

American University in Cairo

AUC Knowledge Fountain

Theses and Dissertations

Student Research

Fall 1-6-2020

Development of IoT based hybrid autonomous network robots (ANR)

Chimsom Isidore Chukwuemeka
The American University in Cairo

Follow this and additional works at: <https://fount.aucegypt.edu/etds>



Part of the [Robotics Commons](#)

Recommended Citation

APA Citation

Chukwuemeka, C. I. (2020). *Development of IoT based hybrid autonomous network robots (ANR)* [Master's Thesis, the American University in Cairo]. AUC Knowledge Fountain.

<https://fount.aucegypt.edu/etds/1742>

MLA Citation

Chukwuemeka, Chimsom Isidore. *Development of IoT based hybrid autonomous network robots (ANR)*. 2020. American University in Cairo, Master's Thesis. *AUC Knowledge Fountain*.

<https://fount.aucegypt.edu/etds/1742>

This Master's Thesis is brought to you for free and open access by the Student Research at AUC Knowledge Fountain. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AUC Knowledge Fountain. For more information, please contact thesisadmin@aucegypt.edu.



THE AMERICAN UNIVERSITY IN CAIRO

الجامعة الأمريكية بالقاهرة

**School of Sciences and Engineering
Robotics, Control and Smart Systems (RCSS)**

**DEVELOPMENT OF IOT BASED HYBRID
AUTONOMOUS NETWORK ROBOTS (ANR)**

By

Chimsom Isidore Chukwuemeka

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Robotics, Control and Smart Systems (RCSS)

Under the supervision of:

Dr. Maki K. Habib

Prof., Department of Mechanical Engineering

December 2019

DEDICATION

This thesis is dedicated to my dad who passed on January 27th, 2019. Rest well, Repe!

ACKNOWLEDGEMENT

I am most grateful to God for His grace and mercy that have sustained me thus far.

I am very grateful to my family that has sustained me emotionally, spiritually and materially during this Master's program. Thank you very much for being my rock-solid, dependable support.

I want to say a big thank you to the RCSS program director and my supervisor, Prof. Dr. Maki for his patience, support and encouragement during this Master's program and especially during this thesis. Thank you very much, sir.

I acknowledge the support of the Office of the Dean of Graduates, especially Dr. Ramadan and Ms. Aya Morsi. Thank you very much for all you do for graduate students.

I am grateful for the graduate student clubs, GSA and AHA that provided an exciting and supportive graduate community for my development during this Master's program.

I appreciate all the departments in AUC that supported me during this Master's, mainly the department of Mechanical Engineering, Archives and the CLT. Thank you for the experience; it was a great pleasure working as a student under the AUC work-study program.

Finally, I sincerely appreciate the help and support of many friends and acquaintances I made during my stay at AUC. Thank you, Amal, Sherif, Lana, Doreen, Joel, Fahad, Samuel, etc. I love and appreciate you all.

ABSTRACT

The integration of wireless sensor networks (WSNs) and multirobot systems (MRS) represents an active research area supporting a wide range of applications. This is because it enables ubiquitous applications due to the robots' mobility and detection capabilities associated with its deployment. These systems have many benefits, such as perception with extended coverage that facilitate wider exploration and surveillance, efficiency in data routing, effective and reliable task environment management, etc. However, integrating two fields of research means dealing with a range of challenges such as using effective architecture for WSNs and MRS, efficient communication protocols within a network of sensors nodes and robots, cost of sensor network deployment, mobile robots coverage and deployment, robots' navigation and cooperation.

This thesis presents the development of an autonomous networked robots (ANR) system featuring WSNs and MRS for event detection and surveillance of a large operational environment with IoT/cloud integration for data storage and analytics. The operational environment is divided into zones and each zone comprises static sensor nodes (SSNs), a mobile robot with onboard static sensor nodes (O-SSNs) and other sensors supporting its navigation. The zone mobile robot with its sensors is called the mobile robot sensor nodes (MRSNs). Additional MRSNs are used to navigate the spaces not covered by a zone and they are called global-MRSNs (G-MRSNs). G-MRSNs are identical to MRSNs but with different task assignments. The MRSN and G-MRSN communicate their data and the data from assigned spaces to the IoT BS through the base station gateway (BSG). A simulation development environment (SDE) is developed to facilitate the development, testing and validation of the different components of the system.

The developed ANR system has a hybrid architecture with a two-tier communication network that enables the system simulation, hardware-in-the-loop (HIL) based simulation and real-time operation with results aggregation and visualization at the IoT base station (BS) graphic user interfaces (GUIs) and the integration of two cloud platforms whereby remote users can access the results for further processing and analytics. Hence, the hybrid architecture enables concurrent operation and integration between all system components in the simulation, HIL based simulation and real-time operation whereby the SSNs within zones detect and characterize events using a fuzzy logic decision making system and communicated them with possible alerts to the zone MRSNs through ZigBee network modules. When no alert is issued, the MRSNs and G-MRSN navigate the assigned paths within their spaces, while relaying detected information by the SSNs

and O-SSNs through a wireless serial network to the IoT BS. In case one of the SSNs issued an alert, the zone MRSN relays the event to the IoT BS and starts to generate and navigate its path to the alerting SSN using the A* algorithm. If the zone MRSN is outside a reference distance to the alerting SSN location, it coordinates with the G-MRSN using contract net protocol (CNP) algorithm to assign the closer G-MRSN, generate the path to the alerting SSN and navigate it.

The developed hybrid ANR architecture was implemented with all of its software and hardware components, and tested on the different levels i.e. the simulation, HIL based simulation and real-time operation, through the use of a selected factory type task environment representing four zones with its connected spaces. The robots' navigation paths are tracked and visualized on the MRS GUI, while the detection sensory data (temperature/RH, carbon monoxide gas, smoke, fire and motion) from the zones are aggregated by the robots and received at the IoT BS, updated and visualized on the SND GUI. From the SND GUI, the sensory information and the event classification are aggregated to the MS cloud platform for storage and basic analytics, while they are directly sent to the ThingSpeak cloud platform for advance analytics by the BSG. The ThingSpeak cloud data analytics involves the evaluation of various operational conditions within each zone such as temperature/relative humidity using regression analysis. Furthermore, it involves using a trained neural network for preservation metrics prediction per zone whereby the duration of storage and type of materials to be stored within the zones are determined.

The developed hybrid ANR system was tested at all levels: the simulation, HIL based simulation and hardware implementation. Full hardware implementation of the system components was physically assembled and tested in a lab environment. The overall results showed a good performance and validated the concept of the developed hybrid architecture supporting the integration and the concurrent operation of all developed system components.

TABLE OF CONTENTS

DEDICATION.....	i
ACKNOWLEDGEMENT.....	ii
ABSTRACT	iii
TABLE OF CONTENTS	v
LIST OF FIGURES	viii
LIST OF TABLES	xii
NOMENCLATURE.....	xiv
Chapter 1	1
INTRODUCTION	1
1.1 Study Background.....	2
1.1.1 Multirobot Systems (MRS)	2
1.1.2 Wireless Sensor Networks (WSNs)	3
1.1.3 Autonomous Networked Robotics (ANR)	4
1.2 Statement of Purpose	4
1.3 Objectives and Expected Outcomes.....	5
1.4 Expected Contributions.....	6
1.5 Thesis Structure	6
Chapter 2	7
LITERATURE REVIEW.....	7
2.1 Related Work in MRS	8
2.1.1 MRS Architectures.....	8
2.1.2 MRS Communication	10
2.1.3 MRS Coordination and Cooperation	12
2.1.4 MRS Path Planning (MRPP) and Navigation	13
2.1.5 MRS Task Allocation (MRTA).....	15
2.2 Related Works in WSNs	18
2.2.1 Major Features of WSNs.....	19
2.2.2 Overview of IEEE 802.15.4/ZigBee Communication Standard.....	22
2.2.3 Overview of Sub-1Ghz RF Standards.....	24
2.2.4 Related Works of WSNs in Surveillance	25
2.3 Related Works in ANR (MRS Integration with WSNs)	27
2.3.1 Related Works in General ANR Systems	27
2.3.2 Related Works of ANR Systems in Surveillance	27
2.4 Research Challenges and Open Areas.....	28
2.4.1 Challenges in MRS for Surveillance Applications	29
2.4.2 Challenges in WSN for Surveillance Applications.....	29
Chapter 3	31
SYSTEM REQUIREMENTS AND RESEARCH METHODOLOGY	31
3.1 Scope of Research.....	31
3.2 System Operational Scenario.....	32
3.3 Functional Requirements, Assumptions and Constraints.....	36
3.3.1 Functional Requirements.....	36
3.3.2 Assumptions	37

3.3.3	Constraints	37
3.4	Research Methodology	38
3.4.1	ANR System Conceptualization	38
3.4.2	ANR System Design Overview	38
3.4.3	ANR System Development	38
Chapter 4	41
ANR SYSTEM ARCHITECTURE	41
4.1	ANR System Architecture Overview	41
4.2	SSNs Architecture	43
4.2.1	SSN Structure	44
4.2.2	Event Characterization System	45
4.2.3	SSN Operation Pseudocode	48
4.3	MRS (MRSNs and G-MRSN) Architecture	49
4.3.1	MRS Structure	50
4.3.2	MRSN/G-MRSN Coordination Algorithm	54
4.3.3	MRSNs and G-MRSN Navigation Algorithm	55
4.4	ANR IoT Base Station Architecture	59
4.4.1	Base Station Gateway (BSG):	60
4.4.2	The BS GUIs	60
4.4.3	Cloud Data Storage, Visualizations and Analysis	65
Chapter 5	77
SIMULATION DEVELOPMENT ENVIRONMENT (SDE) WITH HIL	77
5.1	Design Structure of the SDE	77
5.2	WSNs-SMs Development	78
5.2.1	WSNs-SMs PCD-SE Part Libraries	78
5.2.2	WSNs Simulation: The Operation and Test	81
5.2.3	The Network Emulator	84
5.3	MRS-SMs Development	85
5.3.1	MRS-SMs and Navigation Environment	85
5.3.2	MRS Simulation: The Operation and Test	88
5.3.3	MRS Simulation: Case Examples	89
5.4	IoT BS: Simulation, Results Aggregation, Visualization and Cloud Integration	96
5.4.1	SND GUI Results Aggregation and Visualization	97
5.4.2	Cloud Platforms Results Aggregation, Visualization and Analytics	99
5.5	Hardware in the Loop (HIL) based Simulation	107
5.5.1	Role and Advantages of HIL based Simulation	108
5.5.2	HIL based Simulation: The Setup	108
5.5.3	HIL Based Simulation: The Results	111
Chapter 6	114
IoT BASED HYBRID ANR SYSTEM IMPLEMENTATION	114
6.1	WSN Hardware Components and Implementation	114
6.1.1	The Microcontroller (MCU) Board	114
6.1.2	The Sensors	116
6.1.3	The RF Communication Modules	116
6.1.4	The Sensor Nodes	117
6.2	MRSN Hardware Components	118

