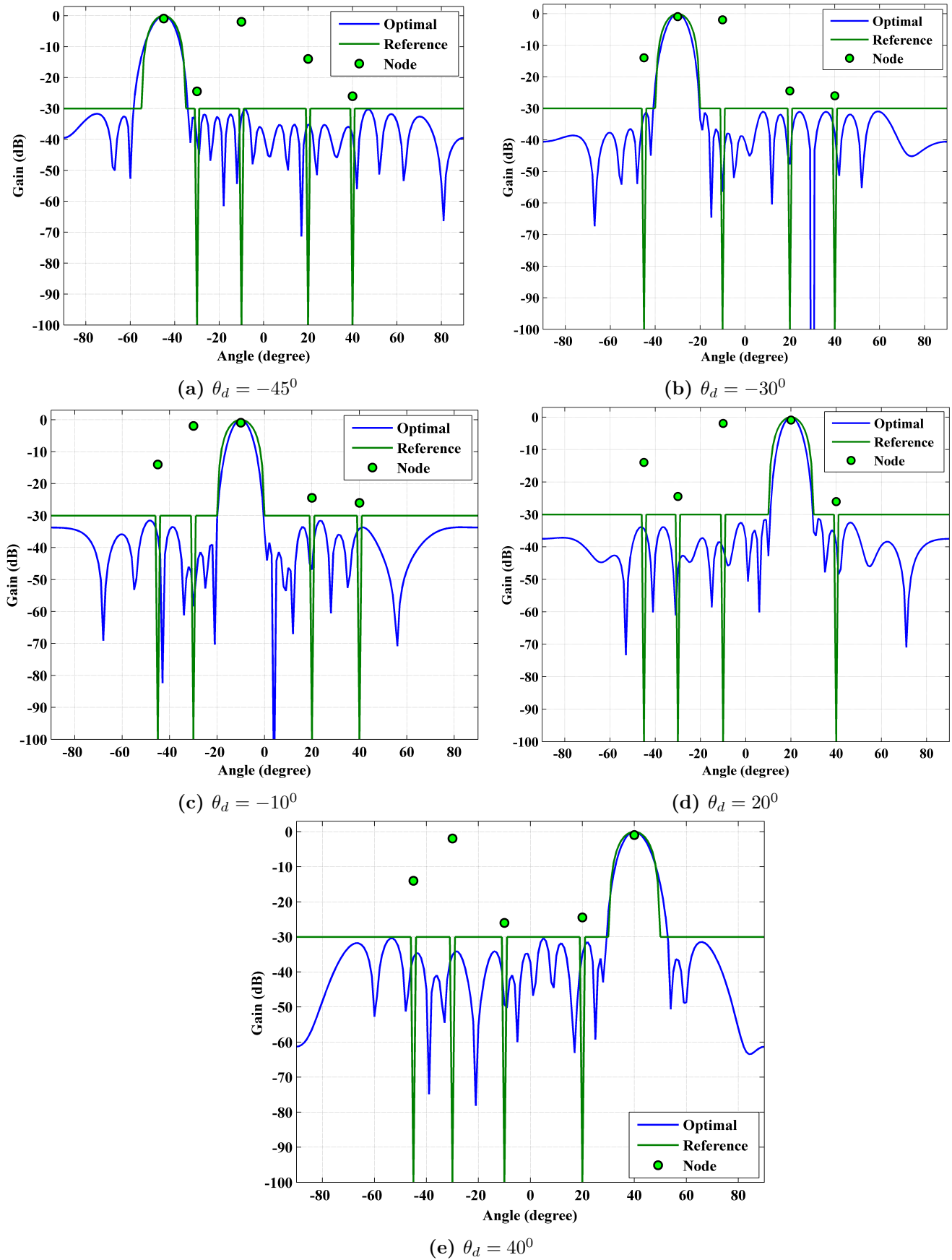


Fig. 7.11. Optimal patterns for a cluster

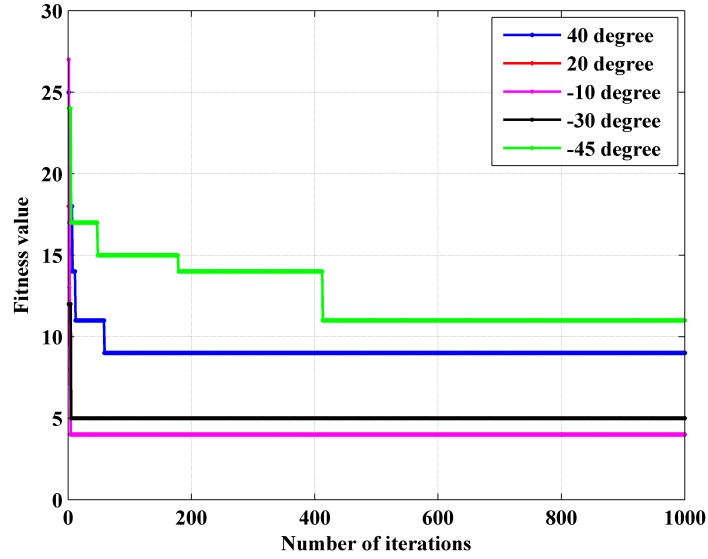


Fig. 7.12. Convergence in the beamforming process

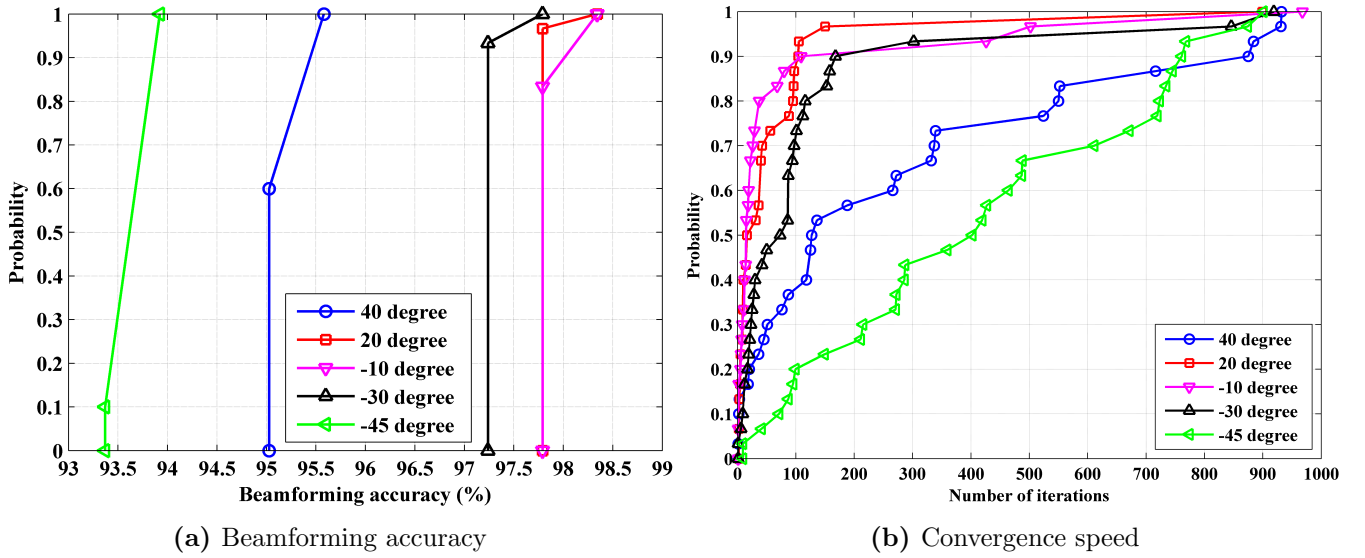


Fig. 7.13. Efficiency of the beamforming process

Table 7.5: FPGA resource utilization

Parameters	Directions of beam orientations (θ_d)				
	-45^0	-30^0	-10^0	20^0	40^0
FFs	724 out of 20480 (3 %)	724 out of 20480 (3 %)	724 out of 20480 (3 %)	724 out of 20480 (3 %)	724 out of 20480 (3 %)
LUTs	1079 out of 20480 (5 %)	1093 out of 20480 (5 %)	1092 out of 20480 (5 %)	1092 out of 20480 (5 %)	1091 out of 20480 (5 %)
Slices	447 out of 5120 (8 %)	442 out of 5120 (8 %)	435 out of 5120 (8 %)	396 out of 5120 (7 %)	453 out of 5120 (8 %)
IOBs	25 out of 360 (6 %)	25 out of 360 (6 %)	25 out of 360 (6 %)	25 out of 360 (6 %)	25 out of 360 (6 %)
BRAM	1 out of 68 (1 %)	1 out of 68 (1 %)	1 out of 68 (1 %)	1 out of 68 (1 %)	1 out of 68 (1 %)
DSP48	7 out of 64 (10 %)	7 out of 64 (10 %)	7 out of 64 (10 %)	7 out of 64 (10 %)	7 out of 64 (10 %)

Table 7.6: Computation time

Parameters	Directions of beam orientations (θ_d)		
	-45^0	-30^0	-10^0
Clock cycles per search = Time elapsed (μ s) \times Device frequency (MHz)	88.76×122.64 = 10885	88.76×122.64 = 10885	88.76×122.64 = 10885
Total clock cycles	10885×9 = 97965	10885×1 = 10885	10885×2 = 21770
Clock period (ns)	8.15	8.15	8.15
Computation time (ms) = Total clock cycles \times Clock period (ns)	$10885 \times 9 \times 8.15$ = 0.798	$10885 \times 1 \times 8.15$ = 0.089	$10885 \times 2 \times 8.15$ = 0.177
			40^0 88.76×122.64 = 10885

Table 7.7: Power consumption

Parameters	Directions of beam orientations (θ_d)		
	-45^0	-30^0	-10^0
Dynamic power (mW)	6.66	6.66	6.66
Static power (mW)	840.10	840.10	840.10
Total on-chip power (mW)	846.76	846.76	846.76

Table 7.8: Comparison of FPGA resource utilization

Beamforming systems	LUTs	FFs	Slices	IOBs	BRAM	DSP48
PSO-CORDIC	1093 out of 20480 (5 %)	724 out of 20480 (3 %)	453 out of 5120 (8 %)	25 out of 360 (6 %)	1 out of 68 (1 %)	7 out of 64 (10 %)
SCC [121, 123]	43560 out of 53248 (82 %)	10727 out of 53248 (20 %)	26622 out of 26624 (100 %)	17 out of 448 (4 %)	...	8 out of 64 (13 %)
MSR-CORDIC [126]	37820 out of 53248 (71 %)	8597 out of 53248 (16 %)	22315 out of 26624 (83 %)	421 out of 448 (93 %)	49 out of 416 (11 %)	...
QRD-RLS [134]	5411 out of 53248 (10 %)	5916 out of 53248 (11 %)	3530 out of 26624 (13 %)	...	6 out of 416 (1 %)	13 out of 64 (20 %)
D3 [135]	42572 out of 63400 (67 %)	18251 out of 63400 (28 %)	12268 out of 15850 (77 %)	...	14 out of 135 (10 %)	64 out of 240 (26 %)

Table 7.9: Comparison of computation time and power consumption

Beamforming systems	Computation time (ms)	Power consumption (mW)
PSO-CORDIC	0.089	846.76
D3 [135]	6	4284
QRD-MGS [136]	0.037	2654
STAP [137]	0.144	6369

7.5.4 Performance analysis

The time complexity of the beamformer can be computed as the sum of the complexities obtained in its three modules (e.g. Template, CORDIC, and PSO). Here, the template generation depends on the number of received signals or the number of neighboring nodes and thus its complexity is $O(n)$. Likewise, the convergence of the PSO algorithm depends on the number of array elements and the number of particles. So, it shows a complexity of $O(n^2)$. However, the CORDIC algorithm normally converges after the 10-th iteration to compute the sine/cosine function and therefore has constant complexity. Thus, the total complexity is $O(n^2)$.

7.6 Summary

In this chapter, we have proposed the design and implementation of a new beamformer for a smart antenna. It has a reconfigurable architecture that provides a stable link to the desired node each time with a higher directive gain pattern. Besides, patterns are produced with lower SLL and by placing deep nulls in the directions of neighboring nodes, keeping them out of radio coverage. Thus, it is well suited for the precise localization of wireless nodes in a hostile environment. The beamformer works with three separate modules, such as Template, CORDIC, and PSO. Based on the estimated AoA data, the Template module produces a reference pattern that has a prescribed SLL, FNBW, and null depth. The CORDIC module produces the steering vector for the smart antenna based on the array design parameters. The PSO module produces an optimal value for the weight vector of the array evaluating the fitness function, resulting in an optimal pattern. In this work, we used the FSMMD design technique to realize the beamformer on a dedicated FPGA chip and verified its performance with multiple fixed-point hardware-level simulations of several metrics. It shows that higher beamforming accuracy is achieved using minimal hardware resources. This validates its operational feasibility by tiny nodes with limited resources. However, this system is developed with a behavioral model and reveals high latency. Thus, it is still possible to test its performance on a parallel and pipeline architecture in the future.

Chapter 8

Implementation of an anchor coprocessor architecture on FPGA

8.1 Introduction

Localization systems must be robust, accurate, and energy-efficient in a WSN. A localization system using a mobile anchor that computes location centrally can avoid computational overhead at resource-constrained nodes. It speeds up the localization process and improves network lifetime by reducing node power consumption. The anchor often plays a vital role in accurately estimating the positions of nodes in the network. Equipped with a GPS receiver, it provides reference information for position computations by trilateration/triangulation. Thus, an anchor coprocessor on dedicated hardware performs the localization process faster with low power consumption. Moreover, the localization process becomes secure by preserving the privacy of the anchor (the introduction of mobility always ensures the lowest probability of being compromised). An anchor often utilizes the CPU (Central Processing Unit) more efficiently by balancing the loads between two processors. The coprocessor architecture must also be reconfigurable to become scalable across the network, coping with radio irregularities. Although several digital platforms currently exist in the literature, the FPGA-based technology is well suited to a WSN infrastructure. FPGA families produce reconfigurable architectures to meet resource constraints requiring relatively minimal hardware [138]. Despite this, only a few research proposals have resulted in the hardware implementation of localization systems in WSNs. These were also based on conventional RSS/AoA measurement techniques which cannot perform well in a harsh environment.

In this chapter, we have proposed the design and implementation of a coprocessor architecture for a mobile anchor on an FPGA chip. The coprocessor operates on a localization system that computes node positions centrally using RSS and AoA information obtained with a smart antenna. It uses CORDIC to evaluate the sine/cosine function for particular AoA data and the appropriate FSMD models. It provides robust and precise node localization with less computation time.

The rest of this chapter is organized as follows. In Section 8.2, an overview of coprocessor functionality in computing node positions is given. In Section 8.3, the design methodology of the proposed coprocessor architecture is discussed. In Section 8.4, the hardware implementation on FPGA is explained. In Section 8.5, the results of the simulation are presented. Finally, Section 8.6 discusses the limitations and possible modifications of this system in the future.

8.2 Coprocessor features

The anchor distributes all the tasks of the localization process between its two processors: the main processor and the coprocessor. Fig. 8.1 illustrates the data communications between the different parts of the anchor for the operation of its coprocessor (the solid line and the dotted line represent the data transfer in the coprocessor and the main processor, respectively). The main processor performs various tasks such as mode switching, GPS data retrieval, smart antenna control, etc. It transforms GPS data collected from geodetic coordinates (latitude, longitude) into Cartesian coordinates enabling position computation with a trigonometric principle (triangulation/trilateration). It collects angle of arrival and RSS data from the built-in AoA estimator and signals intercepted on the smart antenna, respectively. It then passes this data to the coprocessor for further processing. The coprocessor divides the entire localization process into three successive phases, such as

- cluster formation,
- distance and angle (image) estimation, and
- position computation.