6.2.1	The Mobile Robot Platform	118
6.2.2	The Wi-Fi Router.....	119
6.2.3	The MRSN	119
6.3	IoT BS Hardware Components.....	120
6.3.1	The Wi-Fi Programming Module	120
6.3.2	The Host PC	121
6.3.3	Comparing the ANR IoT Gateway and Research/Commercial Gateways	122
6.3.4	IoT at BS and Node Level.....	123
6.4	Case Study Implementation with the Developed IoT based ANR	125
6.4.1	The Test Environment.....	125
6.4.2	Sensor Nodes Distribution and Operation.....	125
6.4.3	MRSN Operation	127
6.4.5	IoT BS Hardware Results Aggregation, Visualization and Cloud Integration	131
6.4.6	Triggering actions on the environment based on Cloud Analytics.....	139
6.5	ANR System Results Discussion	140
Chapter 7	141
CONCLUSIONS AND FUTURE WORK.....		141
7.1	Conclusions	141
7.2	Future Work.....	142
REFERENCES		143

LIST OF FIGURES

Fig. 1.1. Networked robotics paradigm [2].	1
Fig. 1.2. ANR structure for surveillance in large operational environments [22].	4
Fig. 2.1 Classification Path Planning Algorithms.	14
Fig. 2.2. Multirobot Task Allocation Taxonomy. Adapted from [67].	16
Fig. 2.3. WSN Protocol Stack.[27]	18
Fig. 2.4. Generic Sensor Node Architecture.	19
Fig. 2.5. Wireless Networking Categories	21
Fig 2.6. IEEE 802.15.4/ZigBee Protocol Stack [15].	22
Fig 2.7. IEEE 802.15.4/ZigBee Network Topologies [87].	24
Fig. 2.8. Wireless Serial (UART) Communication.	25
Fig. 3.1. System Scope and Hierarchy	34
Fig. 3.2. System scenario.	35
Fig. 3.3. System operational flow chart	35
Fig. 3.4. ANR System Development Methodology	40
Fig. 4.1. ANR system development block diagram	42
Fig. 4.2. ANR system architecture layout.	42
Fig. 4.3. The two-tier network system, forming an overall tree topology network.	43
Fig. 4.4. SSN architecture.	44
Fig. 4.5. Fuzzy set diagrams	47
Fig. 4.6. MRSN architecture.	51
Fig. 4.7. G-MRSN architecture.	53
Fig. 4.8. Contract Net Protocol (CNP).	55
Fig. 4.9. ANR system CNP	55
Fig. 4.10. A* Algorithm Example	58
Fig. 4.11. Simulation Occupancy Grid Map.	59
Fig. 4.12. BS architecture	60
Fig. 4.13. BS SND GUI information tab.	61
Fig. 4.14. MRSN1 tab interface	62
Fig. 4.15. SSN1(NN = 3) state machine operation.	63
Fig. 4.16. MRS GUI information tab	64

Fig. 4.17. MRSN1 tab interface.	65
Fig. 4.18. MS cloud storage platform	68
Fig. 4.19. Web application visualization interfaces	71
Fig. 4.20. ThingSpeak cloud platform	71
Fig. 4.21. ThingSpeak Channel Data Fields and Data Upload Process	72
Fig. 4.22. Neural network for storage conditions prediction.....	76
Fig. 5.1. Simulation Development Environment	78
Fig. 5.2. WSNs-SMs in PCD-SE.	81
Fig. 5.3. WSNs-SMs simulation operation process	81
Fig. 5.4. WSN-SM (SSN) during simulation.....	82
Fig. 5.5. SSN1 and G-MRSN O-SSN LCD.....	82
Fig. 5.6. WSNs-SMs virtual terminal displays.	84
Fig. 5.7. Network emulator interface	85
Fig. 5.8. MRSN1-SM: DaNI 1.0 CAD model.	86
Fig. 5.9. MRS-SMs navigation environment (NE)	87
Fig. 5.10. DaNI 1.0 Starter kit circuit block diagram	88
Fig. 5.11. MRS-SMs navigation simulation operation process	88
Fig. 5.12. MRSN1 SM navigation paths for case 1.	90
Fig. 5.13. LRES visualization of MRSN1 navigation: path 1.	90
Fig. 5.14. LRES visualization of MRSN1 navigation: path 2.	90
Fig. 5.15. MRSN1 coordination status for case 1	90
Fig. 5.16. MRS GUI control interface for alerting SSN notifications: case 2.....	91
Fig. 5.17. MRS-SMs navigation paths for case 2.	92
Fig. 5.18. LRES visualization of MRS-SMs navigation: Case 2.....	93
Fig. 5.19. MRS-SM coordination statuses for case 2.....	94
Fig. 5.20. MRS GUI control interface for alerting SSN notification: case 3	95
Fig. 5.21. MRS-SMs navigation paths for case 3.	95
Fig. 5.22. LRES visualization of MRS-SMs navigation: case 3	95
Fig. 5.23. MRS-SM coordination statuses for case 3.....	96
Fig. 5.24. BS simulation process	96
Fig. 5.25. BS SND GUI information tab during simulation.....	97

Fig. 5.26. MRSN1 SND GUI visualization	98
Fig. 5.27. SSN1 SND GUI visualization.....	98
Fig. 5.28. MS SSMS SQL database visualization and analytics	99
Fig. 5.29. Web application visualization and analytics.....	100
Fig. 5.30. ThingSpeak channel data visualization	100
Fig. 5.31. Zones temperature and RH regression analysis	102
Fig. 5.32. Zones %EMC and PI vs Dewpoint Analytics.....	103
Fig. 5.33. Neural network data inputs/targets for zone 1	104
Fig. 5.34. Zones' neural network regression analysis.....	106
Fig. 5.35. HIL based Simulation Setup	109
Fig. 5.36. Hardware sensor nodes during HIL based simulation.....	109
Fig. 5.37. MRS hardware during HIL based Simulation	110
Fig. 5.38. BS during HIL based Simulation	111
Fig. 5.39. SSN1 hardware module results	111
Fig. 5.40. SSN1 simulation model results	112
Fig. 5.41. MRSN1-SM Navigation	113
Fig. 5.42. LRES visualization of MRSN1 navigation during HIL based simulation	113
Fig. 5.43. MRSN1-SM coordination status during HIL based simulation.....	113
Fig. 6.1. Arduino Uno MCU board [163].....	115
Fig. 6.2. Arduino Mega MCU board [164]	115
Fig. 6.3. WSN Sensors	116
Fig. 6.4. RF communication modules.	117
Fig. 6.5. WSN hardware sensor nodes	118
Fig. 6.6. MRS hardware components.....	119
Fig. 6.7. Assembled MRSN and connection to BS.....	120
Fig. 6.8. NodeMCU Wi-Fi programming board and assembled BSG	121
Fig. 6.9. Host PC connected to BSG.....	121
Fig. 6.10. ANR structure for node based IoT integration	124
Fig. 6.11. The experimental setup lab environment and its grid map.....	125
Fig. 6.12. The actual layout of the implemented lab environment.	126
Fig. 6.13. Sensor nodes at the start of the experiment	127

Fig. 6.14. Sensor nodes used during experiment	127
Fig. 6.15. MRSN1 navigation assigned path within zone	128
Fig. 6.16. MRSN1 navigation of assigned path within zone1	128
Fig. 6.17. MRSN1 Coordination Status for Case 1.....	128
Fig. 6.18. MRS GUI control interface displaying SSN alert.....	129
Fig. 6.19. MRSNs navigation paths for case 2.	130
Fig. 6.20. MRSs navigation of paths to alerting SSNs within zones of lab environment.....	130
Fig. 6.21. MRSNs coordination status	130
Fig. 6.22. BS SND GUI information tab during hardware implementation	131
Fig. 6.23. MRSN O-SSN1 hardware results.....	132
Fig. 6.24. SSN1 hardware results.....	132
Fig. 6.25. MS SSMS SQL database visualization and analytics	133
Fig. 6.26. Web application visualization and analytics.....	134
Fig. 6.27. ThingSpeak channel hardware results visualization	134
Fig. 6.28. Zones temperature and RH regression analysis	136
Fig. 6.29. Zones %EMC and PI vs dewpoint analytics.....	136
Fig. 6.30. Neural network regression analysis per zone.....	137
Fig. 6.31. Downward action/command triggering communication	139

LIST OF TABLES

Table 2.1. Comparison of MRS architectures	9
Table 2.2. Review analysis of example MRS architectures	9
Table 2.3. Review Analysis of example MRS communication applications	11
Table 2.4. Review analysis of example MRS coordination and cooperation applications	12
Table 2.5. Review analysis of example MRS Path Planning applications.....	14
Table 2.6. Review analysis of examples of Market-based MRTA approach	16
Table 2.7. Review analysis of examples of Bio-inspired based MRTA approach	17
Table 2.8. WSN Communication Protocol Stack Descriptions [27]	18
Table 2.9. Sensor Node Architectures.....	20
Table 2.10. Comparison of Low-rate/Low-power Wireless Communication Standards.....	21
Table 2.11. Features/Comparison of ZigBee Topologies [87].	24
Table 2.12. Sub-1GHz Standards compared to 2.4GHz Standards	24
Table 2.13. Review analysis of example applications of imaging sensors surveillance systems..	26
Table 2.14. Review analysis of example applications of non-imaging sensors surveillance systems	26
Table 2.15. Review analysis of example ANR systems in surveillance	27
Table 4.1. Fuzzy sets for temperature, humidity and gas sensors	47
Table 4.2. Digital sensors' input combination with fuzzy rules output	48
Table 4.3. SSN coordinates	59
Table 4.4. Advantages, Challenges and Proposed Solutions of Cloud Data Storage	65
Table 4.5. Example sensor node data visualization on MS SSMS	69
Table 4.6. example sensor node statistics data visualization on MS SSMS	70
Table 4.7. Metal corrosion risk based on %EMC values	75
Table 4.8. Time ranges for organic material decay based on PI values.....	75
Table 5.1. MRS-SMs Navigation Coordinates: Case 2.....	91
Table 5.2. MRS-SMs navigation coordinates: case 3	94
Table 5.3. Statistical data analytics at each Zone	101
Table 5.4. Zone temperature/RH regression fit pattern line slopes	103
Table 5.5. %EMC and PI vs dewpoint best fit pattern line slopes per zone	104
Table 5.6. Neural network data per zone.....	105

Table 5.7. Zone neural network regression analysis R – values.....	106
Table 5.8. %EMC and PI predictions per zone	107
Table. 6.1. Required sensor features and selected sensors	116
Table. 6.2. RF communication modules connections and features	117
Table 6.2. Comparison of IoT Gateway Features (Research based Gateways)	122
Table 6.3. Comparison of IoT Gateway Features (Commercial Gateways)	122
Table 6.4. MRSNs and alerting SSN coordinates with distance between them: case 2	129
Table 6.5. Hardware results statistical data analytics	135
Table 6.6. Temperature/RH regression fit pattern line slopes per zone.....	136
Table 6.7. %EMC and PI vs Dewpoint best fit pattern line slopes.....	137
Table 6.8. Neural network regression analysis (R – Values) per zone	138
Table 6.9. %EMC and PI predictions per zone	138

NOMENCLATURE

Measures

$^{\circ}\text{C}$	Temperature
RH	Relative Humidity
% EMC	Percentage Equilibrium Moisture Content
CO Gas conc. (ppm)	Carbon-monoxide concentration in parts-per-million
Smoke conc. (ppm)	Smoke concentration in parts-per-million

Acronyms

ACO	Ant Colony Optimization
ANR	Autonomous Networked Robots
BS	Base Station
BSG	Base Station Gateway
CAD	Computer-aided design
CNP	Contract Net Protocol
GA	Genetic Algorithm
G-MRSN	Global-Mobile Robot Sensor Nodes
GUIs	Graphic User Interfaces
HIL	Hardware-in-the-loop
IDE	Integrated Development Environment
IPSO	Improved Particle Swarm Optimization
LRES	LabVIEW® Robotics Environment Simulator
MAC	Medium Access Control
MCU	Microcontroller
MRPP	MRS Path Planning
MRS	Multi-robot Systems
MRSNs	Mobile Robot Sensor Nodes
MRS-SMs	MRS simulation models
MRTA	Multirobot Task Allocation
NE	Navigation environment

NI	National Instruments
O-SSNs	Onboard-Static Sensor Nodes
PC	Personal Computer
PCD	Printed Circuit Board
PCD-SE	Proteus® Circuit Design Simulation Environment
PRM	Probabilistic Roadmap Method
SDE	Simulation Development Environment
SE	Simulation Environment
SNG GUI	Sensor Node Data Graphic User Interface
SOSUS	Sound Surveillance System
SSNs	Static Sensor Nodes
WINS	Wireless Integrated Network Sensors
WSNs	Wireless Sensor Networks
WSNs-SMs	WSNs simulation models

Chapter 1

INTRODUCTION

Wireless [1] or Networked [2] robotics is an evolving research field that integrates research from the communities of robotics, sensor networking and artificial intelligence (AI) [2]. It features in many applications such as in healthcare [3], mining [4] to military [5]. The detailed definition of networked robotic systems is provided in [6] and it covers two categories:

- **Teleoperated:** whereby the robots are remotely controlled over the network by human supervisors.
- **Autonomous:** whereby the robots interact with sensor networks exchanging data, and the sensitivity and perception range of the robots are extended by the sensor networks.

The paradigm of networked robotics is presented in Fig. 1.1 [2]. This paradigm lists the research reflected in the field and integrated to facilitate networked robotics systems. Integrating MRS and WSNs increases the efficiency of autonomous networked robot systems by ensuring wider environment coverage in the detection and surveillance of various phenomena. Hence, many studies have been devoted to challenges related to its system architecture, mobile robots' sensing and control (localization, path planning and navigation), cooperation (coordination, task allocation), communication between the robots and the sensor networks, etc. [1]. Other studies dedicated to WSNs have focused on sensor node architectures, network communication protocols, networking routing and topology, etc. [7].

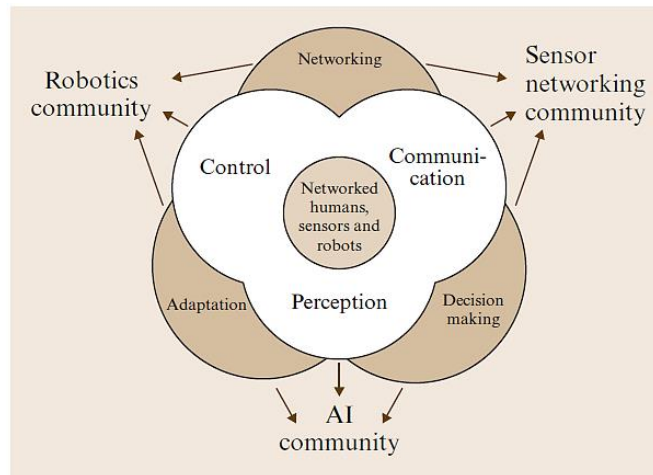


Fig. 1.1. Networked robotics paradigm [2].

1.1 Study Background

This work focuses on the application of Autonomous Networked Robotics (ANR) systems featuring MRS and WSNs in indoor environment surveillance. The MRS system in ANR features mobility and interaction capabilities, that is supported by the distribution of WSN nodes in which the general system sensitivity and perception is increased. Hence, an ANR system should be understood in the light of its two subsystems and how they interact.

1.1.1 Multirobot Systems (MRS)

MRS is a system of multiple robots working together to achieve common goals by executing joint tasks. The research and development of multirobot systems (MRS) started in the 1980s [2], [8], but has since experienced tremendous growth and expansion. Presently, multiple robot systems fall into two major groups:

- i. **Collective Swarm Systems:** These systems mostly consist of a large number of homogenous mobile robots [9], [10]. The robots usually make use of their local control abilities to generate global coherent team behaviors with less dependence and communication with their teammates.
- ii. **Intentionally Cooperative Systems:** These systems consist of robots that have infrastructures that enable intentional communication between members of the team. Hence, they are aware of their teammates within the environment. [8]–[10]. These systems can be strongly or weakly cooperative depending on the applications.

According to [9], [10]–[14], most of the research in multirobot systems are focused on the following topics:

- A. **Architectures:** this area of study deals with how the robots' connections, organization, and interaction. Multirobot architecture is fundamental to all multirobot systems and determines the robustness and scalability of the systems.
- B. **Communication (com.):** it deals with how a robot exchange information both within itself (intra-robot) and between other robots (inter-robot). Most research in MRS communication is inter-robot based and is divided into implicit (com. through team member action on environment), passive (com. through sensors) and explicit (com. through protocols) communications.
- C. **Homogenous/Heterogeneous MRS:** this involves the nature of the robots making the MRS. If all the robots have common characteristics, then the MRS is said to homogenous,

and key study areas in this type of system include swarm and modular robotic systems. Heterogeneous MRS contains robots with different characteristics, usually designed for more complex tasks.

- D. **Task Allocation:** this is a very wide area of MRS research and development known in the robotics community as Multirobot Task Allocation (MRTA). It seeks to provide a solution to the problem of task distribution and execution among the robot team members.
- E. **Learning:** learning in MRS involves adapting to changes in the environment, acquiring new cooperative behaviors, etc. Also, there may be parallel learning whereby other robots in the environment have their own learning tasks and goals to achieve.

1.1.2 Wireless Sensor Networks (WSNs)

MRS has many benefits that make them preferable to single sophisticated robots. But their capabilities such perception, coordination, navigation, etc., can be increased when they are integrated with WSN. In [7], WSNs are defined as networks of sensor devices deployed to monitor processes or environmental conditions, whereby data communication between components of the system is done through wireless standards. In WSNs, several fields of study such as sensing, computing, networking and actuating are integrated.

A typical WSN(s) consists of sensor nodes (static and/or mobile), gateways/sinks, base stations and wireless communication standards that facilitate interactions between the nodes [15]. A sensor node may contain one or more sensors; a conditioning circuit (for example microcontroller board, signal processor, etc.) for data processing and a communication unit. Gateway/sinks are the network coordinators; they are usually set up to synchronize the network. Base stations are for data processing, interpretation, storage and control of the network. From the base stations, further connections may be made to other places through the internet or higher-level network control and synchronizations can be achieved through the internet of things (IoT) connections. Connectivity between the WSN components are established and maintained by several standards such as IEEE 802.15.4/ZigBee, IEEE 802.11, WirelessHART, etc.

The origins of WSNs can be traced to the Sound Surveillance System (SOSUS) by the US military in the 1950s [16]. Building on this research, the US Defense Advanced Research Projects Agency (DARPA) started the Distributed Sensor Network (DSN) program in the 1980s. This program served as a springboard for multiple university-based WSN research such as the UCLA Wireless Integrated Network Sensors (WINS) (1993) and University of California at Berkeley

PicoRadio program (1999) [17], [18], etc. Today WSNs comprise networks of diverse protocols, sensors, devices, and applications.

1.1.3 Autonomous Networked Robotics (ANR)

ANR systems is the cooperation of robots and sensor network systems for the coordination and execution of tasks within a given environment. Therefore, fundamental to these systems is the interaction between the robots and their environment through the WSNs. This study focuses on the evolution and development in the field of ANR and presents a design for surveillance in large operational environments such as factory floors or warehouses, as illustrated in Fig 1.2. Other examples of ANR applications can be found in the military for target surveillance and tracking [5], mining for search and rescue operations [4], smart homes for various home-related tasks such as object recognition, pick and place, etc. [19], healthcare for health monitoring [3], and in recent years, applications have extended to cloud robotics [20], [21] etc.