It only considers the signals received from neighboring nodes having an RSS exceeding a prescribed threshold and thus forms a cluster. For each clustering node, it estimates the distance (d) using the path loss and computes the angle (image) for the accumulated angle of arrival (θ) information. It keeps all these data (ID, d, θ and θ') and its current reference (x_a, y_a) retrieved from the GPS receiver in a table. It then iteratively checks and modifies the table to have a few nodes/IDs twice for two distinct anchor points. Finding a few of these nodes, it computes the position only for them and keeps the data in a separate table. Finally, the main processor transmits the estimated position data to the corresponding nodes using an appropriate scheduling mechanism. It also updates the database for any nodes that are not yet localized. If it turns out that there are still a few nodes to localize, it retains its current mode of operation (active/receive) and activates the servo systems to move to a new reference along a random trajectory. Otherwise, it will switch into a power-saving mode (idle/sleep).

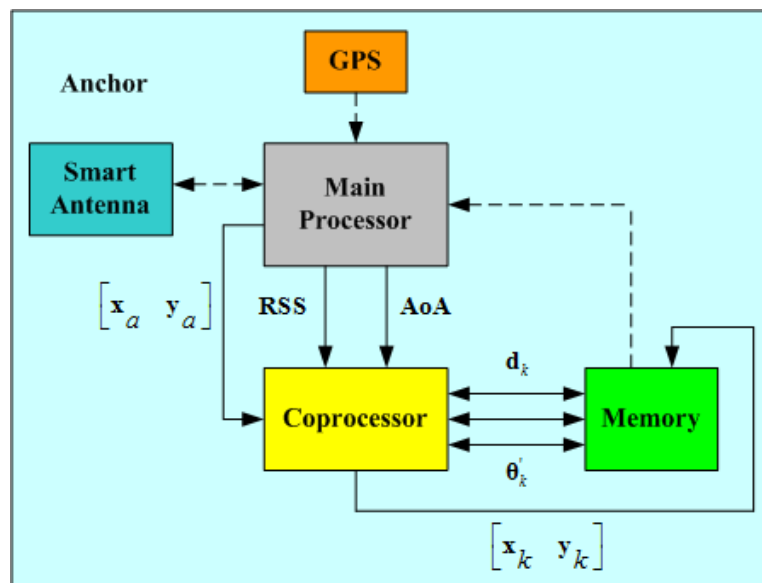


Fig. 8.1. Functional diagram of the anchor coprocessor

8.3 Design methodology

The coprocessor performs all tasks related to node position computation, such as cluster formation, distance estimation, angle (image) calculation, deviation calculation, position estimation (original/imaginary and mean), etc. Each time, it stores the mean position in memory. To achieve this, it also involves evaluating a few mathematical functions like square root, modulus, sine/cosine, etc.

8.3.1 Cluster formation

The coprocessor responds only to signals above a prescribed threshold in signal-to-noise ratio (SNR_{th}). It often forms a unique cluster with a few neighboring nodes whose RSS exceeds the SNR_{th} . It stores all information of clustering nodes in table, U. Algorithm 19 explains cluster formation process given below. Each time, the process begins with a random anchor point and a finite number of nodes. And this procedure continues until it reaches the maximum permissible anchor points (t_{max}) or the localization of all nodes is finished. Since the formation of the cluster requires several data about the signals received from nodes and anchor references, they are considered as input variables. Likewise, the cluster size and the pointer to the data table are considered output variables.

Algorithm 19 Pseudo-code for forming clusters and keeping their databases

Input: Maximum number of permissible anchor points (t_{max}), Number of nodes in active mode (N_s), Threshold level in signal-to-noise ratio (SNR_{th}), Received signal strength (RSS), Angle of arrivals of the received signals (AoA), Neighboring node IDs: $[ID]$, Position of the anchor (x_a, y_a) in Cartesian coordinates, and Receive mode of anchor: mode state (ms) = ‘Set’

Output: Cluster size (CS), Pointer value in data table (Ptr), Idle mode of the anchor: mode state (ms) = ‘Reset’

```

1: Initialize:  $Ptr, R = 0, t, ms = \text{'Set'}, N_s, SNR_{th}, t_{max}$ 
2: while ( $t \leq t_{max}$ ) do
3:    $x_a \leftarrow 1000 * rand(); y_a \leftarrow 1000 * rand(); CS \leftarrow 0; i \leftarrow 1$ 
4:   while ( $i \leq N_s$ ) do
5:     if ( $RSS[i] \geq SNR_{th}$ ) then
6:        $CS \leftarrow CS + 1; Ptr \leftarrow Ptr + 1; U[Ptr][1] \leftarrow ID[i]; U[Ptr][2] \leftarrow RSS[i];$ 
7:        $U[Ptr][3] \leftarrow AoA[i]; U[Ptr][4] \leftarrow x_a; U[Ptr][5] \leftarrow y_a; U[Ptr][6] \leftarrow 0$ 
8:     end if
9:      $i \leftarrow i + 1$ 
10:  end while
11:  if ( $Ptr = 2N_s$ ) then
12:    break
13:  end if
14:   $t \leftarrow t + 1$ 
15: end while
16:  $ms \leftarrow \text{Reset}$ 
17: return  $ms, CS, Ptr$ 

```

8.3.2 Distance and angle (image) estimation

The coprocessor estimates the distance of clustering nodes using the path loss model. The path loss (L_p) is a function of distance (d) which is estimated using equation (2.1). Considering log-normal shadowing conditions in free space propagation, the distance is estimated by choosing the parameters as $X_{\sigma\eta} = 1, \alpha = 2$ and $d_0 = 1$ meter in equation (2.1).

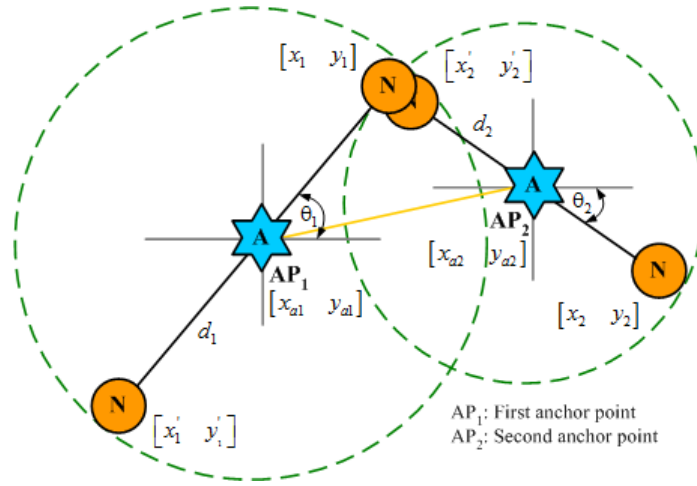


Fig. 8.2. Generation of original and imaginary position for nodes

Usually, the smart antenna has a built-in AoA estimator. A high-resolution AoA estimator based on ESPRIT is considered for this work. Assuming that the entire array is a combination of two identical sub-arrays, the angle (θ) of the clustering node is estimated using equation (2.4). The angle so derived remains in the range of $\{-90^\circ, 90^\circ\}$. However, the same angular value would always be estimated for a clustering node located at one of the two positions in the WSNs. This occurs due to the symmetry property of the geometry. These two positions (one is original, and the other imaginary) are obtained by extending a line of length, d at this particular angle on both sides keeping the anchor in the center, as shown in Fig. 8.2. Thus, each time the coprocessor must also calculate the angle θ' , which is supposed to be an image of the estimated angle θ . The angle (image) is evaluated counter-clockwise using equation (3.5). Algorithm 20 explains the process of estimating distance and angle (image) for clustering nodes as below. It uses data of pointer value, cluster size, EIRP, smart antenna gain, etc., to find the distance and angle (image) of a clustering node. Accordingly, they become the input and output variables, respectively.

Algorithm 20 Pseudo-code for estimating distance and angle (image) of clustering nodes

Input: Cluster size (CS), Pointer value in data table (Ptr), Effective isotropic radiated power of nodes ($EIRP$), Gain of the smart antenna (G_r), and Idle mode of the anchor: mode state (ms) = ‘Reset’

Output: Clustering node distance: $[d]$, Angle (image) for the signals received: $[\theta']$ and Idle mode of the anchor: mode state (ms) = ‘Reset’

```

1: Initialize:  $j = R + 1, EIRP, G_r, ms = \text{'Reset'}$ 
2: if ( $CS \neq 0$ ) then
3:   while ( $j \leq Ptr$ ) do
4:      $L_p[j] \leftarrow \left( \frac{EIRP}{U[j][2]} \right) * G_r; d[j] \leftarrow \left( \frac{\lambda}{4\pi} \right) * \sqrt{L_p[j]}; U[j][2] \leftarrow d[j]; \theta[j] \leftarrow U[j][3]$ 
5:      $\theta'[j] \leftarrow \pi + \theta[j]; U[j][6] \leftarrow \theta'[j]$ 
6:      $j \leftarrow j + 1$ 
7:   end while
8: end if
9:  $ms \leftarrow \text{Reset}$ 
10: return  $ms, d, \theta'$ 

```

8.3.3 Position computation

The coprocessor computes the position of the nodes using the estimated distance, angle, and reference data. Each time it checks the U table for nodes/IDs twice at two distinct anchor points.

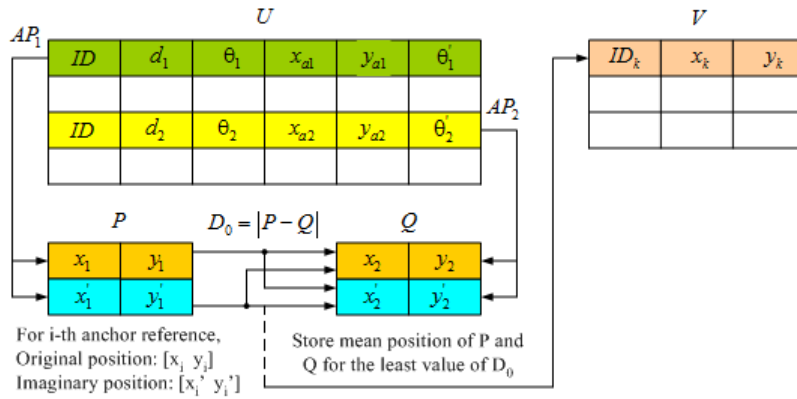


Fig. 8.3. Computation process in position estimation for nodes

Finding such a node, it produces two sets of position data with two anchor references and keeps them in two separate tables, P and Q , shown in Fig. 8.3. However, each table also contains two distinct positions for each node. One is original, and the other is imaginary as mentioned earlier. It computes the original position and the imaginary position using the estimated angle and the angle (image), respectively. For the i -th anchor point ($i = 1, 2$), it is expressed using equations (3.6) and (3.7). Fig. 8.2 illustrates the generation of the original and imaginary positions of a node with two anchor points. It shows that a node can exist at any of the three locations (one of which is original and the other two are imaginary) of the network. However, the original location may slightly deviate from two anchor points due to RSS/AoA measurement error. Thus, the coprocessor must take into account a total of four locations. It uses two steps to compute the correct position of a node, as illustrated in Fig. 8.3. First, it compares the P and Q tables for the locations generated at the two reference points. Finally, it computes the mean position of the tables producing the lowest deviation (D_0). It stores the ID and location obtained for each node in the V table. Algorithm 21 explains the position computation process for nodes as below. It uses data of distance, angle, angle (image), anchor references, etc., to compute the position of clustering nodes. Thus, they become the input and output variables, respectively.

However, the design of the coprocessor aims to speed up the position computation process. It acts as a hardware accelerator. Accordingly, the Babylonian method is used to evaluate the square root function in distance estimation. Algorithm 22 explains the steps for calculating the square root of a given number (s_0). Initially, it considers a random guess value (w_0) in the range $\{0,100\}$ since the maximum communication range of 100 meters is chosen for the nodes. Here, the function, $rand()$ represents a random distribution on the boundary of $\{0,1\}$. It gives an acceptable value for the square root function by iterations. Therefore, they are the input and output variables, respectively.

Additionally, the standard CORDIC is used to evaluate the sine/cosine function which should be used to estimate the original and imaginary position of a node. In the previous chapter, Algorithm 15 explains the calculation of the sine/cosine function of a given angle (θ) in a circular coordinate system. Similarly, the evaluation of a modulus function is used to estimate deviation data. Algorithm 23 explains the steps for calculating the modulus of a given number (f_0). Regardless the value of the input number, it only finds its magnitude at the output.

8.4 System implementation

The coprocessor is implemented on an FPGA board. The FPGA enables parallel processing and thus becomes the most popular and flexible design solution. Unlike conventional processors, it avoids instruction fetch and data load/store bottlenecks at runtime.

Algorithm 21 Pseudo-code for computing the position of clustering nodes

Input: Clustering node distance: $[d]$, Angle of arrival and angle (image) for the signals received: $[\theta]$ and $[\theta']$, Position of the anchor (x_a, y_a) , Pointer value in data table (Ptr), Number of nodes in active mode (N_s), Node IDs in sleep mode: $[s'ID]$, Node IDs in active mode: $[n'ID]$ and Idle mode of the anchor: mode state (ms) = 'Reset'

Output: Position of the node (x_k, y_k) and Active mode of the anchor: mode state (ms) = 'Set'

```

1: Initialize:  $k, idx_1, idx_2, idx_3 = 1, l = k + 1, D_{min} = \infty, ms = \text{'Reset'}$ 
2: if ( $Ptr \geq 2$ ) then
3:   while ( $k \leq Ptr$ ) do
4:     while ( $l \leq Ptr$ ) do
5:       if ( $U[k][1] = U[l][1]$ ) then
6:          $r' \leftarrow l; idx_1 \leftarrow idx_1 + 1$ 
7:       end if
8:        $l \leftarrow l + 1$ 
9:     end while
10:    if ( $idx_1 = 2$ ) then
11:       $P[1][1] \leftarrow U[k][4] + U[k][2] * \cos(U[k][3]); P[1][2] \leftarrow U[k][5] + U[k][2] * \sin(U[k][3])$ 
12:       $P[2][1] \leftarrow U[k][4] + U[k][2] * \cos(U[k][6]); P[2][2] \leftarrow U[k][5] + U[k][2] * \sin(U[k][6])$ 
13:       $Q[1][1] \leftarrow U[r'][4] + U[r'][2] * \cos(U[r'][3]); Q[1][2] \leftarrow U[r'][5] + U[r'][2] * \sin(U[r'][3])$ 
14:       $Q[2][1] \leftarrow U[r'][4] + U[r'][2] * \cos(U[r'][6]); Q[2][2] \leftarrow U[r'][5] + U[r'][2] * \sin(U[r'][6])$ 
15:      while ( $m \leq 2$ ) do
16:        while ( $n \leq 2$ ) do
17:           $D_0 \leftarrow (|P[m][1] - Q[n][1]| + |P[m][2] - Q[n][2]|)$ 
18:           $x_{temp} \leftarrow \frac{P[m][1] + Q[n][1]}{2}; y_{temp} \leftarrow \frac{P[m][2] + Q[n][2]}{2}$ 
19:          if ( $D_0 \leq D_{min}$ ) then
20:             $D_{min} \leftarrow D_0; x_k \leftarrow x_{temp}; y_k \leftarrow y_{temp}$ 
21:          end if
22:           $n \leftarrow n + 1$ 
23:        end while
24:         $m \leftarrow m + 1$ 
25:      end while
26:       $s'ID[idx_2] \leftarrow U[k][1]; V[idx_2][1] \leftarrow s'ID; V[idx_2][2] \leftarrow x_k; V[idx_2][3] \leftarrow y_k; idx_2 \leftarrow idx_2 + 1$ 
27:    else
28:       $n'ID[idx_3] \leftarrow U[k][1]; U[idx_3][1] \leftarrow n'ID; U[idx_3][2] \leftarrow U[k][2]; U[idx_3][3] \leftarrow U[k][3];$ 
29:       $U[idx_3][4] \leftarrow U[k][4]; U[idx_3][5] \leftarrow U[k][5]; U[idx_3][6] \leftarrow U[k][6]; idx_3 \leftarrow idx_3 + 1$ 
30:    end if
31:     $k \leftarrow k + 1$ 
32:  end while
33:   $Ptr \leftarrow idx_3; N_s \leftarrow N_s - idx_2 + 1; R \leftarrow Ptr$ 
34:  if ( $N_s = 0$ ) then
35:    break
36:  end if
37: end if
38:  $ms \leftarrow \text{'Set'}$ 
39: return  $ms, x_k, y_k$ 

```

Algorithm 22 Pseudo-code to realize the square root function using the Babylonian method

Input: Given number (s_0), Guess value (w_0) and Maximum permissible iterations (it_{max})

Output: Square root function $\{\sqrt{s_0}\}$: w_0

```

1: Initialize:  $it = 0, w_0 = 100 * rand(), it_{max}$ 
2: while ( $it \leq it_{max}$ ) do
3:    $w_0 \leftarrow \left(w_0 + \frac{s_0}{w_0}\right) * 0.5$ 
4:    $it \leftarrow it + 1$ 
5: end while
6: return  $w_0$ 

```

Algorithm 23 Pseudo-code to realize the Modulus function

Input: Given number (f_0)

Output: Modulus function $\{mod(f_0)\}$: Y_0

```

1:
2: if ( $f_0 \geq 0$ ) then
3:    $Y_0 \leftarrow f_0$ 
4: else
5:    $Y_0 \leftarrow -f_0$ 
6: end if
7: return  $Y_0$ 

```

It develops a reconfigurable and distributed arithmetic structure using its internal logic blocks. The architecture of the coprocessor consists of two functional blocks, such as

- control unit, and
- data unit.