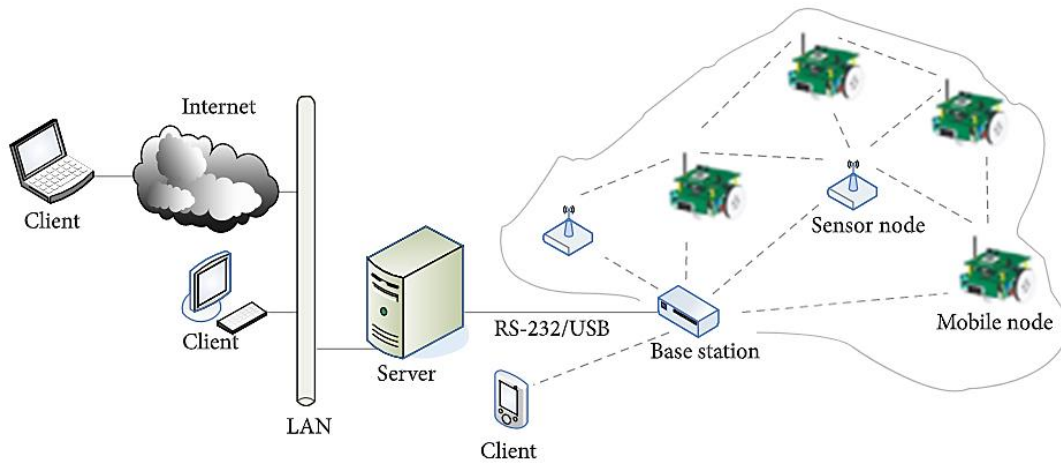


Fig. 1.2. ANR structure for surveillance in large operational environments [22].

1.2 Statement of Purpose

Environment surveillance is always a requirement for organizations that operate warehousing systems, supply chain logistics, etc. The current expansion and automation [23] of these systems require innovative measures of surveillance to cover the wide and dispersed operational environments. With the current use of mobile robotic systems in these large environments [24], [25], an extension to ANR offers benefits such as increased scalability allowing more expansion, robust environment surveillance in addition to other normal operations, etc.

Despite these benefits, ANR systems have many technical challenges. Some of them include the effective deployment of MRS and WSNs, coverage and localization, system coordination

(between MRS robot team members, and between MRS and WSNs) and application related challenges [26]. Given these challenges, the purpose of this study is to develop, simulate, implement and evaluate an ANR system for indoor intruder and fire surveillance using non-imaging sensing devices.

1.3 Objectives and Expected Outcomes

The objectives of the research are as follows:

1. To design an ANR system:
 - i. The development of an ANR simulation environment supporting hardware-in-the-loop (HIL) testing and validation.
 - ii. Develop and implement an ANR system comprising of static sensor nodes and dynamic sensor nodes on mobile robots.
 - iii. Develop and implement a hybrid architecture for the coordination of the ANR system, with the MRS as the task executor and secondary task distributor and static sensor nodes as the primary task distributor.
 - iv. Application of map-based navigation algorithm (A* path planning) for the MRS control.
2. Use the developed ANR system for demonstration in an indoor surveillance application for effective fire and intruder detection using non-imaging sensors, whereby:
 - i. The non-imaging sensors are distributed in the environment and attached to the robot to sense and analyze events in the environment using logic-based systems.
 - ii. The analyzed events and sensor data are communicated to the base station where they are visualized and uploaded to cloud platforms for remote monitoring and analytics.

The deliverables of the work include the development of:

1. An ANR surveillance application using non-imaging sensors with the Internet of Things (IoT) and cloud data visualization and analysis integration.
2. A simulation environment supporting HIL testing and validation of the ANR system.
3. An implementation of the market-based Contract Net Protocol (CNP) for ANR MRS system coordination.

1.4 Expected Contributions

The development of the ANR system for indoor environment surveillance contributes the following to existing research and development of networked robotics and its applications:

1. An ANR system for surveillance featuring environment zoning, IoT and cloud data storage and analysis
2. A hybrid architecture for market-based ANR MRS coordination approach.
3. A simulation development environment supporting HIL integration to facilitate ANR system development, testing, validation, and effective deployment.

1.5 Thesis Structure

The rest of the thesis report is structured as follows; chapter two presents the literature review of the various systems involved in ANR, while chapter three discusses the requirements and research methodology. Chapter four presents the system architecture while chapter five presents the system simulation environment, testing, and results. In chapter six, the system implementation is reported, while chapter seven concludes on the research with recommendations and future work.

Chapter 2

LITERATURE REVIEW

The key features of the ANR system involve group-level interactions between the robots and their environments through the deployed WSNs. These features are reflected in the key properties of MRS, including architectures, communication, coordination and cooperation, path planning and task allocation [2]. WSNs, on the other hand, are featured by characteristics such as physical and medium access control (MAC) properties, networking and routing topologies, etc. [27].

MRS architectures refer to the building framework of the whole system. The MRS control architecture is one of the major determinants of the overall system efficiency, reliability, scalability and robustness. They form the basis of the operation of any MRS. After an MRS architecture has been designed, the next supporting structure is the communication system between members of the MRS. MRS communication provides the means through which higher forms of interactions (such as planning) within the MRS are enabled [28]. When communication has been established within the MRS, then members of the system can start to coordinate and/or cooperate for different tasks. MRS coordination can be for task cooperation, whereby the robots work together, or task competition whereby tasks are executed based on factors like nature of robots, time constraints or nature of the tasks [29]. With the MRS coordination structure defined, then navigation and tasks can be planned and allocated, leading to the MRS characteristics of planning and task allocation. This section presents case studies of each of these MRS characteristics.

On the other hand, WSNs are characterized by sensor nodes, gateways and other components that are designed based on the WSN protocol stack. The generic WSNs protocol stack is divided into communication protocols (Physical, MAC, Network, etc. layers) and the management protocols (power, mobility and task management) [27]. The generic WSN protocol influences the design architectures of sensor nodes, and the functions of gateways and base stations. This study specifically explores WSNs based on low-rate, low-power communication standards. The IEEE 802.15.4/ZigBee and sub-1GHz standards are presented along with several examples of surveillance systems based on these communication standards.

Finally, this section presents example applications integrating both MRS and WSNs technologies, highlighting their achievements and limitations. Then based on all the review studies, the research challenges and open areas are discussed.

2.1 Related Work in MRS

In [10], coordination and control, communication, mapping and localization, and architectures were presented as the key areas of research in MRS. Hence, in this subsection, related works will be discussed under these areas with the addition of the multirobot planning and task allocation problems.

2.1.1 MRS Architectures

MRS architecture is the framework upon which the group's collective behaviors are implemented, and thereby it determines the robustness, scalability, capabilities, and limitations of the system. Four essential architectures have been adopted for MRS, the include central, hierarchical, decentralized and hybrids. These four architectures have been explored and implemented by various researchers in diverse ways, and brief discussions of each architecture is hereby presented. Meanwhile Table 2.1 presents the comparative summary of the various architectures.

- A. **Centralized Architectures:** this architecture coordinates all the control of the team from a single point. It is limited in that the whole system can fail when the central control point fails. Also, real-time control is sometimes unachievable due to the communication cost of sending signals back to the central point from the team members. Hence, system suffers from accurate scalability and can only be employed when there is a direct line of sight from the central point to every team member. Therefore, they are mostly theoretical.
- B. **Hierarchical Architectures:** these architectures are limited to specific applications, especially those involving 'leaders/followers' frameworks. A robot (that acts as the leader) surveys the actions of a small group of other robots (the followers). This formation is repeated throughout the system hierarchically down to the last robot that simply executes its assigned task. They have better scalability than centralized architectures, but they suffer failures when there are system breakdowns high in the control tree.
- C. **Decentralized Architectures:** unlike the first two architectures, decentralized architectures have a wide variety of applications in real-time environments and are the commonest in multirobot systems. Each robot in the team acts based on its local knowledge of the environment, making the whole system robust to failure since the robots are less

dependent on each. But limitations exist where global tasks and goals for the whole team are to be incorporated into the local control of each robot member.

- D. **Hybrid Architectures:** these architectures combine both local and higher-level control approaches to attain robustness, higher scalability. Hence, they easily execute tasks both locally and globally.

Table 2.1. Comparison of MRS architectures

Architecture Type	Performance Metrics			
	Reliability	Robustness	Scalability	Efficiency
Central	✓	✓	✓	✓ ✓ ✓
Hierarchical	✓	✓ ✓	✓ ✓	✓ ✓
Decentralized	✓ ✓	✓ ✓	✓ ✓	✓ ✓
Hybrid	✓ ✓ ✓	✓ ✓ ✓	✓ ✓ ✓	✓ ✓

Some of the earliest implementations of multirobot architectures included ACTRESS [30] and ALLIANCE [31], while other examples are presented in Table 2.2.

Table 2.2. Review analysis of example MRS architectures

Ref.	Architecture Type	Communication System	Test Method	Results (Achievements)	Challenges/Future Work
[32]	Hybrid (Hierarchical and Behavioral algorithms)	Distributed among robots	Simulations on Microsoft VC++ 6.0 development IDE.	Efficient team formation due to behavioral-based architecture.	No implementation.
[33]	Hierarchical/Distributed	Distributed among robots	Simulations for Hunter-Prey problem and Robotic Soccer	Successful formation of robotic micro-societies for dynamic role assumption	Real-time implementation on Robotic Soccer
[34]	Hybrid	Distributed	Experiments using Two Pioneer AT TM Robots	Successful cooperation among robots for transportation of box along the desired path	Optimization algorithms to define robot positions
[35]	Hierarchical/Distributed	Distributed	Simulations on MATLAB	Fast redeployment of tasks among robot teammates with the ability to execute closely interacting tasks	Real-time implementations and experiments

[36]	Distributed	Distributed	Simulations on ROS and experiments using actual robots	Successful validation of the decentralized architecture with multiple layer leading to robot collaboration	Extension of component interfaces to extend robot member uses.
[37]	Hybrid, based on Service Oriented Architecture (SOA)	Distributed	Simulations	Successful development of a generic environment supporting MRS protocols such as task execution and failure recovery	Incorporating humans into the developed architecture

2.1.2 MRS Communication

Communication in MRS plays a very important role, especially the fact that it provides the basis for high-level interaction among the robots [38]. Through communication robots in an MRS can share information on their locations, perceptions, thereby distributing their sensing capabilities over more extensive areas of coverage. There are several means of communication in MRS such as bio-inspired as presented in [38], [39], and wireless communication standards (ad-hoc networks). Generally, MRS communication is categorized into implicit (stigmergy), passive and explicit communication, as briefly discussed below.