Thus, it is developed using the Finite State Machine with Datapath (FSMD) design methodology. The Finite State Machine (FSM) acts as a controller, and the datapath performs signal (data) processing operations in the system. It translates the sequence of operations (statements) of the localization algorithm into states and represents them by state diagrams. A state diagram includes several circles and arcs that describe the operational flow through a series of commands and actions specified by arithmetic expressions. Such expressions involve external inputs, outputs, and other variables used in the algorithm.

In this work, we realized the functionality of the coprocessor with four distinct modules, such as

- cluster formation,
- distance and angle (image) estimation,
- CORDIC evaluation, and
- position computation.

Based on the RSS information, each time the first module forms a cluster of size, CS. Considering the parameters of wave propagation ($\lambda, X_\sigma, \alpha, d_0$) and the antenna characteristics ($EIRP, G_r$), the second module produces the distance, d and the angle (image), θ' . The third module evaluates the sine/cosine function based on the *Flag* content. Finally, by comparing the (original/imaginary) positions estimated at two anchor points, the fourth module evaluates the mean position, $[x_k, y_k]$ corresponding to the smallest difference, D_0 .

8.4.1 Finite state machine with datapath

The datapath is a set of registers (to store interim data), data processing units (to compute arithmetic expressions), and routing networks (to transfer data via buses). It performs a specific operation based on the command signal from the FSM and produces the internal status signal (flag). On the other hand, FSM has a state register and input and output logic for state transition. It uses external commands and the datapath status signal as input. Thus, it generates the control signals to specify the datapath operation. Also, it can produce an external status signal (reset/halt) which indicates the functional status of the system.

We used four distinct FSMD modules that describe the behaviors of the proposed coprocessor architecture. Fig. 8.4 illustrates the state diagram for forming a cluster based on RSS information. For each anchor point, the cluster forming module checks the signals received from the nodes whose RSS exceeds the SNR_{th} and collects the data to be kept in the U register. It updates the CS and Ptr registers each time it finds a new node in the cluster. The verification process for a particular cluster formation continues for all nodes in active mode. The cluster

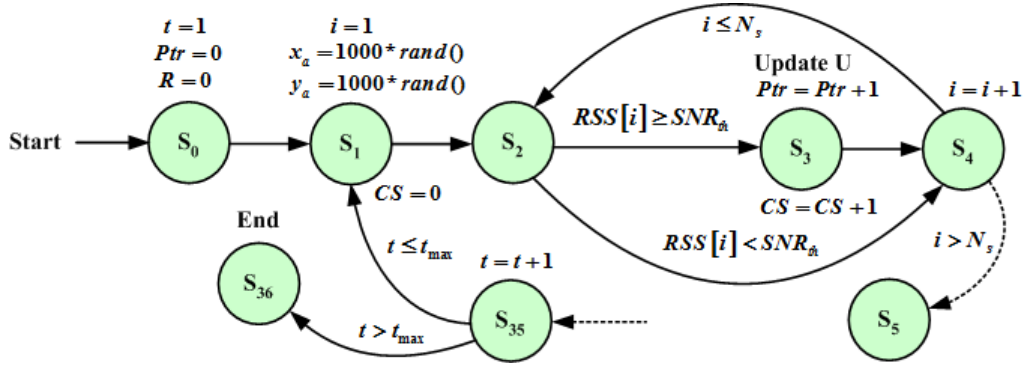


Fig. 8.4. State diagram of cluster formation module

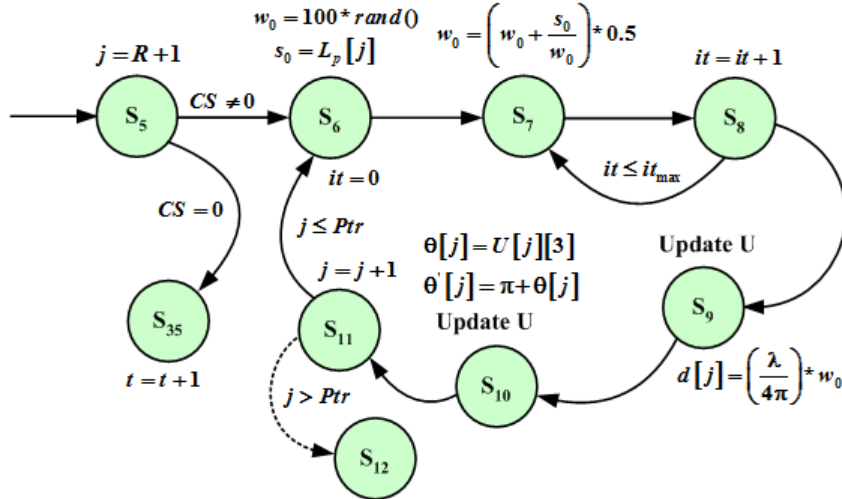


Fig. 8.5. State diagram of distance and angle (image) estimation module

formation process repeats itself until all nodes go into sleep mode or it reaches the maximum allowed time. Likewise, Fig. 8.5 shows the state diagram for estimating distance and angle (image) based on wave propagation and antenna characteristics. Each time, the distance and angle (image) estimation module uses the data from the L_p register to calculate the distance for each clustering node. It also calculates the angle (image) by taking the data from the appropriate cell of the U register. It then updates the contents of the corresponding cells with this data. The estimation process continues for all nodes in a cluster.

If no cluster forms at a particular anchor point (this often occurs when most nodes go into sleep mode), it updates the contents of the t counter. Fig. 8.6 shows the state diagram for generating the original and imaginary positions for nodes encountered twice. It uses four status signals (flag) to control the calculation of trigonometric sine and cosine functions on the CORDIC module.

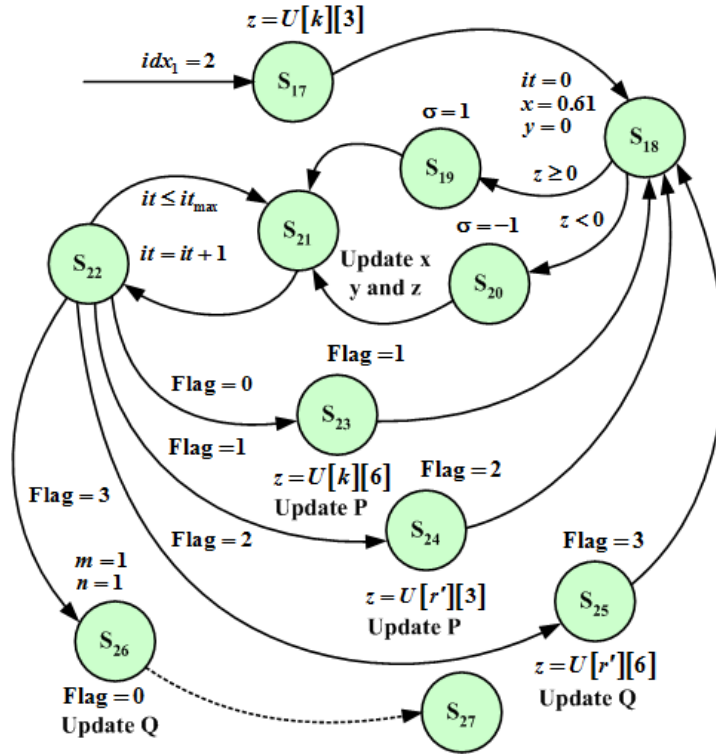


Fig. 8.6. State diagram of the CORDIC module

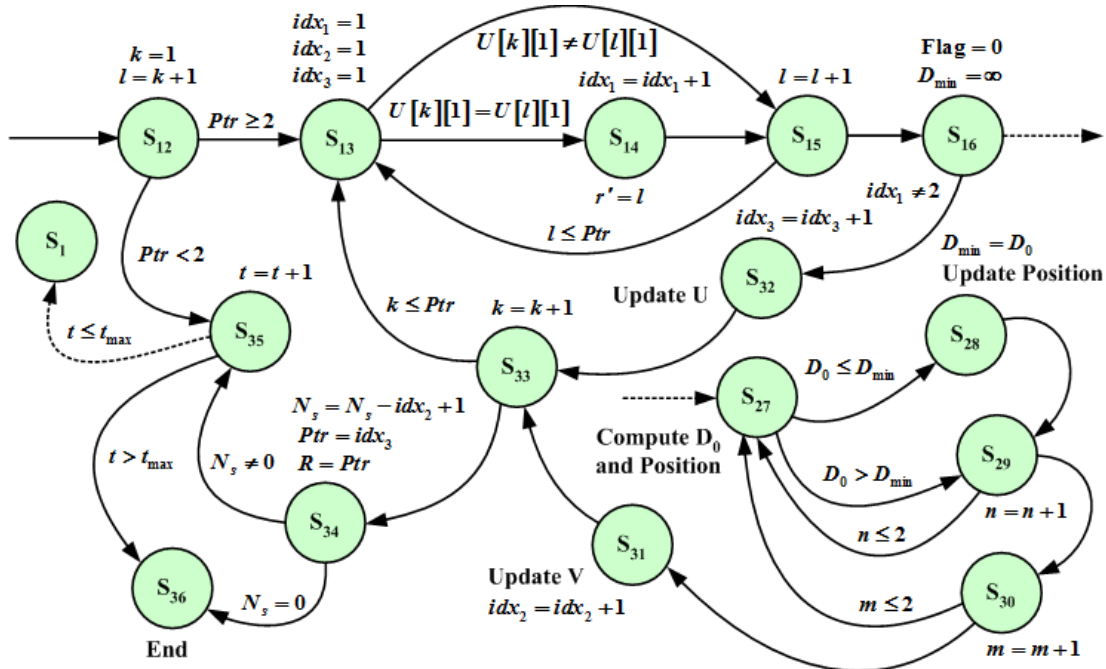


Fig. 8.7. State diagram of position computation module

By setting the *Flag* register to a value from 0 to 3, the CORDIC module calculates the original and imaginary positions based on the data stored in the U register for two anchor points. It keeps this data in the P and Q registers, respectively. Also, Fig. 8.7 shows the state diagram for node position computation.

When the value of the register Ptr becomes greater than 2, the position computation module checks the nodes encountered twice in the register U . It compares the positions (original or imaginary) estimated at two anchor points for such nodes. It then calculates the mean positions giving the minimum difference (D_{min}). It initializes the minimum difference content to infinity ($\infty \geq 100$) to update with the first difference content (D_0). It keeps the IDs and locations so derived in the appropriate cell of the V register. Likewise, it updates the U register and the idx_3 counter for the nodes encountered once. The computation and update process continues until it becomes equal to the contents of the Ptr register. Then, it updates the N_s , Ptr , and R registers with the contents of the idx_2 and idx_3 counters. The coprocessor stops its operation if there is no node in active mode or if it reaches the maximum allowed time. Otherwise, it passes control to the cluster forming module or when the content of the Ptr register becomes less than 2. Table 8.1 describes the features of the anchor processor in each state.

8.4.2 Proposed architecture

The FSM design technology implements the Register Transfer Level (RTL) circuit. The RTL abstraction models a synchronous circuit that describes the flow of signals (data) between registers and performs logical operations on those signals/data. It can conveniently represent the circuit at register level by Hardware Description Language (HDL). Also, it is easier to derive gate-level representation and wire connectivity. For the algorithm, an HDL (Verilog/VHDL) declares the variables with the registers and describes the statements (*if-then-else*, arithmetic expressions) with the combinational logic circuits. Here, the coprocessor architecture is developed as a combination of four modules that perform clustering, distance/angle (image) calculation, CORDIC evaluation, and position computation. It uses an embedded clock to synchronize the operation of each module. Fig. 8.8, Fig. 8.9, Fig. 8.10, and Fig. 8.11 illustrate the internal circuits of each module, respectively. In RTL theory, Boolean expressions (binary arithmetic) define data processing

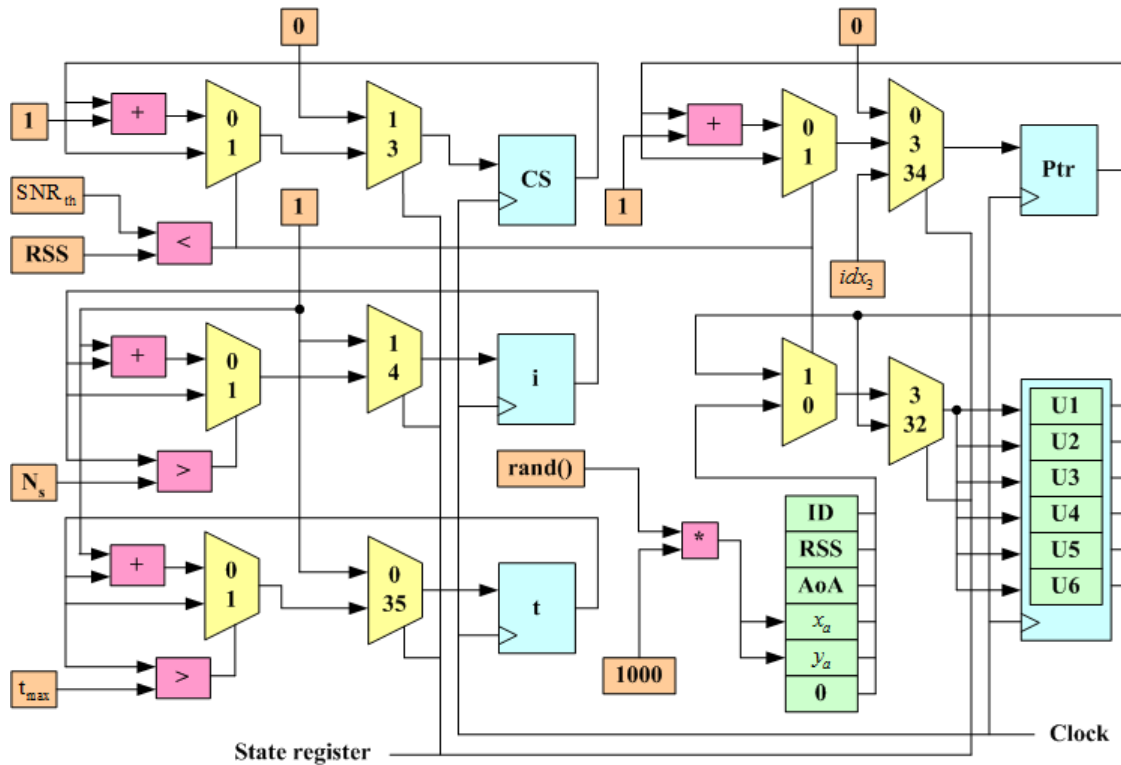


Fig. 8.8. RTL schematic of cluster formation module

operations using variables, constants, and register contents as inputs. Combinational logic circuits implement them. They always operate clock-by-clock and make a state transition of the state register.

Table 8.1: Anchor coprocessor behaviors in the FSM

State	Command	Action	Mode
S_0	Reset/Start	Activate counter t ; Set register Ptr and R to 0	Receive
S_1	Clock, $t \leq t_{max}$	Activate counter i ; Set register CS to 0; x_a and y_a to $\{0,1000\}$	Receive
S_2	Clock, $i \leq N_s$	Compare register RSS with SNR_{th}	Idle
S_3	$RSS \geq SNR_{th}$	Increment register CS and Ptr by 1; Load register U	Idle
S_4	Clock, $RSS < SNR_{th}$	Update counter i	Idle
S_5	Clock	Activate counter j ; Check register CS for Null	Idle
S_6	$CS \neq 0, j \leq Ptr$	Set register w_0 to $\{0,100\}$; Activate counter it	Idle
S_7	Clock, $it \leq it_{max}$	Update register w_0	Idle
S_8	Clock	Update counter it	Idle
S_9	Clock	Load register d ; Update register $U[2]$	Idle
S_{10}	Clock	Load register θ' ; Update register $U[6]$	Idle
S_{11}	Clock	Update counter j	Idle
S_{12}	Clock, $j > Ptr$	Activate counter k and l	Idle
S_{13}	$k \leq Ptr, l \leq Ptr, Ptr \geq 2$	Set register idx_1, idx_2 and idx_3 to 1; Compare register $U1[k]$ with $U1[l]$	Idle
S_{14}	$U1[k] = U1[l]$	Update register idx_1 ; Load register r'	Idle
S_{15}	Clock, $U1[k] \neq U1[l]$	Update counter l	Idle
S_{16}	Clock	Set register $Flag$ to 0 and D_{min} to ∞	Idle
S_{17}	$idx_1 = 2$	Load register z	Idle
S_{18}	Clock	Activate counter it ; Set register x to 0.61 and y to 0	Idle
S_{19}	$z \geq 0$	Set register σ to 1	Idle
S_{20}	$z < 0$	Set register σ to -1	Idle
S_{21}	Clock, $it \leq it_{max}$	Update register x, y and z	Idle
S_{22}	Clock	Update counter it ; Check $Flag$ register content	Idle
S_{23}	$Flag = 0$	Set register $Flag$ to 1; Load register z ; Load register P_{11} and P_{12}	Idle
S_{24}	$Flag = 1$	Set register $Flag$ to 2; Load register z ; Load register P_{21} and P_{22}	Idle
S_{25}	$Flag = 2$	Set register $Flag$ to 3; Load register z ; Load register Q_{11} and Q_{12}	Idle
S_{26}	$Flag = 3$	Set register $Flag$ to 0; Activate counter m and n ; Load register Q_{21} and Q_{22}	Idle
S_{27}	Clock, $m \leq 2, n \leq 2$	Load register D_0, x_{temp} and y_{temp} ; Compare register D_0 with D_{min}	Idle
S_{28}	$D_0 \leq D_{min}$	Move register D_0 to D_{min} ; Move register x_{temp} to x_k ; Move register y_{temp} to y_k	Idle
S_{29}	Clock, $D_0 > D_{min}$	Update counter n	Idle
S_{30}	Clock	Update counter m	Idle
S_{31}	Clock	Update register idx_2 ; Load register V	Active
S_{32}	$idx_1 \neq 2$	Update register U and idx_3	Idle
S_{33}	Clock	Update counter k	Idle
S_{34}	Clock	Load register N_s ; Move register idx_3 to Ptr and R ; Check register N_s for Null	Idle
S_{35}	$N_s \neq 0, Ptr < 2$	Update counter t	Idle
S_{36}	$N_s = 0, t > t_{max}$	Halt the process	Sleep

Thus, a data transfer occurs to a specific sequential logic circuit/register at the next rising edge of the clock. The state register output is used to control the function of the multiplexer and selects the data desired in a particular operation.

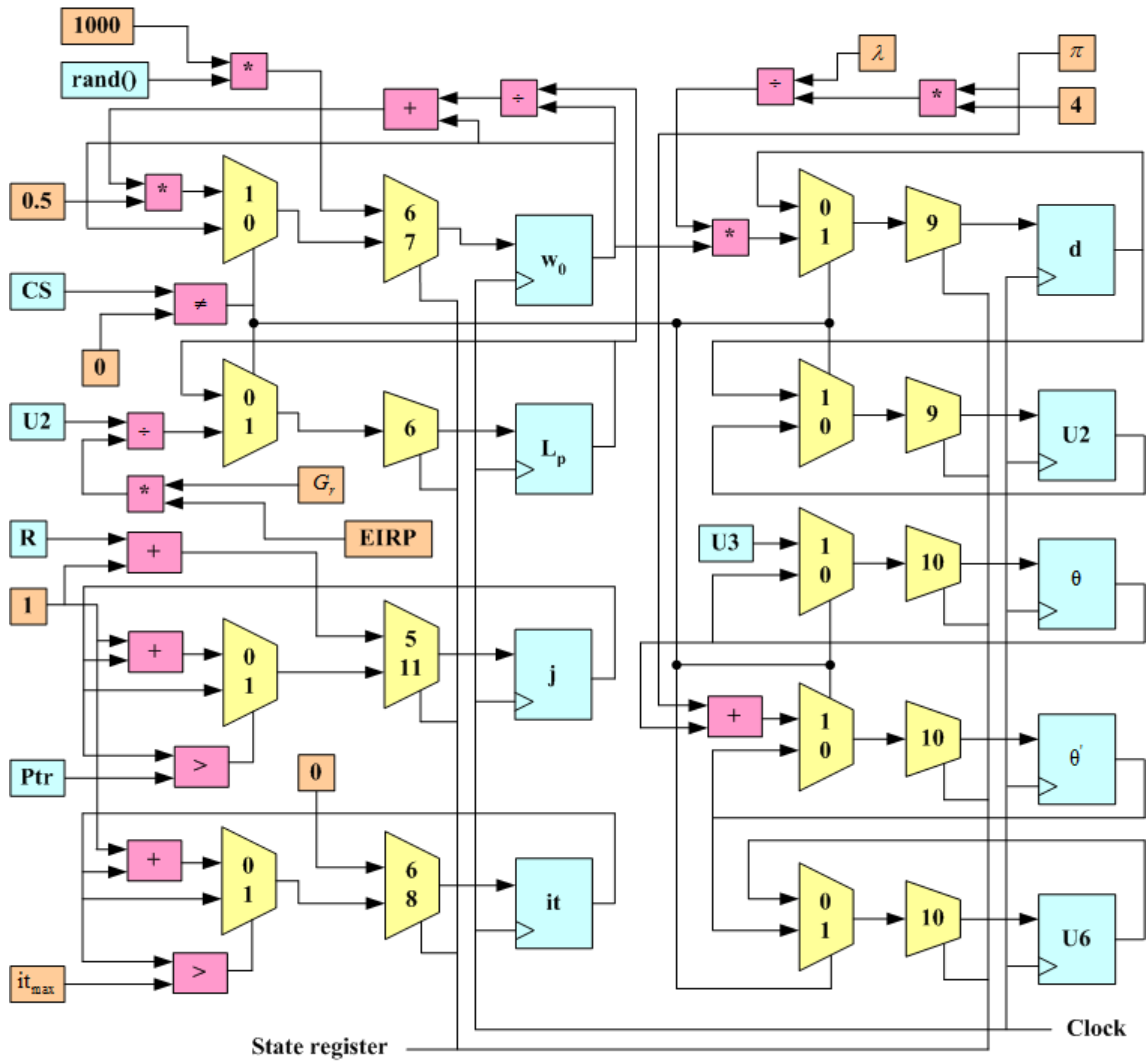


Fig. 8.9. RTL schematic of distance/angle (image) calculation module

Closed feedback loops preserve the current contents of the sequential logic circuits (registers) when they are not changed. Likewise, a multiplexing circuit (multiplexer, tri-state buffers) implements a data routing operation. It correctly transfers the desired data to a specific register in each state. Also, a 2×1 multiplexer executes any conditional data assignment statement (*if-then-else*) in which the select input contains a Boolean (relational/logical) expression. When the condition is satisfied, it passes the data from the true (logic 1) input port to the output port. Otherwise, it transfers the data from the false (logic 0) input port. Likewise, the execution of any loop statement (*for/while*) in the algorithm requires three sequential data operations: *preliminary data assignment*, *conditional data assignment*, and *increment/decrement data assignment*. Thus, we implemented a loop counter composed of three-level hardware architectures, such as

- a register (corresponding to the loop variable) which is loaded with the initial data and updates itself at each count,
- a 2×1 multiplexer (having a relational expression as a select input) to check the status of an update or keep the contents of the register unchanged, and
- a multiplexer (containing the output of the state register as a select input) which transfers the data to the sequential logic circuit/register in a suitable clock.

An FPGA chip stores all constants and parametric values of user-defined variables in Block Random Access Memory (BRAM).

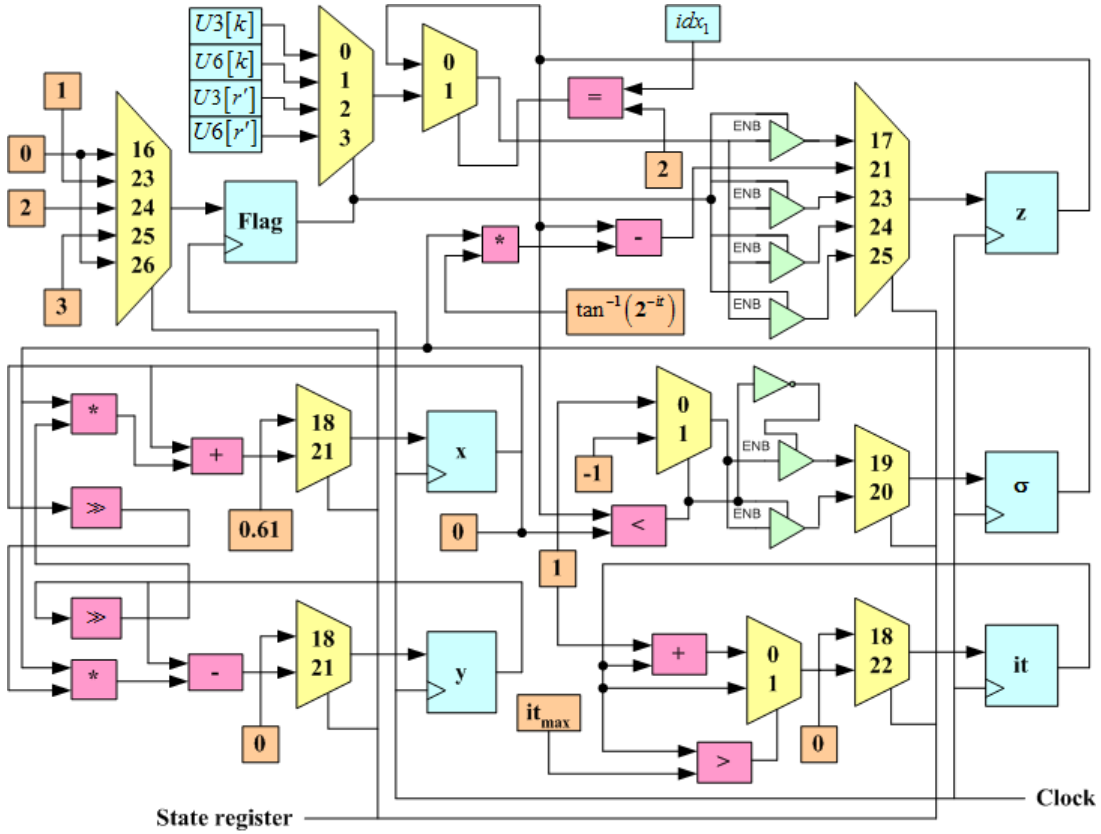


Fig. 8.10. RTL schematic of CORDIC evaluation module

We also implemented the random function, $rand()$ with a 16-bit Linear Feedback Shift Register (LFSR). We used 12-bit registers for performing all other data store/transfer operations and a lookup table to load the pre-computed value of the $arctan$ function for CORDIC evaluation.

8.5 Performance evaluation

The coprocessor must involve less hardware to compute the position of the nodes. It must also ensure the minimum of time, power, and error in the computation process. Therefore, proper benchmark functions are often desirable to evaluate its performance in the simulation environment.

8.5.1 Performance metrics

The performance of the coprocessor depends on the precision obtained in the localization of the nodes and on the complexity of its system (use of the hardware resources, computational overhead, and power consumption). Therefore, it is verified under four metrics as follows.

- *Localization accuracy* (Λ_l): It is a measure of the exactness of computing node positions based on information from RSS, AoA, and anchor references. In other words, it represents the number of deviations (ε) obtained between the estimated and actual positions of the nodes. Thus, we expressed the localization accuracy (in percentage), taking an average for all nodes as follows.

$$\Lambda_l = \left\{ \frac{\sum_{\forall k} (1 - \varepsilon_k)}{N_s} \right\} \times 100\% \quad (8.1)$$

Here, N_s is the total number of nodes in WSN and $k \in N_s$.

- *Hardware resource utilization* (v): It is a quantitative measure that indicates the number of resources used out of the total resources available in the FPGA device.