- A. **Implicit Communication (Stigmergy):** this is robot communication with its teammates through their actions and effects on the environment. It is advantageous because of its simplicity since no communication standards are required. But disadvantageous to the extent that a robot can rightly perceive its environment to know what stage of the task the team is currently executing.
- B. **Passive Communication:** here, robots use sensors to observe the actions of their teammates. It is free of problems associated with protocolled communications such as bandwidth, faults, attacks and fallibility of the communication mechanism. But it is also plagued with similar defects as in implicit communication such as rightly perceiving a teammate's actions since sensor data can be noisy.
- C. **Explicit (Intentional) Communication:** this is the most standard type of communication in MRS. Here a robot intentionally communicates a specific message to its team members, usually through a standardized communication protocol, such as radio communication

standards. Such communications are direct and simplifies synchronization among MRS. Hence, robots know with relative ease the state of their teammates. Explicit communications are mostly beneficial for MRS with few numbers of robots, and applications that require fast responses such as disaster management [10]. Yet, explicit communication methods are limited by fault tolerance and reliability of the mechanisms and protocols since the communication channels can be noisy and have limited bandwidths and may not continually connect all members of the team.

Wireless communication are the commonest means of communication among MRS since they are explicit; hence, this study focuses on this communication type, particularly the low power standards such IEEE 802.15.4/ZigBee, Bluetooth, IEEE 802.11, etc. Example, MRS communication applications, are presented in Table 2.3.

Table 2.3. Review Analysis of example MRS communication applications

Ref.	Communication System	Test Method	Results (Achievements)	Challenges/ Future Work
[38]	Biological (Centralized and Distributed) using Pheromones	Implementations using E-puck robots	Successful regulation, allocation and execution of tasks for a group of 8-16 robots for both systems	Implementation of communication systems for more dynamic tasks
[40]	Distributed IR and RF (IEEE 802.15.4/ZigBee for RF) Hybrid Wireless Network	Simulations on MATLAB	Successful synchronization and localization of a swarm of 50 robots	Errors due to the use of RSSI measurements
[41]	Peer-to-peer RDP protocol (IEEE 802.11)	Implementations on Stargate MUCs and High Processing core laptops	Unicast messaging best suited for small size MRS while broadcasting for large size MRS	Dynamic implementation of the algorithm based on the size of MRS
[42]	Distributed Network routing using RF communication (IEEE 802.15.4/ZigBee for RF)	Scarabs using ZigBee Technology on XBee transceiver modules	Efficient trajectory planning and following by the robots even in dynamic and unexpected events	Time spent in task execution and robustness to complex tasks and unknown environments.
[43]	Distributed Network based on (IEEE 802.15.4/ZigBee for RF)	Lab design robots based on Arduino MCU and XBees	Successful validation of the (IEEE 802.15.4/ZigBee for RF) protocol for localization and communication in large MRS	Better handling of localization errors
[44]	Distributed network based on IEEE 802.11 (Wi-Fi)	Simulations using ROS and OMNET	Successful validation of the Wi-Fi communication protocol for MRS in	

			warehouse storage facility with various types of interference, blockage and robots' mobility	
[45]	Distributed network	Simulations	Successful communication among robots for sharing and updating of environment maps for navigation	Extension of experiments to time varying environments
[46]	Distributed communication based on reasoning	Simulations	Reduction in channel utilization for MRS communication	The use of different communication models such as point-to-point for robots' communication.

2.1.3 MRS Coordination and Cooperation

In MRS, robots can either cooperate or compete for available tasks. In cooperation, all robots work together to achieve a common task or multiple tasks while under competitive architectures, robots act individually to perform the available tasks [11]. As noted in [11], MRS coordination is classified into:

- A. **Static Coordination:** where the coordination rules and instructions are adopted before the robots engage the tasks. This is also referred to as deliberative or offline coordination.
- B. **Dynamic Coordination:** here, coordination is online and reactive i.e. it occurs during the execution of the task. It is largely dependent on the analysis and synthesis of shared information between the robots, obtained through communication means.

Typical examples of MRS coordination and cooperation includes multirobot mapping/coverage [47] [48], localization [49], [50], monitoring, exploration and tracking [51]. Table 2.4 presents the analysis of some researches with the above cooperation examples.

Table 2.4. Review analysis of example MRS coordination and cooperation applications

Ref.	Algorithm (s)	Application Type	Test Method	Results (Achievements)	Challenges/Future Work
[47]	Backtracking Spiral – Cooperative Multiagent (BSA-CM) on homogenous robots	Coverage and Exploration	Simulation over JAVA based multiagent framework BESA	Successful deployment of robot team to cover all exploration, with tasks conflict resolutions and deadlock situations.	Implementation of real robots

[48]	Loosely Couple and Serialized Operation Cooperation on heterogeneous robots	Mapping	Simulations on Matlab ® and Webots ®	Efficient mapping with limited sensing robots.	No real implementation and testing on larger MRS
[49]	Probabilistic Graphical Models	Localization	Experimentations on three vehicles on urban roads.	Minimized pose estimation uncertainties and scalable localization.	
[50]	Approximate Decentralized Algorithm	Localization	Simulations on Matlab ®	Scalable localization using robot group parts as landmarks	Incorporation of orientation noise and localization interdependence.
[51]	Decentralize Partially Observable Markov Decision Processes (POMDPs)	Exploration and Tracking	Simulations on Matlab ®	Reduced computational complexity in MR cooperation	Sub optimal results
[52]	Risk based coordination algorithm	Task allocation	Simulations	Improved team planning and task allocation used risk-based coordination	Making the coordination approach adaptive to handle sudden changes in the environment
[53]	Coordination based on Petri nets and temporal logic specifications	Exploration	Simulations	Successful coordination of robot teams with the creation of team supervisors	Real-time implementation.
[54]	Distributed coordination	Task allocation for object tracking	Simulations	Effective tracking of targets based on a competitive mechanism	Implement competitive control at robot dynamic levels

2.1.4 MRS Path Planning (MRPP) and Navigation

Path planning is the sequence of motions (translation and rotation) a robot undertakes to move from its starting position to its destination while avoiding obstacles in its environment [55]. In MRS research, there is usually the MRPP problem definition for the given scenario. The generic definition is given as:

“Given m robots in k -dimensional workspace, each with start and goal pose, determine the path each robot should take to reach its goal, while avoiding collisions with other robots and obstacles [55].”

MRPP can be **global** when the robots generate the path on a known environment or based on current and past perception of the environment; or **local** when the robots’ path planning occurs in

real-time using its onboard sensors while navigating the environment [56]. Global path planning can sometimes involve paths that are generated by a central planner and communicated to various robot team members [57]. There are two groups of path planning algorithms: Classical and Heuristic-based Algorithms as shown in Fig. 2.1. Current research and applications in planning have mostly focused on heuristics methods [55]; hence, the review analysis of example heuristic-based studies is presented in Table 2.5.

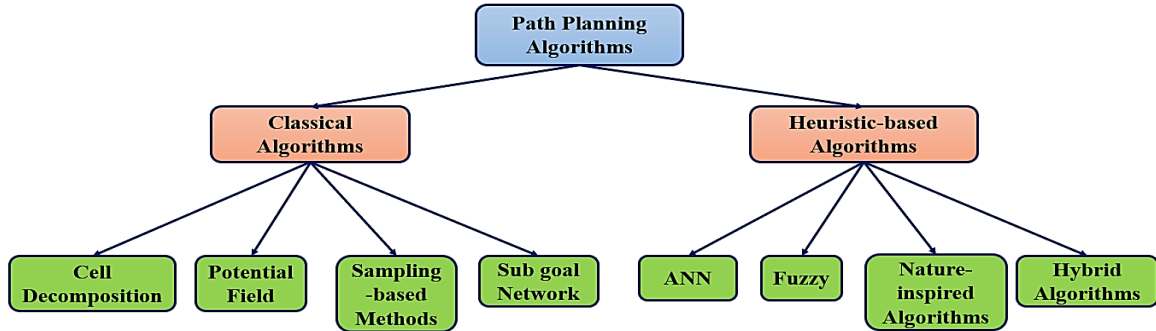


Fig. 2.1 Classification Path Planning Algorithms.

Table 2.5. Review analysis of example MRS Path Planning applications

Ref.	Algorithm (s)	Obstacle Type	Test Method	Results (Achievements)	Challenges/Future Work
[58]	Improved Particle Swarm Optimization (IPSO) and Probabilistic Roadmap Method (PRM)	Static	Simulations	For 600 problem instances, 17% and 34% faster speed than standard PSO+PRM was achieved	No real implementation
[59]	Probabilistic Neuro-Fuzzy	Static and moving	Simulations and Implementations on AutoMerlin	Improved robustness and performance on the navigation system.	Implementation on a real and complex system
[60]	Improved Particle Swarm Optimization (IPSO) and Differentially Perturbed Velocity (DV)	Static	Simulations and experimentations using Khepera II Networked Robotics	Better performance than conventional IPSO and Differential Evolution in path planning and navigation	The environment was static, future implementation on dynamic environments
[61]	Genetic Algorithm and Adaptive Fuzzy-Logic Control	Static	Experimental Implementations on RobuTER	Successful perception, environment modeling, finding the best collision-free path, online position and orientation determination and	Deployment of robots to larger, complex environments with dynamic obstacles

				execution of path generated	
[62]	Multi-Agent Distributed and Local Asynchronous (MADLA) Planner	N/A	Computational Simulations and implementations	Successful development of the algorithm as a novel multi-heuristic multi-agent search scheme	Yet to be implemented for optimal multi-agent planning.
[63]	DARP (Divide Areas based on Robots Initial Positions) algorithm and ACO	Static	Simulations	Achievement of complete coverage with minimum length, non-drawbacks and least number of turns	Further reducing the turning around obstacles
[64]	Blockchain based path planning using PRM	Static	Simulations	Reduction of path planning latencies and delays	
[65]	Improved Push and Spin (PASp) algorithm	Static	Simulation	Reduction of the number of moves, hence, path is planned is lesser time	