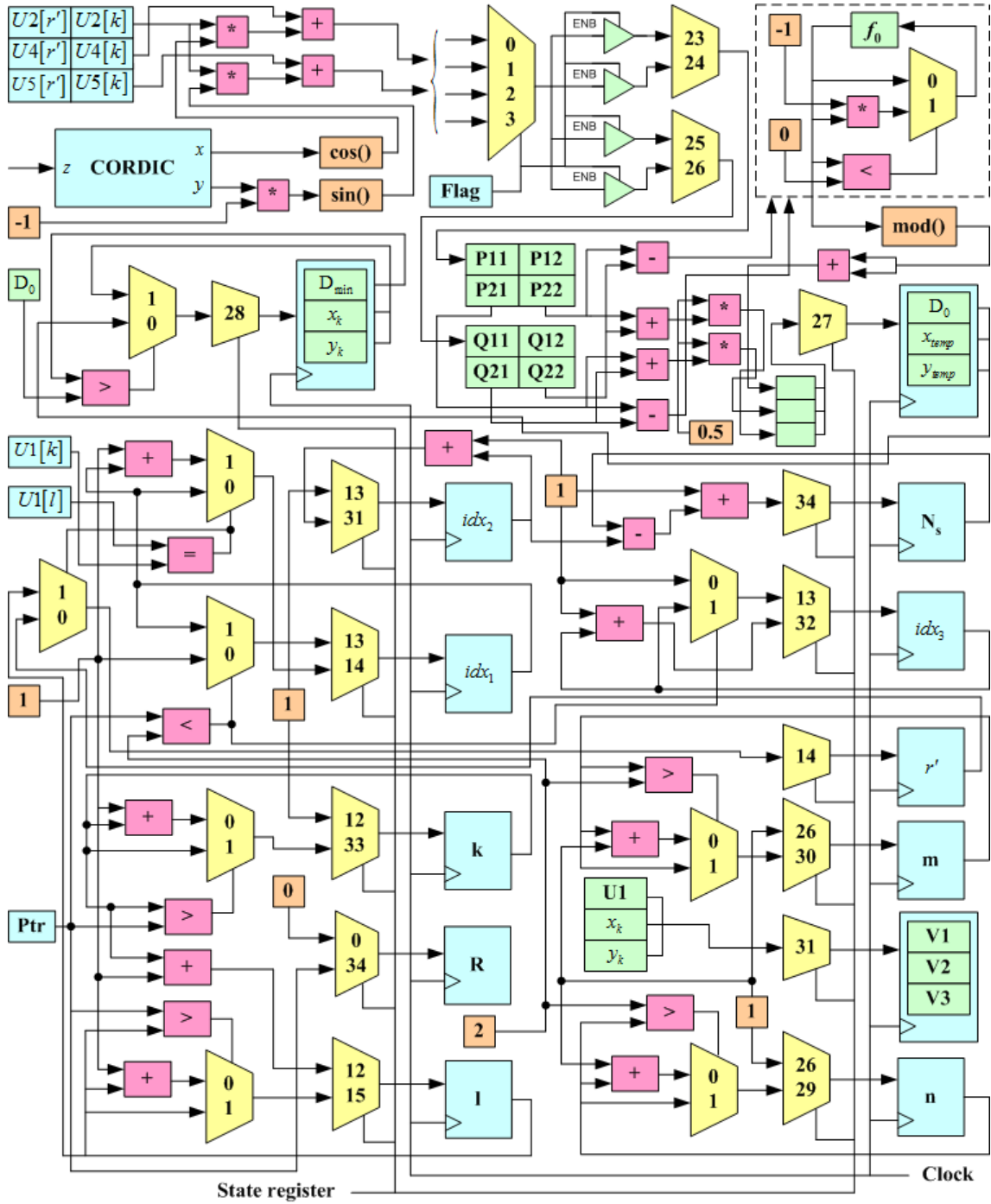


Fig. 8.11. RTL schematic of position computation module

Usually, it depends on the number of nodes, and we expressed the hardware resource utilization (in percentage) as the ratio of these two factors as in equation (7.17).

- *Computation time* (τ'_b): It measures the total time needed by the coprocessor to compute the positions of the nodes. It depends on the total number of clock cycles required until it finds all node IDs twice at distinct anchor points in the U register table. Therefore, we have expressed the computation time (in milliseconds) for an individual node position as follows.

$$\tau'_b = \frac{t_{con} N_c t_c}{N_s} \quad (8.2)$$

where, N_c , t_{con} , and t_c indicate the number of clock cycles needed for the computation in each anchor point, the total number of anchor points used, and the clock time, respectively.

- *Power consumption* (ρ_t): It is the measurement of total power dissipation in internal circuits, which arises in two forms: static power (caused by leakage currents) and dynamic power (caused by capacitive loads). It depends on the number of nodes, and we expressed the power consumption (in milliwatts) as the sum of these two factors as in equation (7.19).

8.5.2 Simulation environments

A set of simulations are performed on the Xilinx Virtex-5 FPGA board (device: XC5VFX30T, frequency: 122.64 MHz) for localization accuracy, FPGA resource utilization, power consumption, and computation time. Simulation parameters are chosen keeping compatibility with small wireless measurement systems such as the Crossbow MICAz mote which works with a 2.4 GHz RF transceiver and uses IEEE 802.15.4 MAC protocols. Table 8.2 shows all the simulation parameters for this work. Verilog programming code was used as a Hardware Description Language (HDL) that maps the design directly to the FPGA board.

Table 8.2: Simulation parameters

Parameters	Values
Network size	1000 m \times 1000 m
Number of nodes	50, 100, 150
Number of anchor	1
Maximum communication range	80 m, 90 m, 100 m
SNR threshold level	-40 dBm
FPGA board frequency	122.64 MHz
Maximum permissible anchor points	800
Maximum permissible iterations	10

8.5.3 Simulation results

The robustness of coprocessor performance is tested by performing simulations under different network conditions. For this purpose, the node density is modified by considering three distinct sets of nodes ($N_s = 50, 100,$ and 150) in the network. Besides, three sets of communication ranges ($d_{max} = 80, 90,$ and 100 meters) are chosen to measure the accuracy and speed of the localization process. Also, a total of 30 runs of the program are considered in each case to justify the results. Fig. 8.12(a), Fig. 8.12(b), and Fig. 8.12(c) show the Empirical Cumulative Distribution Function (ECDF) of average localization accuracy for different node densities. These plots illustrate that the accuracy does not vary significantly for a variation in the density of nodes. But, it increases with a reduction in the maximum communication range. This usually occurs because a reduction in communication range always keeps the value of RSS well above the SNR_{th} , which yields more precision in estimating distance and angle. Likewise, the Empirical Cumulative Distribution Function (ECDF) of anchor points required for localization under different node densities is shown in Fig. 8.13(a), Fig. 8.13(b), and Fig. 8.13(c). These plots also illustrate that fewer anchor points are needed in localization when it involves fewer nodes and higher communication range. This occurs because an increase in communication range often includes more nodes in the cluster formed at each anchor point. Thus, it speeds up the localization process. Table 8.3 provides the minimum and maximum levels of accuracy and the anchor points attained in each case. Data obtained from experiments on the Xilinx Virtex 5 (version: ISE 14.7) FPGA board verify the space and time complexities of the coprocessor. Table 8.4 and Table 8.5 show these data, respectively. It uses resources within an acceptable limit in all cases. It needs more resources for computation in higher node density. Here, the execution time is computed for minimum number of iterations used, and it also shows that it can perform with less time in any case. However, the computation time varies slightly with node density and communication range.

Table 8.6 gives the power consumption statistics. It requires low power which varies slightly with node density. The simulation results are compared to some well-known methods in the literature

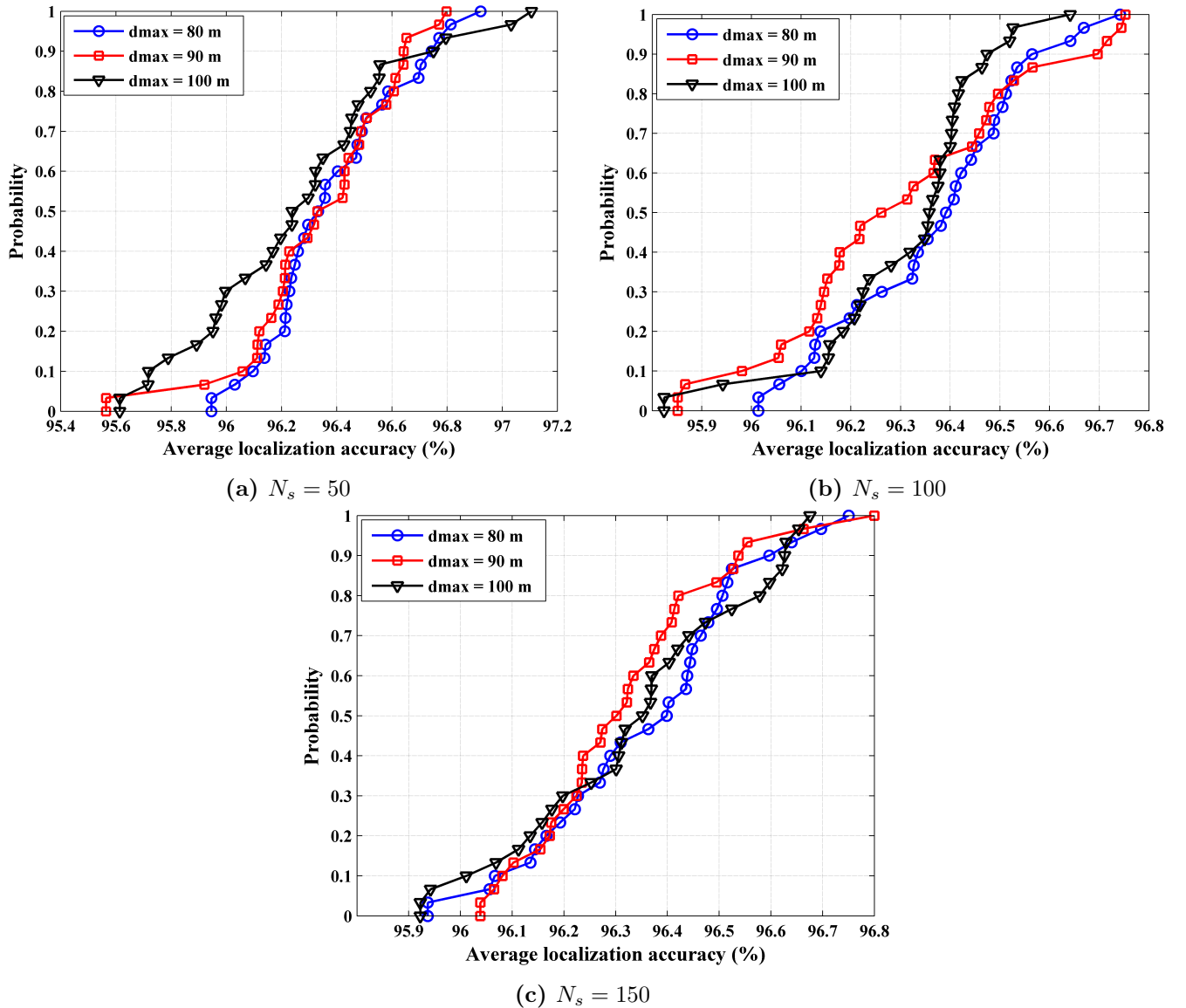


Fig. 8.12. Average localization accuracy

for localization accuracy, resource usage, computation time, and power consumption. These are provided in Table 8.7 and Table 8.8, respectively. It is evident that the coprocessor architecture often requires less execution time and yields greater localization accuracy. However, it requires much more hardware and more power than others. This occurs because we realized a centralized algorithm in which an anchor computes the positions of all nodes. It presents resource usage and power consumption in the computation for all nodes that will be more. Instead, the others used distributed algorithms that calculated the location of an individual node. Thus, they were always less numerous to implement on the node side.

8.5.4 Performance analysis

The time complexity of the coprocessor is $O(n)$. It is the sum of the complexities obtained in its four modules (e.g., cluster formation, distance/angle (image) calculation, CORDIC evaluation, and position computation). Here, the formation of the cluster depends on the number of neighboring nodes whose RSS exceeds the SNR_{th} . Thus, its complexity is $O(n)$. Likewise, the distance/angle (image) calculation depends on the value of the pointer to the data table U.

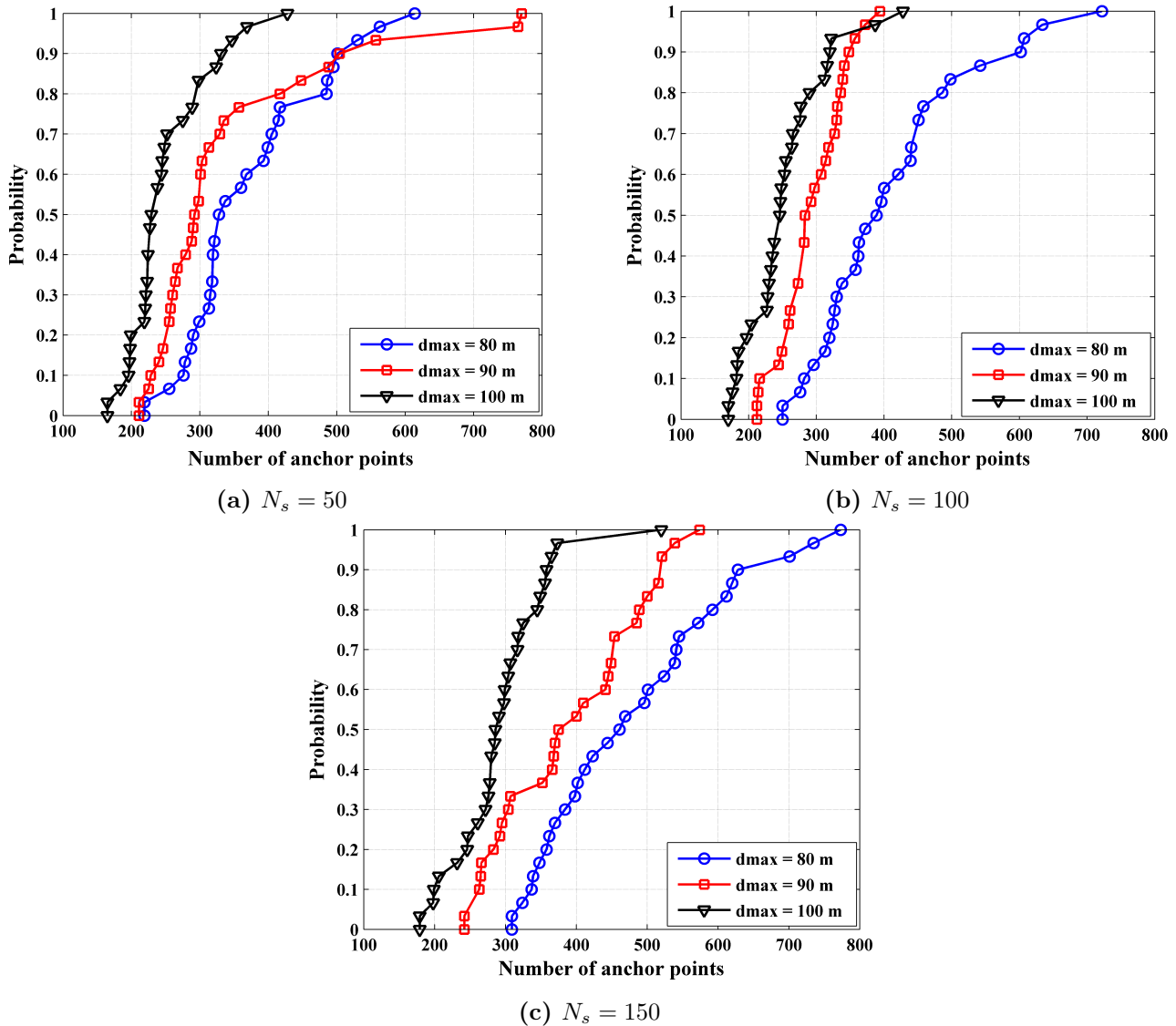


Fig. 8.13. Number of anchor points

Table 8.3: Accuracy and anchor points in the localization process

Number of nodes	Maximum communication range (m)	Accuracy (%)		Anchor points	
		Maximum	Minimum	Maximum	Minimum
50	80	96.9214	95.9458	614	219
	90	96.7978	95.5637	770	211
	100	97.1057	95.6138	428	165
100	80	96.7418	96.0135	722	250
	90	96.7523	95.8513	394	212
	100	96.6412	95.8240	428	170
150	80	96.7504	95.9366	773	309
	90	96.7999	96.0384	574	242
	100	96.6756	95.9224	520	179

And its complexity is also $O(n)$. The CORDIC algorithm generally converges after the 10-th iteration to compute the sine/cosine function and thus has constant complexity. The convergence of the position computation algorithm depends on the number of nodes encountered twice in the data table U or on the number of anchor points required. Therefore, it also has the complexity of $O(n)$.

Table 8.4: FPGA resource utilization

Parameters	Number of nodes		
	50	100	150
FFs	6712 out of 20480 (32 %)	12513 out of 20480 (61 %)	16897 out of 20480 (82 %)
LUTs	9773 out of 20480 (47 %)	19620 out of 20480 (95 %)	17080 out of 20480 (83 %)
Slices	3904 out of 5120 (76 %)	5098 out of 5120 (99 %)	5116 out of 5120 (99 %)
IOBs	11 out of 360 (3 %)	11 out of 360 (3 %)	11 out of 360(3 %)

Table 8.5: Computation time

Number of nodes	Maximum communication range (m)	Time elapsed (μ s)	Device frequency (MHz)	Clock cycles per anchor point	Total anchor points	Total clock cycles	Clock period (ns)	Computation time per node position (ms)
50	80	17.78	122.64	2181	219	477636	8.15	0.078
	90	3.88	122.64	476	211	100446	8.15	0.016
	100	3.52	122.64	432	165	71277	8.15	0.012
100	80	30.81	122.64	3779	250	944581	8.15	0.077
	90	10.34	122.64	1268	212	268706	8.15	0.022
	100	6.38	122.64	783	170	132961	8.15	0.011
150	80	32.56	122.64	3993	309	1233614	8.15	0.067
	90	12.09	122.64	1483	242	358866	8.15	0.019
	100	9.83	122.64	1206	179	215751	8.15	0.012

Table 8.6: Power consumption

Parameters	Number of nodes		
	50	100	150
Dynamic power (mW)	84.552	122.386	132.011
Static power (mW)	9.302	9.302	9.302
Total on-chip power (mW)	93.854	131.688	141.313

Table 8.7: Comparison of hardware resource utilization and localization accuracy

Localization systems	Resource utilization				IOBs	Accuracy (m)
	LUTs	FFs	Slices			
RSS/AoA hybrid method	9773 out of 20480 (47 %)	6712 out of 20480 (32%)	3904 out of 5120 (76 %)	11 out of 360 (3 %)	0.97	
ToA-CORDIC method [138]	219 out of 30720 (0.7 %)	54 out of 19200 (0.2 %)	108 out of 4800 (2.25 %)	...	0.83	
Least-squares method [139, 140]	...	3000	0.94	
Geometric method [141]	1371	594	772	30	0.60	

Table 8.8: Comparison of computation time and power consumption

Localization systems	Computation time (ms)	Power consumption (mW)
RSS/AoA hybrid method	0.012	93.85
Least-squares method [139, 140]	0.071	1.70
Centroid method [142]	0.072	1.45

8.6 Summary

In this chapter, we have proposed a coprocessor architecture for an anchor to locate nodes in WSNs. It works on a localization system where the anchor has random mobility. It computes the positions of all nodes centrally. It works in three phases: cluster formation, distance and angle (image) calculation, and position computation. By comparing the RSS of neighboring nodes with a prescribed threshold, it performs a clustering of nodes in each anchor point. Using the path loss model of signal propagation in free space, it estimates the distance to each clustering node. It then keeps all the estimated data with the anchor reference in a table. By iteratively checking this table and finding a few nodes in common with two distinct anchor points, it computes the positions of these nodes through the triangulation. The architecture is realized on a dedicated FPGA chip using FSM design technology and its performance has been verified with several fixed-point hardware-level simulations of a few benchmark metrics. It requires less time and yields higher computational accuracy for an RSS/AoA-based hybrid localization system. This is a more flexible design and can also be used in ToA/TDoA-based systems. However, this system is developed with a centralized algorithm. It requires higher hardware resources and power than distributed ones. Therefore, it is still possible to design a low-power architecture with minimal hardware resources in the future.

Chapter 9

Conclusion

9.1 Thesis outcomes

In this thesis, we studied the difficulties of localization systems in wireless sensor networks, focusing on attacks in the physical layer. Such attacks arise in the form of misbehavior in a few benevolent nodes or tampered radio environments in particular regions. The main objective of our study is to verify the operational feasibility and performance of localization systems based on smart antennas. We used the concept of adaptive beamforming and direction finding features to design the secure localization framework capable of detecting and eliminating malicious nodes from the network. For this purpose, and assuming resource constraints in the network, we considered the integration of smart antennas on a few mobile anchors. This is a new area of research at the nascent stage and has great potential for further development.

Chapter 2 presents a review of the most common security attacks and their effects on localization systems in wireless sensor networks. It includes a detailed description of all range-based secure localization algorithms developed over the past decades. It also contains a discussion of their application limits in the physical layer.

Chapter 3 presents the development of a robust localization system based on RSS and AoA using smart antennas, namely Row Matching Algorithm (RMA). It includes two new mobility control strategies (centroid and fuzzy). It also explains the robustness of the system to achieve significant improvement in results (localization accuracy and energy efficiency) through simulations under different anchor mobility models (e.g., random, centroid, and fuzzy).

Chapter 4 presents the development of a secure localization system against node capture attacks (Sybil and replay) using a two-step anchor mobility strategy via a centroid-based/fuzzy logic controller. It reports higher accuracy in locating benevolent nodes and a higher success rate in detecting malicious nodes.

Chapter 5 also presents the development of a secure localization system, namely Consistent Variant Assortment (CVA), against attacks by altering the radio environments in particular regions. It explains the efficiency of the system with simulation results compared to existing methods.

Chapter 6 presents the development of a security framework for anchor using antenna pattern control against PHY layer attacks altering GPS data. For a localization system, it describes the merits of protecting the privacy of anchors and relaying a pseudo-reference to estimate a node position with higher accuracy.

Chapter 7 presents the methodology for designing and implementing a reconfigurable beamforming architecture on an FPGA board. It describes the operational feasibility of smart antennas to easily integrate such an architecture with anchors requiring less hardware.

Chapter 8 also presents the methodology for designing and implementing a coprocessor architecture for anchors on an FPGA board. It describes the reduction of computational overhead in nodes since such an architecture computes the location of nodes centrally at a faster rate.

In summary, we listed the findings of this thesis as follows.

- The outcome of the literature survey on security attacks and existing localization systems for wireless sensor networks reveals that schemes using only cryptography or transmission power control and jamming cannot provide the complete solutions to improve localization accuracy in a corrupted radio environment or against attacks in the physical layer.
- The situation also becomes worse when adversaries eavesdrop to capture a few benevolent nodes and relay misleading information into the network by reprogramming them with malicious codes.
- The development of localization systems based on smart antennas is a new research concept that uses adaptive beamforming and direction finding features to isolate malicious nodes and secure the localization process in wireless sensor networks.
- The introduction of two new anchor mobility control schemes (centroid and fuzzy) is very useful to keep all nodes encountered first in a cluster to the succeeding cluster. Upon identifying any discrepancy between two successive clusters, such schemes iteratively separate benevolent nodes from malicious nodes. These methods are easier and more efficient in removing malicious nodes from the network.
- The proposal of a new scheme (CVA) is very effective in securing the localization process in a hostile radio environment. By dynamically adjusting the threshold level of the signal-to-noise ratio and the maximum permissible variation, such a scheme ensures the estimation of localization data (e.g., RSS, AoA, ToA, TDoA, etc.) from signals propagating on an identical radio. For any node, this method considers only the positions making consistent variations of all the estimations and removes those that are inconsistent. The simulation results show a significant improvement in energy consumption and localization accuracy.
- The proposal of a new scheme (antenna pattern control and pseudo-reference generation) is very worthwhile to preserve the privacy of the anchors in the network. By keeping adversaries outside the beam, such a scheme minimizes the chances of successful attacks on the anchor reference. Again, relaying pseudo-references to nodes ensures that data confidentiality is maintained. The simulation results show an acceptable success rate to protect the privacy of the anchors against attacks.
- The proposed implementation of reconfigurable beamforming architecture for smart antennas is very convenient to integrate with anchors. Such an architecture on dedicated FPGA hardware requires minimal resources and performs beamforming with acceptable accuracy.
- The proposed implementation of the coprocessor architecture for anchors is well suited to centrally compute node positions based on RSS and AoA information. Such an architecture on dedicated FPGA hardware performs the localization faster and improves network lifetime by avoiding computational overhead and reducing power consumption at resource-constrained nodes.

9.2 Future scope

We have also listed some possible modifications to the proposed systems to give it complete form in the future as follows.

- It is still possible to study the operational feasibility of secure localization systems based on smart antennas with real-time outdoor deployment of MICAz motes.
- The results presented in this thesis are based on a sparse and homogeneous network infrastructure with a finite number of nodes. Thus, it is still possible to study the scalability issues of the proposed systems on a dense and heterogeneous network infrastructure.
- The proposed architecture for the smart antenna beamformer and anchor coprocessor is implemented with a behavioral model in an FPGA design environment. Thus, it is always possible to verify the complexities of the system (space and time) and its adaptability with a parallel and pipelined model in an ASIC design environment.
- In all cases, system performance is evaluated for some common and simple attacks producing compromised nodes/anchors, and a compromised environment. Thus, it is always possible to verify its effectiveness against more complex attacks.
- The present research work only considers the basic form of PSO, CORDIC and fuzzy logic yielding more iterations to converge. Thus, it is always possible to improve the speed of convergence by using the most recent versions of such heuristic search methods.

9.3 Summary

This research work claims the merits of secure localization systems based on smart antennas in wireless sensor networks. The outcomes of this research proposal reveal an effective means to mitigate security issues in the physical layer. Integrating smart antennas with anchors is an effective way to restrict the negative impact of malicious nodes on the localization process. The simulation results show a significant improvement in energy efficiency and localization accuracy compared to existing methods. The success rate in detecting and blocking misbehaving nodes is more impressive compared to conventional secure localization systems. However, the simulation environment of this research only considers offline data generated on a personal computer. Thus, the feasibility and usefulness of the proposed schemes still need to be tested in real-time.

Bibliography

- [1] Culler D., Estrin D., and Srivastava M. “Guest Editors’ Introduction: Overview of Sensor Networks”. In: *Computer* 37.8 (2004), pp. 41–49. DOI: 10.1109/mc.2004.93.
- [2] Akyildiz I. F. and Su W. and Sankarasubramaniam Y. and Cayirci E. “A survey on sensor networks”. In: *IEEE Communications Magazine* 40.8 (2002), pp. 102–114. DOI: 10.1109/mcom.2002.1024422.
- [3] Heinzelman W. B., Chandrakasan A. P., and Balakrishnan H. “An application-specific protocol architecture for wireless microsensor networks”. In: *IEEE Transactions on Wireless Communications* 1.4 (2002), pp. 660–670. DOI: 10.1109/twc.2002.804190.
- [4] Zhao F. and Guibas L.J. *Wireless Sensor Networks: An information Processing Approach*. Elsevier, 2004.
- [5] Boukerche A., Ahmad M. Z., Turgut D., and Turgut B. “A taxonomy of routing protocols in sensor networks algorithms and protocols for wireless sensor networks”. In: *Algorithms and Protocols for Wireless Sensor Networks*. Ed. by A. Boukerche. Hoboken, New Jersey: John Wiley & Sons Inc., 2009, pp. 129–160.
- [6] Yick J., Mukherjee B., and Ghosal D. “Wireless sensor network survey”. In: *Computer Networks* 52.12 (2008), pp. 2292–2330. DOI: 10.1016/j.comnet.2008.04.002.
- [7] University of British Columbia. *MICAz Datasheet*. 2023. URL: http://courses.ece.ubc.ca/494/files/MICAz_Datasheet.pdf.
- [8] Bharathidasan A. and Ponduru V. A. S. *Sensor networks: An overview*. Tech. rep. Department of Computer Science: University of California, 2002.
- [9] Karlof C. and Wagner D. “Secure routing in wireless sensor networks: attacks and countermeasures”. In: *Ad Hoc Networks* 1.2-3 (2003), pp. 293–315. DOI: 10.1016/s1570-8705(03)00008-8.
- [10] Jiang J., Han G., Zhu C., Dong Y., and Zhang N. “Secure localization in wireless sensor networks: A survey”. In: *Journal of Communications* 6.6 (2011), pp. 460–470.
- [11] Boukerche A., Oliveira H. A. B. F., Nakamura E. F., and Loureiro A. A. F. “Secure localization algorithms for wireless sensor networks”. In: *IEEE Communications Magazine* 46.4 (2008), pp. 96–101. DOI: 10.1109/mcom.2008.4481347.
- [12] Gezici S. “A survey on wireless position estimation”. In: *Wireless Personal Communications* 44.3 (2007), pp. 263–282. DOI: 10.1007/s11277-007-9375-z.
- [13] Zia T. and Zomaya A. Y. “Security issues and countermeasures in wireless sensor networks”. In: *Algorithms and Protocols for Wireless Sensor Networks*. Ed. by A. Boukerche. Hoboken, New Jersey: John Wiley & Sons Inc., 2009, pp. 479–502.
- [14] El-badry R., Sultan A., and Youssef M. “HyberLoc: Providing physical layer location privacy in hybrid sensor networks”. In: *Proceedings of International Conference on Communications:IEEE*. Cape Town, South Africa, 2010, pp. 1–5.
- [15] Oh S. and Gruteser M. “Multi-node coordinated jamming for location privacy protection”. In: *Proceedings of the Military Communications Conference :IEEE*. Baltimore, USA, 2011, pp. 1243–1249.