2.1.5 MRS Task Allocation (MRTA)

This is a very expansive area of MRS research and development known in the robotics community as Multirobot Task Allocation (MRTA). There is a definite structure that categorizes all the possible ways in which tasks could be allocated in an MRS. This structure is called the MRTA Taxonomy. The basic concept of task allocation is that robots in any given MRS must execute tasks either individually, collectively or otherwise. Tasks can be subdivided into subtasks, hierarchical task trees, etc. Independent subtasks may be executed concurrently, while task trees are executed according to their interdependence. Researches usually mention the *task allocation problem* which usually presents the challenges that the task allocation algorithm seeks to solve with minimal costs. According to works of literature cited in [2] and [66], the MRTA taxonomy is broken down as given in Figure 2.2. In the figure, level 1 of the taxonomy is called MRTA Schemes [67], where the planning of the task allocation is made depending on the nature of the tasks and the robots available. Level 2 are the existing paradigms for implementing level 1; there is a centralized paradigm where all tasks are allocated from a central point and the decentralized paradigm where tasks are distributed among the robots. Finally, level 3 determines what approach will be used to execute the task. In behavioral approaches tasks are differentiated and assigned to groups called behavioral groups. Market-based approaches are mainly based on the concept of auctions and bidding, hence, they seek to maximize “*income*” (information and speed), while minimizing “*costs*” (convergence and communication time). The auctioneers can be central

controlling system in centralized paradigms, or other robots, systems (such as WSN) in decentralized paradigms. The robot team members are always bidders. Bio-inspired approaches seek to mimic biological systems of task execution as existent in ant colonies, beehives, and other social insects.

Market-based and bio-inspired MRTA approaches are the commonest in literature, hence Table 2.6 and Table 2.7 presents applicational examples of researches in market-based and bio-inspired approaches respectively.

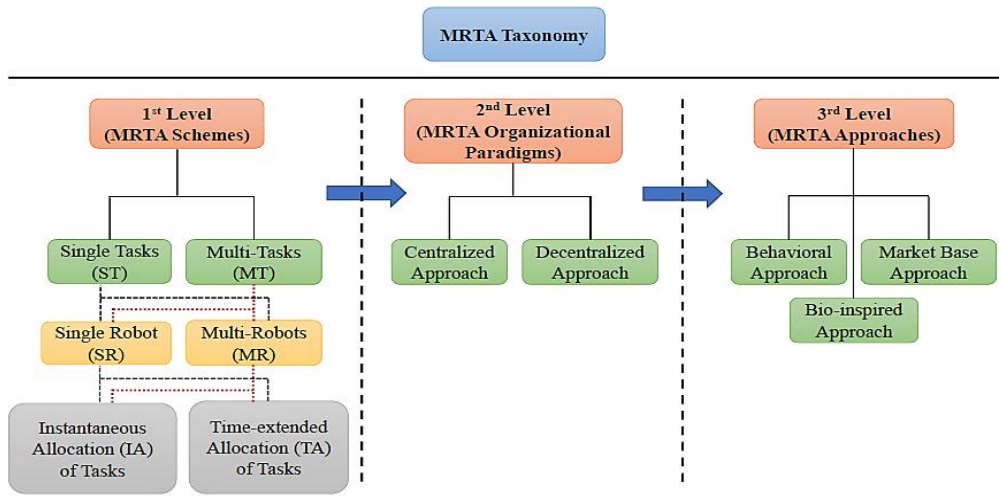


Fig. 2.2. Multirobot Task Allocation Taxonomy. Adapted from [67].

Table 2.6. Review analysis of examples of Market-based MRTA approach

Ref.	Algorithm	Heuristics	Test Method	Results (Achievements)	Challenges/Future Work
[68]	Resource-Oriented, Decentralized Auction Algorithm (RODAA).	Robot resources, bids based on task completion time	Simulations on Webots Simulator	Successful resource management during task allocation and execution	No real-time implementation
[69]	Round-robin (RR), Ordered single-item auction (OSI), Sequential single-item auction (SSI), Parallel single-item auction (PSI).	Auctions and bids	Computer Simulations to validate the four auction algorithms against standard performance metrics	Sequential single-item auction (SSI) was found to be the best trade-off algorithm	No Real-time validation yet.

[70]	Consensus-Based Parallel Auction and Execution (CBPAE)	Highest robot bidders and consensus among team members	Simulations and experiments with Pioneer 3Dx robots	Improved in task execution time and fast response to emergency tasks	Incorporation of learning algorithms and extension to multi-tasks
[71]	Standard Sequential-Single-Item (SSI) auction algorithm	Bids based on the dynamicity of tasks	Simulations on Blocks World for Teams (BW4T) environment	Effective team coordination in stepwise task completions	Real-time experiments and faster task completion time
[72]	Theory of Comparative Advantage	Benefit of specialization	Simulations and real-time implementation with a team of 3 heterogenous robots	Reduction in task execution time, thereby improving robot team behavior in unknown environments	Implement the algorithm for distributed robot systems.

Table 2.7. Review analysis of examples of Bio-inspired based MRTA approach

Ref.	Algorithm	Heuristics	Test Method	Results (Achievements)	Challenges/Future Work
[73]	Particle Swarm Optimization	Particle position and velocity	Experiments using ELISA III swarm robots	Robustness in execution time, communication and task allocation convergence	
[74]	Ant-like Behavioral Algorithm	Pheromones sensing for food collection	Computer Simulations	Efficient task identification and task switch among robot team members	No Real-time implementation
[75]	Improved Ant Colony Optimization (ACO)	Pheromone concentrations	MATLAB Simulations	Increased task allocation benefits	No Real-time experimentations
[76]	Genetic Algorithms (GA) combined with Local Search schemes	Population sizes, local search schemes permutations	Computational experiments on a storage tank facility with the robots as cleaning agents.	Fast convergence of the algorithm for the distribution of tasks in the facility.	Improving the global search stage by the GA
[77]	Neural Network and GA	Population size	Simulations	Improved neural network training time leading to reduced task execution time by robot team	Real-time implementation

For an ANR system, an MRS with the above characteristics clearly defined is integrated with a WSN. The next section of this chapter discusses the key features of WSNs, with further elaboration of the specific communication standard used in this research, as well as examples cases of the application of WSNs for surveillance.

2.2 Related Works in WSNs

Generally, every WSN can be studied and evaluated based on the nature of its protocol stack. The basic protocol stack of any WSN is shown in Figure 2.3. Researches in WSN have focused on improving the efficiency of the whole network by concentrating on particular layers of the stack. Table 2.8 gives the various functions of each layer in the stack. In relation to robotic applications, the major areas of research in WSNs have been the Link Layer (with the Medium Access Control-MAC) for improving network efficiency, energy management, delays, throughput, robustness and stability. Also, the Networked Layer is the next in research concentration since it manages the network scalability, routing, and topology protocols. These two layers contribute towards a major application of WSN in robotics, namely control, navigation, localization and tracking, coverage and exploration and data collection [78].

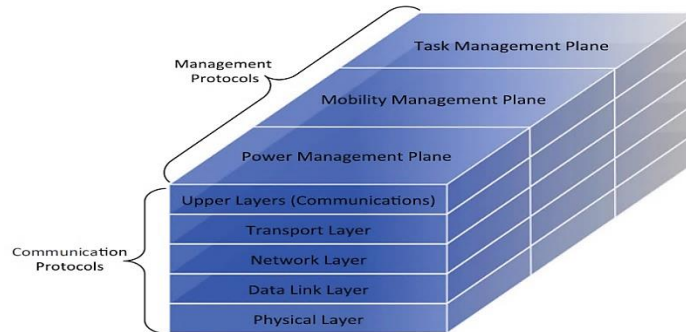


Fig. 2.3. WSN Protocol Stack.[27]

Table 2.8. WSN Communication Protocol Stack Descriptions [27]

Upper Layers	In-network applications, including application processing, data aggregation, external querying, query processing, and external database
Layer 4	Transport, including data dissemination and accumulation, caching, and storage
Layer 3	Network layer: routing and topology management
Layer 2	Link layer: channel sharing (MAC), timing, and locality
Layer 1	Physical medium: communication channel, sensing, actuation, and signal Processing

2.2.1 Major Features of WSNs

WSNs have 3 major features, the sensor nodes (which can be end nodes, routers or coordinators), gateways or base stations and the communication standard linking the nodes to the gateway or base station and other wider networks.

A. Sensor Nodes

Sensor nodes are the key building blocks of WSNs. Their primary purpose is to sense, process and report. They usually consist of a single or a variety of sensors depending on the application of the WSN. Sometimes, the sensors are built on a separate module called a sensor module. Depending on the application, there may be actuators that respond based on the sensed information either from the sensors themselves or received information from the network. There are two generic architectures for sensor nodes, the MCU-based architecture and the other processor type architectures [79]. Both architectures feature sensors (analog or digital), memory unit, communication unit, power unit and the processing unit, which can be MCUs or reconfigurable devices such as FPGAs, as shown in Fig. 2.4. Table 2.9 presents example sensor nodes architectures, their main processors and example implementation.

Basically, most WSNs usually have two types of sensor nodes: Full Function Devices (FFDs) and Reduced Function Devices (RFDs)

- a. Full Function Devices (FFDs): these are sensor nodes that can both sense and act as routers and/or coordinators for other nodes. Hence, they are equipped with additional memories and computing power.
- b. Reduced Function Devices (RFDs): these sensor nodes are deployed only for sensing only. Hence, they have limited functionalities, and are equipped with just enough memory and computing power to make them perform their functions efficiently.

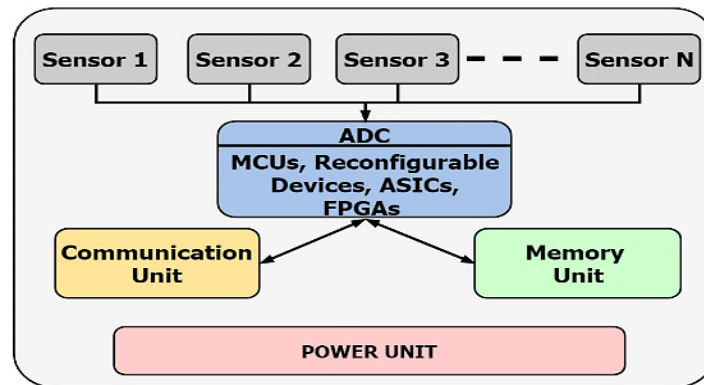


Fig. 2.4. Generic Sensor Node Architecture

Table 2.9. Sensor Node Architectures

Sensor Node Architecture	Processing/Programmable Devices	Examples
Microcontroller Based	Microcontrollers	DiGi XBee [26], MICA 2
Digital Signal Processor (DSP) Based	Digital Signal Processor chips	ARM-DSP nodes
Application Specific Integrated Circuit (ASIC) Based	MCUs and/or ASICs, Field Programmable Gate Arrays (FPGAs), etc.	
Programmable Hardware Devices based	FPGAs, Complex Programmable Logic Devices (CPLDs)	Cyclops [80].
Field Programmable Analog Array (FPAA).	FPAA, DSPIC (digital signal controller)	
System-on-Chip (SoC)/System in Package (SiP) based	DSPs, Intellectual Property (IPs) processors, ASICs, etc	3D-SiP [27]
Subsystems on Programmable Chips (SoPCs) based	FPGAs, hard/soft processor cores	Sensor nodes on ALTERA FPGA [27]
Multi-Processor System-on-Chip (MPSoC)/Massively Parallel Processing System-on-Chip (MPPSoC) based	Several processing devices as the application may need	
Institute of Electrical and Electronics Engineers (IEEE) 1451 Standard-based	Network Capable Application Processors (NCAPs) and Transducer Interface Modules (TIMs).	
Mixed Architecture based	FPGAs, CPLDs, ASICs, MCUs, etc.	PowWow [81]

B. Gateways and Base Stations

Gateways can be sensor nodes called Network Coordinators, capable of sensing and routing; or simply Sinks to which all information from the network is sent to. They are also responsible for starting the network either through broadcast messages or multicast messages. From the gateway, all information is sent to the base stations [15].

Base stations, on the other hand, are the network controllers. They have oversight of the network, processing, and interpreting every data received. One important function of the base stations is to determine how nodes are added or removed from the network. They also perform network maintenances such as overall security checks, and synchronizations. Finally, they provide connections to further remote systems such as web internets and internet of things (IoT) applications.

C. WSN Wireless Communication Standards

WSN standards are a broad set of protocols and specifications that manage WSN features such as communication and connectivity, networking and routing topology, distribution and deployment, energy efficiency, network lifetime, security, etc. WSN Communication standards

are usually short-ranged and divided into two broad categories: Wireless Local Area Networks (WLANs) and Wireless Personal Area Networks (WPANs). Fig. 2.5 shows the structure and divisions of short-range wireless networks. Low-power standards are the most used in WSNs. These include IEEE 802.15.4/ZigBee, IEEE 1451, IPv6 over IEEE 802.15.4, WirelessHART, Z-wave, Ultra-Low-Power Bluetooth, sub-1GHz RF standards. Table 2.10 presents a comparative overview of the various characteristics of some of these standards [82]–[86].

This work deployed a two-tier network involving a variant of the IEEE 802.15.4/ZigBee and the sub-1GHz RF transceivers. The ZigBee standard networked the sensor nodes in the environment to those on the robots, while the sub-1GHz RF standard networked the nodes on the robots to the base station. These standards are discussed briefly in the proceeding sections.

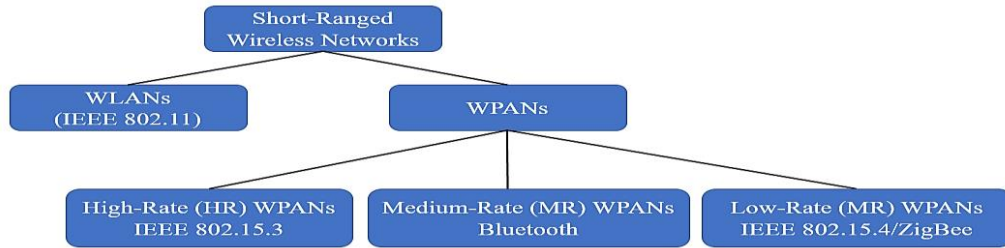


Fig. 2.5. Wireless Networking Categories

Table 2.10. Comparison of Low-rate/Low-power Wireless Communication Standards

Wireless Protocol	6LoWPAN	Bluetooth	ZigBee	WIMAX	Zwave	Sub-1GHz RF
Network	WPAN	WPAN	WPAN	MAN	WPAN	WLAN
Standard	IEEE 802.15.4	IEEE 802.15.1	IEEE 802.15.4	IEEE 802.16	Z-Wave	IEEE 802.24, IEEE 802.11ah, WMBUS
Frequency Bands	868Mhz (EU) 915Mhz (USA) 2.4Ghz (Global)	2.4 GHz	2.4 GHz	10-66 GHz	868 MHz-908 MHz	300-348 MHz, 387-464 MHz and 779-928 MHz
Topology	Star Mesh	Star-Bus	Star, Mesh Cluster Tree	Mesh	Mesh	NA
Power	Very Low	Low	Very Low	High	Very low	Ultra Low
Data rate	250kbps	2.1Mbps	250kbps	11-100Mbps	40kbps	Up to 600kbps
Range	Short Range 10-100m	Short Range ~15~30m	Short Range 10-100m	Long Range 50km	~30m	Long range 0 – 10km
Network Size	Very large	Small	Very large	Large	Large	Very Large

B. The Link Layer (MAC Sub-Layer)

The MAC layer is also under the management of IEEE 802.15.4. It is responsible for controlling the shared channels available for connection and transmission/reception and ensuring reliable data deliveries. It makes sure the network is performing optimally. It has two operational modes [87]:

- a. **The Beacon-enabled Mode:** Where beacons are periodically sent through the network to synchronize nodes and identify the Personal Area Network (PAN). Medium access is operated by slotted CSMA/CA and Guaranteed Time Slots (GTS) algorithms.
- b. **The Non-Beacon-enabled Mode:** This mode makes the medium access to be operated strictly by the unslotted CSMA/CA Mechanism since no beacons are sent over the network.

C. The Network Layer

From the network layer to the upper layers, the protocol is managed by the ZigBee Alliance. The network layer determines the routing, topology, and type of devices [87], [88].

- a. **Routing:** The ZigBee protocol support three types of routing namely Neighbor Routing based the availability and reachability of the all the neighboring devices within communication range; Table Routing (Ad-hoc On-Demand Distance Vector AODV), based on the efficiency, discoverability and the cost of routing paths; Tree-Routing, based on device addresses and hierarchy of nodes upstream/downstream
- b. **Topology:** ZigBee supports three types of topology, namely Star Topology which is a centralized network system with a coordinator that controls communication in the network; Mesh Topology is a decentralized network system with coordinators and routers also, but every node can communicate with other nodes within their radio range; Cluster-Tree Topology which is like a hierarchical mesh topology with several clusters having routers and connect to one coordinator. A comparison of these topologies is shown in Table 2.11, while Fig. 2.7 shows the three topologies.
- c. **Type of Devices:** These include ZigBee Coordinator (ZC) or Gateway which initiates, configure and synchronizes the network and implements the full protocol, hence, are called Full Function Devices (FFD); the ZigBee Router (ZR), another FFD, participates in rerouting messages in the network; the ZigBee End Device (ZED), which does not route nor allow other devices to associate with it, it is just a sensor/actuator node, hence, implements a reduced set of the protocol and is usually called Reduced Function Device.

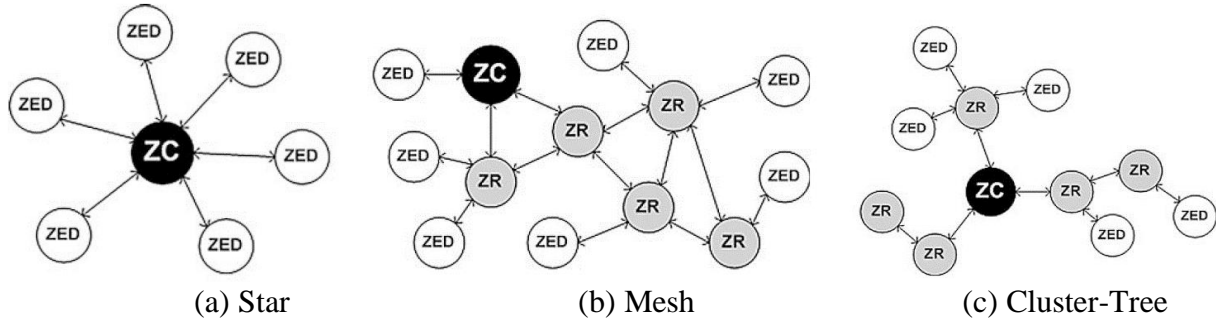


Fig 2.7. IEEE 802.15.4/ZigBee Network Topologies [87].

Table 2.11. Features/Comparison of ZigBee Topologies [87].

Features	Star	Mesh	Cluster-Tree
Scalability	No	Yes	Yes
Synchronization	Yes (no)	No	Yes
Inactive period	All nodes	ZEDs	All nodes
Guaranteed bandwidth	Yes (GTS)	No	Yes (GTS)
Redundant paths	N/A	Yes	No
Routing Protocol Overhead	N/A	Yes	No
Commercially available	Yes	Yes	No

D. The Application Layer

This is the highest layer of the protocol and it is usually determined by manufacturers of ZigBee technologies based on the ZigBee Alliance standards [88]. This has three sections, namely the Application Support Sublayer (APS) which serves as an interface of the network and application layers; the ZigBee Device Objects (ZDO) which supports the interface for APS and Application Framework (AF); and finally, the Application Framework (AF) which provides the application profiles for developing applications.

2.2.3 Overview of Sub-1Ghz RF Standards

Sub-1GHz standards refer to wireless RF protocols that operate on frequencies below 1GHz (315, 433/434, 470, 868, 915 MHz) bands [89]. They are ultra-low power and cover very long ranges. Their strengths against the wireless standards operating the 2.4GHz bands are presented in Table 2.12, [89], [90]. Example sub-1GHz standards include LORA, SIGFOX, etc.