- [16] Liberti J. C. and Rappaport T. S. *Smart antennas for wireless communications: IS-95 and third generation CDMA applications*. Prentice Hall, 1999.
- [17] Wang H., Wen Y., Lu Y., Zhao D., and Ji C. “Secure localization algorithms in wireless sensor networks: A review”. In: *Advances in Computer Communication and Computational Sciences*. Ed. by Bhatia S., Tiwari S., Mishra K., and Trivedi M. Singapore: Springer, 2019, pp. 543–553.
- [18] Zeng Y., Cao J., Hong J., Zhang S., and Xie L. “Secure localization and location verification in wireless sensor networks: a survey”. In: *The Journal of Supercomputing* 64.3 (2010), pp. 685–701. DOI: 10.1007/s11227-010-0501-4.
- [19] Boukerche A., Oliveira H. A. B. F., Nakamura E. F., and Loureiro A. A. F. “Localization systems for wireless sensor networks”. In: *IEEE Wireless Communications* 14.6 (2007), pp. 6–12. DOI: 10.1109/mwc.2007.4407221.
- [20] Bal M., Liu M., Shen W., and Ghenniwa H. “Localization in cooperative wireless sensor networks: A review”. In: *Proceedings of the 13th International Conference on Computer Supported Cooperative Work in Design :IEEE*. Santiago, Chile, 2009, pp. 438–443.
- [21] Winters J. H. “Smart antenna techniques and their application to wireless ad hoc networks”. In: *IEEE Wireless Communications* 13.4 (2006), pp. 77–83. DOI: 10.1109/mwc.2006.1678168.
- [22] Gross F.B. *Smart antennas for wireless communications*. McGraw-Hill Companies Inc., 2005.
- [23] Wang T. and Yang Y. “Location privacy protection from RSS localization system using antenna pattern synthesis”. In: *Proceedings of the International Conference on Computer Communications :IEEE*. Shanghai, China, 2011, pp. 2408–2416.
- [24] Godara L. C. *Smart antennas*. CRC Press, 2004.
- [25] Roy R. and Kailath T. “ESPRIT-estimation of signal parameters via rotational invariance techniques”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.7 (1989), pp. 984–995. DOI: 10.1109/29.32276.
- [26] Priyantha N. B., Balakrishnan H., Demaine E., and Teller S. *Anchor free distributed localization in sensor networks*. Tech. rep. Laboratory for Computer Science: MIT, 2003.
- [27] Kwon O-H. and Song H.J. “Localization through map stitching in wireless sensor networks”. In: *IEEE Transactions on Parallel and Distributed Systems* 19.1 (2008), pp. 93–105. DOI: 10.1109/tpds.2007.70706.
- [28] Shang Y., Ruml W., Zhang Y., and Fromherz M. P. J. “Localization from mere connectivity”. In: *Proceedings of the 4th International Symposium on Mobile Ad Hoc Networking & Computing :ACM*. Annapolis, Maryland, USA, 2003, pp. 201–212.
- [29] Sichitiu M.L. and Ramadurai V. “Localization of wireless sensor networks with a mobile beacon”. In: *Proceedings of the 1st International Conference on Mobile Ad-hoc and Sensor Systems :IEEE*. Fort Lauderdale, Florida, USA, 2004, pp. 174–183.
- [30] Ssu K.-F., Ou C.-H., and Jiau H. C. “Localization with mobile anchor points in wireless sensor networks”. In: *IEEE Transactions on Vehicular Technology* 54.3 (2005), IEEE Transactions on Vehicular Technology. DOI: 10.1109/tvt.2005.844642.
- [31] Yu G., Yu F., and Feng L. “A three dimensional localization algorithm using a mobile anchor node under wireless channel”. In: *Proceedings of the International Joint Conference on Neural Networks :IEEE*. Hong Kong, 2008, pp. 477–483.
- [32] Ou C.-H. “A localization scheme for wireless sensor networks using mobile anchors with directional antennas”. In: *IEEE Sensors Journal* 11.7 (2011), pp. 1607–1616. DOI: 10.1109/jsen.2010.2102748.
- [33] Zhang B. and Yu F. “LSWD: localization scheme for wireless sensor networks using directional antenna”. In: *IEEE Transactions on Consumer Electronics* 56.4 (2010), pp. 2208–2216. DOI: 10.1109/tce.2010.5681092.

- [34] Vivekanandan V. and Wong V. W. S. “Concentric anchor beacon localization algorithm for wireless sensor networks”. In: *IEEE Transactions on Vehicular Technology* 56.5 (2007), pp. 2733–2744. DOI: 10.1109/tvt.2007.899962.
- [35] Kim E. and Kim K. “Distance estimation with weighted least squares for mobile beacon-based localization in wireless sensor networks”. In: *IEEE Signal Processing Letters* 17.6 (2010), pp. 559–562. DOI: 10.1109/lsp.2010.2047169.
- [36] Bulusu N., Heidemann J., and Estrin D. “GPS-less low-cost outdoor localization for very small devices”. In: *IEEE Personal Communications* 7.5 (2000), pp. 28–34. DOI: 10.1109/98.878533.
- [37] Niculescu D. and Nath B. *Ad hoc positioning system (APS)*. Tech. rep. Rutgers University, 2001.
- [38] He T., Huang C., Blum B.M., Stankovic J.A., and Abdelzaher T. *Range-free localization schemes for large scale sensor networks*. Tech. rep. Computer Science Department: University of Virginia, 2003.
- [39] Nagpal R., Shrobe H., and Bachrach J. “Organizing a global coordinate system from local information on an ad hoc sensor network”. In: *Information Processing in Sensor Networks*. Ed. by F. Zhao and Guibas L. Berlin Heidelberg: Springer, 2003, pp. 333–348.
- [40] Ash J.N. and Potter L.C. “Sensor network localization via received signal strength measurements with directional antennas”. In: *Proceedings of the 42nd Annual Allerton Conference on Communication, Control, and Computing :Illinois*. Monticello, 2004, pp. 1861–1870.
- [41] Peng R. and Sichertiu M.L. “Angle of arrival localization for wireless sensor networks”. In: *Proceedings of the 3rd Annual Communications Society on Sensor and Ad Hoc Communications and Networks :IEEE*. Reston, VA, USA, 2006, pp. 374–382.
- [42] Kułakowski P., Vales-Alonso J., Egea-López E., Ludwin W., and García-Haro J. “Angle-of-arrival localization based on antenna arrays for wireless sensor networks”. In: *Computers & Electrical Engineering* 36.6 (2010), pp. 1181–1186. DOI: 10.1016/j.compeleceng.2010.03.007.
- [43] Niculescu D. and Nath B. “Ad hoc positioning system (APS) using AOA”. In: *Proceedings of the Twenty-Second Annual Joint Conference of the Computer and Communications Societies :IEEE*. San Francisco, CA, USA, 2003, pp. 1734–1743.
- [44] Doherty L., Pister K. S. J., and El Ghaoui L. “Convex position estimation in wireless sensor networks”. In: *Proceedings of the Twentieth Annual Joint Conference of the Computer and Communications Society :IEEE*. Anchorage, AK, USA, 2001, pp. 1655–1663.
- [45] Shang Y. and Ruml W. “Improved MDS-based localization”. In: *Proceedings of Twenty-third Annual Joint Conference of the Computer and Communications Societies :IEEE*. Hong Kong, China, 2004, pp. 2640–2651.
- [46] Shu J., Zhang R., Liu L., Wu Z., and Zhou Z. “Cluster-based three-dimensional localization algorithm for large scale wireless sensor networks”. In: *Journal of Computers* 4.7 (2009), pp. 585–592. DOI: 10.4304/jcp.4.7.585-592.
- [47] Sun C., Xing J., Ren Y., Liu Y., Sha J., and Sun J. “Distributed grid-based localization algorithm for mobile wireless sensor networks”. In: *Advances in Electronic Engineering, Communication and Management*. Ed. by Jin D. and Lin S. Berlin Heidelberg: Springer, 2012, pp. 315–321.
- [48] Xing J., Wang D., and Liu Y. “Distributed range-free localization algorithm for 3d wireless sensor networks under irregular radio propagation model”. In: *Applied Informatics and Communication*. Ed. by Zeng D. Berlin Heidelberg: Springer, 2011, pp. 299–306.
- [49] Tran D. A. and Nguyen T. “Localization in wireless sensor networks based on support vector machines”. In: *IEEE Transactions on Parallel and Distributed Systems* 19.7 (2008), pp. 981–994. DOI: 10.1109/tpds.2007.70800.
- [50] Nguyen X., Jordan M. I., and Sinopoli B. “A kernel-based learning approach to ad hoc sensor network localization”. In: *ACM Transactions on Sensor Networks* 1.1 (2005), pp. 134–152. DOI: 10.1145/1077391.1077397.

- [51] Pan J.J., Yang Q., and Pan J.S. “Online co-localization in indoor wireless networks by dimension reduction”. In: *Proceedings of the 22nd Conference on Artificial Intelligence and the 19th Innovative Applications of Artificial Intelligence Conference :AAAI*. Vancouver, Canada, 2007, pp. 1102–1107.
- [52] Bahl P. and Padmanabhan V. N. “RADAR: an in-building RF-based user location and tracking system”. In: *Proceedings of the Nineteenth Annual Joint Conference of the Computer and Communications Societies :IEEE*. Tel Aviv, Israel, 2000, pp. 775–784.
- [53] Priyantha N. B., Chakraborty A., and Balakrishnan H. “The cricket location-support system”. In: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking :ACM*. Boston, Massachusetts, United States, 2000, pp. 32–43.
- [54] Baggio A. and Langendoen K. “Monte Carlo localization for mobile wireless sensor networks”. In: *Ad Hoc Networks* 6.5 (2008), pp. 718–733. DOI: 10.1016/j.adhoc.2007.06.004.
- [55] Niculescu D. and Nath B. “DV based positioning in ad hoc networks”. In: *Telecommunication Systems* 22.1/4 (2003), pp. 267–280. DOI: 10.1023/a:1023403323460.
- [56] Savvides A., Park H., and Srivastava M.B. “The n-hop multilateration primitive for node localization problems”. In: *Mobile Networks and Applications* 8 (2003), pp. 443–451. DOI: 10.1023/A:1024544032357.
- [57] Čapkun S., Hamdi M., and Hubaux J.-P. “GPS-Free positioning in mobile ad-hoc networks”. In: *Cluster Computing* 5.2 (2002), pp. 157–167. DOI: 10.1023/a:1013933626682.
- [58] Malhotra N., Krasniewski M., Yang C., Bagchi S., and Chappell W. “Location estimation in ad hoc networks with directional antennas”. In: *Proceedings of the 25th International Conference on Distributed Computing Systems :IEEE*. Columbus, OH, USA, 2005, pp. 633–642.
- [59] Lazos L. and Poovendran R. “SeRLoc: Secure range-independent localization for wireless sensor networks”. In: *Proceedings of the 3rd Workshop on Wireless Security :ACM*. Philadelphia, Pennsylvania, USA, 2004, pp. 21–30.
- [60] Lazos L. and Poovendran R. “HiRLoc: high-resolution robust localization for wireless sensor networks”. In: *IEEE Journal on Selected Areas in Communications* 24.2 (2006), pp. 233–246. DOI: 10.1109/jsac.2005.861381.
- [61] Kucuk K. and Kavak A. “Scalable location estimation using smart antennas in wireless sensor networks”. In: *Ad Hoc Networks* 8.8 (2010), pp. 889–903. DOI: 10.1016/j.adhoc.2010.04.005.
- [62] Xiaorong Z., Yong W., and Hongbo Z. “A precise 2-D wireless localization technique using smart antenna”. In: *Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery :IEEE*. Huangshan, China, 2010, pp. 59–63.
- [63] Shan Z. and Yum T.-S. P. “Precise localization with smart antennas in ad-hoc networks”. In: *Proceedings of the Global Telecommunications Conference :IEEE*. Washington, DC, USA, 2007, pp. 1053–1057.
- [64] Anita E. A. M., Geetha R., and Kannan E. “A novel hybrid key management scheme for establishing secure communication in wireless sensor networks”. In: *Wireless Personal Communications* 82.3 (2015), pp. 1419–1433. DOI: 10.1007/s11277-015-2290-9.
- [65] Chakavarika T. T., Gupta S. K., and Chaurasia B. K. “Energy efficient key distribution and management scheme in wireless sensor networks”. In: *Wireless Personal Communications* 97.1 (2017), pp. 1059–1070. DOI: 10.1007/s11277-017-4551-2.
- [66] Choi J., Bang J., Kim L., Ahn M., and Kwon T. “Location-based key management strong against insider threats in wireless sensor networks”. In: *IEEE Systems Journal* 11.2 (2015), pp. 494–502. DOI: 10.1109/jsyst.2015.2422736.
- [67] Li L., Xu G., Jiao L., Li X., Wang H., Hu J., Xian H., Lian W., and Gao H. “A secure random key distribution scheme against node replication attacks in industrial wireless sensor systems”. In: *IEEE Transactions on Industrial Informatics* 16.3 (2019), pp. 2091–2101. DOI: 10.1109/tii.2019.2927296.

- [68] Mi Q., Stankovic J. A., and Stoleru R. “Practical and secure localization and key distribution for wireless sensor networks”. In: *Ad Hoc Networks* 10.6 (2012), pp. 946–961. DOI: 10.1016/j.adhoc.2011.12.008.
- [69] Sun X., Wu X., Huang C., Xu Z., and Zhong J. “Modified access polynomial based self-healing key management schemes with broadcast authentication and enhanced collusion resistance in wireless sensor networks”. In: *Ad Hoc Networks* 37.P2 (2016), pp. 324–336. DOI: 10.1016/j.adhoc.2015.08.027.
- [70] Delaët S., Mandal P. S., Rokicki M. A., and Tixeul S. “Deterministic secure positioning in wireless sensor networks”. In: *Theoretical Computer Science* 412.35 (2011), pp. 4471–4481. DOI: 10.1016/j.tcs.2011.04.010.
- [71] Conti M., Di Pietro R., Mancini L. V., and Mei A. “Distributed detection of clone attacks in wireless sensor networks”. In: *IEEE Transactions on Dependable and Secure Computing* 8.5 (2011), pp. 685–698. DOI: 10.1109/tdsc.2010.25.
- [72] Nguyen T.N., Le V.V., Chu S.I., Liu B.H., and Hsu Y.C. “Secure localization algorithms against localization attacks in wireless sensor networks”. In: *Wireless Personal Communications* 127 (2022), pp. 767–792. DOI: 10.1007/s11277-021-08404-4.
- [73] Zhu W. T., Xiang Y., Zhou J., Deng R. H., and Bao F. “Secure localization with attack detection in wireless sensor networks”. In: *International Journal of Information Security* 10.3 (2011), pp. 155–171. DOI: 10.1007/s10207-011-0127-4.
- [74] Han G., Jiang J., Shu L., Guizani M., and Nishio S. “A two-step secure localization for wireless sensor networks”. In: *The Computer Journal* 56.10 (2012), pp. 1154–1166. DOI: 10.1093/comjnl/bxr138.
- [75] Zhu B., Setia S., Jajodia S., Roy S., and Wang L. “Localized multicast: efficient and distributed replica detection in large-scale sensor networks”. In: *IEEE Transactions on Mobile Computing* 9.7 (2010), pp. 913–926. DOI: 10.1109/tmc.2010.40.
- [76] Zhang Z., Wu F., Jiang C., and Deng J. “An efficient detection scheme of node replication attacks for wireless sensor networks”. In: *International Journal of Security and Networks* 10.4 (2015), pp. 228–238. DOI: 10.1504/ijasn.2015.072440.
- [77] Jokhio S. H., Jokhio I. A., and Kemp A. H. “Node capture attack detection and defence in wireless sensor networks”. In: *IET Wireless Sensor Systems* 2.3 (2012), pp. 161–169. DOI: 10.1049/iet-wss.2011.0064.
- [78] Ho J-W., Wright M., and Das S.K. “Zone trust: fast zone-based node compromise detection and revocation in wireless sensor networks using sequential hypothesis testing”. In: *IEEE Transactions on Dependable and Secure Computing* 9.4 (2012), pp. 494–511. DOI: 10.1109/tdsc.2011.65.
- [79] Ho J-W., Wright M., and Das S.K. “Distributed detection of mobile malicious node attacks in wireless sensor networks”. In: *Ad Hoc Networks* 10.3 (2012), pp. 512–523. DOI: 10.1016/j.adhoc.2011.09.006.
- [80] Wei Y. and Guan Y. “Lightweight location verification algorithms for wireless sensor networks”. In: *IEEE Transactions on Parallel and Distributed Systems* 24.5 (2013), pp. 938–950. DOI: 10.1109/tpds.2012.42.
- [81] Basilico N., Gatti N., Monga M., and Sicari S. “Security games for node localization through verifiable multilateration”. In: *IEEE Transactions on Dependable and Secure Computing* 11.1 (2014), pp. 72–85. DOI: 10.1109/tdsc.2013.30.
- [82] Chiang J. T., Haas J. J., Choi J., and Hu Y.-C. “Secure location verification using simultaneous multilateration”. In: *IEEE Transactions on Wireless Communications* 11.2 (2012), pp. 584–591. DOI: 10.1109/twc.2011.120911.101147.
- [83] Jadliwala M., Zhong S., Upadhyaya S. J., Qiao C., and Hubaux J.-P. “Secure distance-based localization in the presence of cheating beacon nodes”. In: *IEEE Transactions on Mobile Computing* 9.6 (2010), pp. 810–823. DOI: 10.1109/tmc.2010.20.

- [84] Huang M. and Yu B. “SDN-based secure localization in heterogeneous WSN”. In: *Information and Communications Security*. Ed. by Qing S., Mitchell C., Chen L., and Liu D. Cham: Springer, 2018, pp. 276–287.
- [85] Giri A., Dutta S., and Neogy S. “Information-theoretic approach for secure localization against sybil attack in wireless sensor network”. In: *Journal of Ambient Intelligence and Humanized Computing* 12.10 (2020), pp. 9491–9497. DOI: 10.1007/s12652-020-02690-9.
- [86] Kumar G., Rai M. K., Kim H., and Saha R. “A secure localization approach using mutual authentication and insider node validation in wireless sensor networks”. In: *Mobile Information Systems* 2017 (2017), pp. 1–12. DOI: 10.1155/2017/3243570.
- [87] Esposito C. and Choi C. “Signaling game based strategy for secure positioning in wireless sensor networks”. In: *Pervasive and Mobile Computing* 40 (2017), pp. 611–627. DOI: 10.1016/j.pmcj.2017.06.025.
- [88] Liu X., Su S., Han F., Liu Y., and Pan Z. “A range-based secure localization algorithm for wireless sensor networks”. In: *IEEE Sensors Journal* 19.2 (2018), pp. 785–796. DOI: 10.1109/jsen.2018.2877306.
- [89] Das S. K. and Ho J.-W. “A synopsis on node compromise detection in wireless sensor networks using sequential analysis (Invited Review Article)”. In: *Computer Communications* 34.17 (2011), pp. 2003–2012. DOI: 10.1016/j.comcom.2011.07.004.
- [90] Garcia-Alfaro J., Barbeau M., and Kranakis E. “Secure geolocalization of wireless sensor nodes in the presence of misbehaving anchor nodes”. In: *Annals of Telecommunications* 66.9-10 (2011), pp. 535–552. DOI: 10.1007/s12243-010-0221-z.
- [91] Garg R., Varna A. L., and Wu M. “An efficient gradient descent approach to secure localization in resource constrained wireless sensor networks”. In: *IEEE Transactions on Information Forensics and Security* 7.2 (2012), pp. 717–730. DOI: 10.1109/tifs.2012.2184094.
- [92] Liu C., Yao X., and Luo J. “Multiregional secure localization using compressive sensing in wireless sensor networks”. In: *ETRI Journal* 41.6 (2019), pp. 739–749. DOI: 10.4218/etrij.2017-0116.
- [93] Xue W.-c., Peng B., Wang S.-h., and Hua Y. “A type of energy-efficient secure localization algorithm FM based in dynamic sensor networks”. In: *EURASIP Journal on Wireless Communications and Networking* 39 (2020), pp. 1–7. DOI: 10.1186/s13638-020-1658-z.
- [94] Ssu K.-F., Wang W.-T., and Chang W.-C. “Detecting sybil attacks in wireless sensor networks using neighboring information”. In: *Computer Networks* 53.18 (2009), pp. 3042–3056. DOI: 10.1016/j.comnet.2009.07.013.
- [95] Sarigiannidis P., Karapistoli E., and Economides A. A. “Detecting sybil attacks in wireless sensor networks using UWB ranging-based information”. In: *Expert Systems with Applications* 42.21 (2015), pp. 7560–7572. DOI: 10.1016/j.eswa.2015.05.057.
- [96] Perazzo P., Taponecco L., D’amico A. A., and Dini G. “Secure positioning in wireless sensor networks through enlargement miscontrol detection”. In: *ACM Transactions on Sensor Networks* 12.4 (2016), pp. 1–32. DOI: 10.1145/2943782.
- [97] Compagno A., Conti M., D’Amico A. A., Dini G., Perazzo P., and Taponecco L. “Modeling enlargement attacks against UWB distance bounding protocols”. In: *IEEE Transactions on Information Forensics and Security* 11.7 (2016), pp. 1565–1577. DOI: 10.1109/tifs.2016.2541613.
- [98] Wang C. and Xiao L. “Sensor localization under limited measurement capabilities”. In: *IEEE Network* 21.3 (2007), pp. 16–23. DOI: 10.1109/mnet.2007.364254.
- [99] Sabitha R. and Thyagarajan T. “Fuzzy logic-based transmission power control algorithm for energy efficient MAC protocol in wireless sensor networks”. In: *International Journal of Communication Networks and Distributed Systems* 9.3/4 (2012), pp. 247–265. DOI: 10.1504/ijcnds.2012.048873.
- [100] University of the Mediterranean of Reggio Calabria. *IEEE 802.15.4 -2006 standard*. 2023. URL: https://www.unirc.it/documentazione/materiale_didattico/599_2009_192_7229.pdf.

- [101] Xiao B., Chen H., and Zhou S. “Distributed localization using a moving beacon in wireless sensor networks”. In: *IEEE Transactions on Parallel and Distributed Systems* 19.5 (2008), pp. 587–600. DOI: 10.1109/tpds.2007.70773.
- [102] Civicioglu P. “Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm”. In: *Computers & Geosciences* 46 (2008), pp. 229–247. DOI: 10.1016/j.cageo.2011.12.011.
- [103] Chandran J.J.G. and Victor S.P. “Optimized energy efficient localization technique in mobile sensor networks”. In: *IACSIT International Journal of Engineering and Technology* 2.2 (2010), pp. 149–156.
- [104] Mendel J. M. “Fuzzy logic systems for engineering: a tutorial”. In: *Proceedings of the IEEE* 83.3 (1995), pp. 345–377. DOI: 10.1109/5.364485.
- [105] Alakhras M., Oussalah M., and Hussein M. “A survey of fuzzy logic in wireless localization”. In: *EURASIP Journal on Wireless Communications and Networking* 89.2020 (2020), pp. 1–45. DOI: 10.1186/s13638-020-01703-7.
- [106] Prieto J., Mazuelas S., Bahillo A., Fernandez P., Lorenzo R. M., and Abril E. J. “Adaptive data fusion for wireless localization in harsh environments”. In: *IEEE Transactions on Signal Processing* 60.4 (2012), pp. 1585–1596. DOI: 10.1109/tsp.2012.2183126.
- [107] Li Y. and Ren J. “Source-location privacy through dynamic routing in wireless sensor networks”. In: *Proceedings of the 29th Conference on Computer Communications :IEEE*. San Diego, CA, USA, 2010, pp. 1–9.
- [108] Myles G., Friday A., and Davies N. “Preserving privacy in environments with location-based applications”. In: *IEEE Pervasive Computing* 2.1 (2003), pp. 56–64. DOI: 10.1109/mprv.2003.1186726.
- [109] Yilmaz M. H. and Arslan H. “A survey: Spoofing attacks in physical layer security”. In: *Proceedings of the 40th Local Computer Networks Conference Workshops :IEEE*. Clearwater Beach, FL, USA, 2015, pp. 812–817.
- [110] Godara L. C. “Application of antenna arrays to mobile communications, part II: Beam-forming and direction-of-arrival considerations”. In: *Proceedings of the IEEE* 85.8 (1997), pp. 1195–1245. DOI: 10.1109/5.622504.
- [111] Zardi F., Nayeri P., Rocca P., and Haupt R.L. “Artificial intelligence for adaptive and reconfigurable antenna arrays- A review”. In: *IEEE Antennas and Propagation Magazine* 63.3 (2020), pp. 28–38. DOI: 10.1109/map.2020.3036097.
- [112] Gies D. and Rahmat-Samii Y. “Particle swarm optimization for reconfigurable phase- differentiated array design”. In: *Microwave and Optical Technology Letters* 38.3 (2003), pp. 168–175. DOI: 10.1002/mop.11005.
- [113] Clerc M. and Kennedy J. “The particle swarm - explosion, stability, and convergence in a multidimensional complex space”. In: *IEEE Transactions on Evolutionary Computation* 6.1 (2002), pp. 58–73. DOI: 10.1109/4235.985692.
- [114] Robinson J. and Rahmat-Samii Y. “Particle swarm optimization in electromagnetics”. In: *IEEE Transactions on Antennas and Propagation* 52.2 (2004), pp. 397–407. DOI: 10.1109/tap.2004.823969.
- [115] Balanis C. A. *Antenna theory: analysis and design*. John Wiley and Sons Inc., 2005.
- [116] Zurita L.N.R. “Optimising multiple antenna techniques for physical layer security”. PhD thesis. The University of Leeds, 2014.
- [117] Zhu J. “Physical layer security in massive MIMO systems”. PhD thesis. The University of British Columbia, 2016.

- [118] Hong Y.-W. P., Lan P.-C., and Kuo C.-C. J. “Enhancing physical-layer secrecy in modern wireless communication systems”. In: *Signal Processing Approaches to Secure Physical Layer Communications in Multi-Antenna Wireless Systems*. Ed. by Gan W.-S. and Kuo C.-C. J. Singapore: Springer, 2014, pp. 125–132.
- [119] Jayaprakasam S., Abdul Rahim S. K., Leow C. Y., Ting T. O., and Eteng A. A. “Multiobjective beam pattern optimization in collaborative beamforming via NSGA-II with selective distance”. In: *IEEE Transactions on Antennas and Propagation* 65.5 (2017), pp. 2348–2357. DOI: 10.1109/tap.2017.2684187.
- [120] Razavilar J., Rashid-Farrokhi F., and Liu K. J. R. “Software radio architecture with smart antennas: a tutorial on algorithms and complexity”. In: *IEEE Journal on Selected Areas in Communications* 17.4 (1999), pp. 662–676. DOI: 10.1109/49.761043.
- [121] Dikmese S., Kavak A., Kucuk K., Sahin S., and Tangel A. “FPGA based implementation and comparison of beamformers for CDMA 2000”. In: *Wireless Personal Communications* 57 (2011), pp. 233–253. DOI: 10.1007/s11277-009-9855-4.
- [122] Nuteson T. W., Stocker J. E., Clark J. S., Haque D. S., and Mitchell G. S. “Performance characterization of FPGA techniques for calibration and beamforming in smart antenna applications”. In: *IEEE Transactions on Microwave Theory and Techniques* 50.12 (2002), pp. 3043–3051. DOI: 10.1109/tmtt.2002.805151.
- [123] Dikmese S., Kavak A., Kucuk K., Sahin S., Tangel A., and Dincer H. “Digital signal processor against field programmable gate array implementations of space-code correlator beamformer for smart antennas”. In: *IET Microwaves, Antennas & Propagation* 4.5 (2010), pp. 593–599. DOI: 10.1049/iet-map.2009.0151.
- [124] Choi S. and Shim D. “A novel adaptive beamforming algorithm for a smart antenna system in a CDMA mobile communication environment”. In: *IEEE Transactions on Vehicular Technology* 49.5 (2000), pp. 1793–1806. DOI: 10.1109/25.892584.
- [125] Kucuk K., Kavak A., Karakoc M., Yigit H., and Ozdemir C. “A practical space-code correlator receiver for DSP based software radio implementation in CDMA2000”. In: *Wireless Personal Communications* 49.2 (2009), pp. 245–261. DOI: 10.1007/s11277-008-9570-6.
- [126] Thiripurasundari C., Sumathy V., and Thiruvengadam C. “An FPGA implementation of novel smart antenna algorithm in tracking systems for smart cities”. In: *Computers & Electrical Engineering* 65 (2017), pp. 59–66. DOI: 10.1016/j.compeleceng.2017.06.009.
- [127] Grout I. *Digital systems design with FPGAs and CPLDs*. Elsevier, 2008.
- [128] Herve N., Menard D., and Sentieys O. “Data wordlength optimization for FPGA synthesis”. In: *Proceedings of Workshop on Signal Processing Systems Design and Implementation :IEEE*. Athens, Greece, 2005, pp. 623–628.
- [129] Vanderbauwhede W., Chalamalasetti S. R., and Margala M. “High-performance FPGA-accelerated real-time search”. In: *High-Performance Computing Using FPGAs*. Ed. by Vanderbauwhede W. and Benkrid K. New York: Springer, 2013, pp. 209–244.
- [130] Farooq U., Marrakchi Z., and Mehrez H. *Tree-Based Heterogeneous FPGA Architectures*. Springer, 2012.
- [131] Meher P. K., Valls J., Juang T.-B., Sridharan K., and Maharatna K. “50 years of CORDIC: algorithms, architectures, and applications”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 56.9 (2009), pp. 1893–1907. DOI: 10.1109/tcsi.2009.2025803.
- [132] Chu P.P. *FPGA prototyping by Verilog examples*. John Wiley & Sons Inc., 2008.
- [133] Vahid F. *Digital design with RTL design, VHDL, and Verilog*. John Wiley & Sons Inc., 2012.
- [134] Dick C., Harris F., Pajic M., and Vuletic D. “Real-time QRD-based beamforming on an FPGA platform”. In: *Proceedings of the Fortieth Asilomar Conference on Signals, Systems and Computers :IEEE*. Pacific Grove, CA, USA, 2006, pp. 1200–1204.

- [135] Jarrah A. and Jamali M. "Software tool for efficient FPGA design of direct data domain approach for space-time adaptive processing". In: *Electronics Letters* 49.13 (2013), pp. 789–791. DOI: 10.1049/e1.2013.1307.
- [136] Hasanikhah N., Amin-Nejad S., Darvish G., and Moniri M. R. "Efficient implementation of space-time adaptive processing for adaptive weights calculation based on floating point FPGAs". In: *The Journal of Supercomputing* 74.7 (2018), pp. 3193–3210. DOI: 10.1007/s11227-018-2369-7.
- [137] Hasanikhah N., Amin-Nejad S., Darvish G., and Moniri M. R. "Comparison of practical methods for an efficient FPGA implementation of STAP". In: *International Journal of Electronics* 106.8 (2019), pp. 1113–1126. DOI: 10.1080/00207217.2018.1553247.
- [138] Zaidi M., Bouazzi I., Al-Rayif M.I., Shamim M.Z.M., and Usman M. "Low power hardware design and its mathematical modeling for fast-exact geolocalization system in wireless networks". In: *International Journal of Communication Systems* 35.9 (2022), pp. 1–16. DOI: 10.1002/dac.5128.
- [139] Karalar T. C., Yamashita S., Sheets M., and Rabaey J. "A low power localization architecture and system for wireless sensor networks". In: *Proceedings of Workshop on Signal Processing Systems :IEEE*. Austin, Texas, USA, 2004, pp. 89–94.
- [140] Karalar T. C., Yamashita S., Sheets M., and Rabaey J. "An integrated, low power localization system for sensor networks". In: *Proceedings of The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services :IEEE*. Boston, MA, USA, 2004, pp. 1–7.
- [141] Zaidi M., Ouni R., Bhar J., and Tourki R. "A novel positioning technique with low complexity in wireless LAN: Hardware implementation". In: *Proceedings of the World Congress on Engineering :IAENG*. London, U.K, 2011, pp. 1–7.
- [142] Oliveira L. L., Dessbesell G. F., Martins J. B., and Monteiro J. "Hardware implementation of a centroid-based localization algorithm for mobile sensor networks". In: *Proceedings of International Symposium of Circuits and Systems :IEEE*. Rio de Janeiro, Brazil, 2011, pp. 2829–2832.

Rathindranath Ramesh
15/06/2023