Table 2.12. Sub-1GHz Standards compared to 2.4GHz Standards

Feature	Sub-1GHz Standards	2.4GHz Standards
Range	Long-range of up to 10km	Short ranges
Power	Ultra-low power consumption	Very low power consumption

Interference	Less interference due to fewer devices operate within the band frequencies	More interference, since most devices operate within the band frequencies
Non-line of sight performance	Better performance, since RF waves easily squeeze through and around walls	Lesser performance, with degrees of obstruction based on the particular standard and protocol used.
Antenna Size	Larger antenna sizes	Smaller antenna sizes

Since the sub-1GHz frequency bands are under the unlicensed ISM bands, there exist several implementations and protocols as developed by various manufacturers. This work used the Wireless Serial (UART) 433/434MHz sub-1Ghz protocols based on Texas Instruments chip CC1101 [91]. This chip was incorporated with other chips to facilitate the UART interface. The outcome was the HC-11 Wireless UART module [86].

The module by default operates the half-duplex communication mode, where communication is not simultaneous between two modules. But further settings on the Wireless UART transparent modes offer the ability of the module to operate on the full-duplex mode where two modules can both send and receive data at the same time. Fig. 2.8 illustrates the module connectivity.

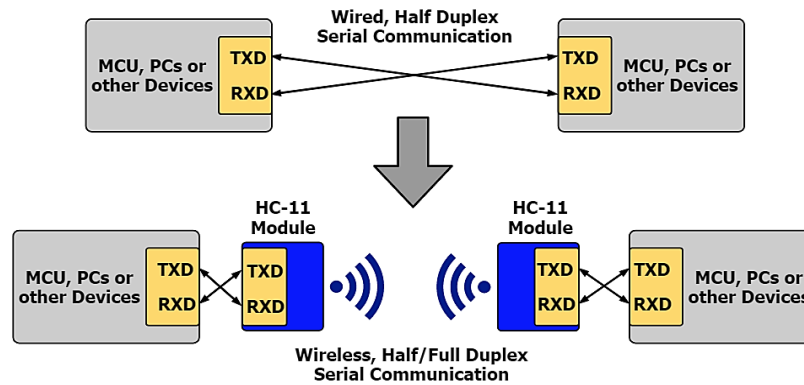


Fig. 2.8. Wireless Serial (UART) Communication

2.2.4 Related Works of WSNs in Surveillance

For surveillance applications, there are two broad types of WSNs, one network involving imaging sensors and the other involving non-imaging sensors. Imaging sensor networks mostly deploy cameras, while non-imaging networks deploy low-cost monitoring sensors. Therefore, surveillance cameras have been used for both fire and intruder detection systems in [92]–[95]. Surveillance systems using non-imaging sensors include works presented in [96]–[99]. Table 2.13 and 2.14 presents a review analysis of works in both approaches.

Table 2.13. Review analysis of example applications of imaging sensors surveillance systems

Ref.	Features Observed	Feature Extraction Technique	Test Method	Results (Achievements)	Challenges/Future Work
[92]	Visible flame and Smoke	Robust AdaBoost	Classification tests on video datasets	Effective flame and smoke detection with 91% average accuracy.	Sample denoising of extracted features.
[93]	Visible flame with smoke	Tchebichef Moment Invariants with PSO-SVM	Videos of fire and non-fire scenarios	Effective flame and smoke detection with 98% average accuracy.	Deployment of system in an actual application
[94]	Whole video scenes	Smoothing, filtering, estimation-based algorithms	Summary of video frames based on objects detected	Successful classification of events in video frames	No real-time test.
[95]	Humans	Image Segmentation techniques	Video tests carried using Raspberry Pi connected camera and PC	80% human motion detection accuracy in indoor environments with SMS and email alert for system users	Development of system to include outdoor environments

Table 2.14. Review analysis of example applications of non-imaging sensors surveillance systems

Ref.	Sensor Types	Event Classification Technique	Test Method	Results (Achievements)	Challenges/Future Work
[96]	Flame, Smoke, Temperature, Gas and Wet	Dempster-Shafer evidence theory. (Fire Detection)	Hardware Implementation	Effective fire detection with fewer uncertainties	
[97]	Temperature, Humidity, CO ₂ , CO and light	None. Sensing and communication (Environment Monitoring)	Hardware implementation with phone and web apps	Effective environmental monitoring and data communication	More tests to further validate WSN architecture
[98]	Smoke, Temperature, Light	Dempster-Shafer evidence theory (Fire Detection)	Hardware Implementation with web, phone and email notification system	Effective fire detection with 97% accuracy in daytime and 89% in the nighttime	
[99]	Temperature and Humidity	None. Sensing and communication	Hardware implementation with a phone app	Effective communication of environmental data to smartphone.	Addition of air quality sensors

2.3 Related Works in ANR (MRS Integration with WSNs)

As stated in [78], the three major areas of WSN integration with MRS include: control, navigation and localization. These areas of research then stretch into applications such as search and rescue, exploration and coverage, and data acquisition and surveillance.

2.3.1 Related Works in General ANR Systems

Authors in [100] proposed a WSN architecture to control mobile robots in an area monitored by an unstructured WSN. They compared the control performance of WiFi (IEEE 802.11) and IEEE 802.15.4 based networks. After simulations on TinyOS and TOSSIM, and implementations on Pioneer P3-DX robots and the Mensic's iMote2 sensor platforms, the WiFi-based networks offered a more robust control to the IEEE 802.15.4 networks. Further studies envisaged by the authors included using dynamic protocols that could determine how robots enter and leave the network. Several SLAM methods have been applied for localization in MRS and WSN. WSN localization measurements like RSSI have been combined with SLAM algorithms such as EKF and Delayed Particle Filter (DPF)-EKF [101], Particle Filter (PF) Extended Information Filter (EIF) EKF [102] and EKF only [103]. These works reported strong points for the consideration of SLAM based algorithms for localization of robots and nodes alike in WSN. In navigation, authors in [104] described a navigation system for multirobots which involved an initial map building of the environment, followed by an online path planning around the environment. The experimented system proved that based on the offline map, the robots could plan and navigate their path successfully in the environment in real-time.

2.3.2 Related Works of ANR Systems in Surveillance

The many advantages of ANR systems make them favorable for surveillance applications. Table 2.15 presents a review analysis of example applications.

Table 2.15. Review analysis of example ANR systems in surveillance

Ref	Application	System Features	Test Method	Achievements	Challenges/ Future Work
[105]	Indoor surveillance for intruders	<ul style="list-style-type: none">▪ Single robot system▪ ZigBee WSN system▪ Monte Carlo localization and wavefront path planning	Hardware implementation	Successful robot navigation to event tasks as notified by the WSN	Improvements on robot localization

[106]	Indoor surveillance for intruders	<ul style="list-style-type: none"> ▪ Single robot system ▪ ZigBee WSN 	Hardware implementation	Successful WSN event detection, robot navigation to sensor node location and event report to base station	Packet loss due to multihop system
[107]	Indoor surveillance for intruders	<ul style="list-style-type: none"> ▪ MRS/WSN system ▪ Wi-Fi communication system ▪ MRS coordination and task allocation 	Simulation on Microsoft Robotics Developer Studio (MRDS)	Successful MRS intruder detection and navigation	Actual system implementation
[66]	Indoor surveillance for intruders	<ul style="list-style-type: none"> ▪ Two-tier network (6LoWPAN & Wi-Fi). ▪ MRS ▪ Market-based task allocation. ▪ Hybrid GA/ACO path planning 	Simulations on Stage/Player Simulation Environment and Matlab	Good robot path planning and system networking	No implementation
[108]	Surveillance for indoor heat and lighting conditions	<ul style="list-style-type: none"> ▪ ZigBee based network ▪ Single robot system 	Implementation using com. modules and fabricated robot	Successful communication between robot and WSN	Making the system an MRS, adding actuators, etc.
[109]	Indoor surveillance for visible flame and combustible gas leakage	<ul style="list-style-type: none"> ▪ MRS/WSN system ▪ Wireless RF communication between robot teams and base station ▪ Dempster-Shafter evidence theory for event detection classification 	Hardware implementation	Successful event detection, classification and communication with base station	Integration of static WSNs in the environment
[110]	Indoor surveillance for human abnormal behaviors	<ul style="list-style-type: none"> ▪ MRSN/RFID WSN system ▪ Prey-predator coordination scheme ▪ Use of cameras and RFID tags 	Simulations and hardware implementation	Successful tracking of humans (preys) by the predators (robots) based on detected behaviors by RFID WSN	Improving system architecture to increase robustness and precision.

2.4 Research Challenges and Open Areas

Despite the advances in ANR systems research, there still exist several areas of growth and advancement. Both MRS and WSNs that makeup ANR systems can be improved on and be suited for particular applications such as surveillance. Therefore, this section presents open areas in ANR systems research in relation to surveillance systems.

2.4.1 Challenges in MRS for Surveillance Applications

From the above example applications of robots in surveillance applications, the following open areas have been identified [26], [111]:

- i. Deployment and coverage challenges in MRS based surveillance systems.
- ii. Effective architectures for cooperation and task distribution between the MRS and WSNs.
- iii. Effective yet simple algorithms for localization and path planning.

2.4.2 Challenges in WSN for Surveillance Applications

From the example applications of WSN in surveillance and deployment of WSN in ANR systems for surveillance, two major solutions were discussed:

- i. Solutions involving imaging sensors (video surveillance) for detection of any kind of event (fire, intruder).
- ii. Solutions involving non-imaging sensors (for example temperature/humidity sensors, gas sensors, etc.).

The basic requirements for effective and reliable surveillance are presented below [112]:

- Reduce system interference, interruption and collision during data communication.
- Reduced false alarms and fast alarm transmission.
- Location encoding in data communication to ensure directed response.
- Cost effective and long system life cycles.

Based on these requirements, the following challenges have been identified with the current solutions:

- i. Cost of installation, operation, and maintenance in video surveillance systems.
- ii. The challenge of data processing techniques performance for faster detection in real-time applications, especially in video surveillance systems.
- iii. Appropriate hardware selection to handle data processing demands, especially in video surveillance systems.
- iv. Data loss due to network overload in monitoring systems using only one type of communication protocol before connecting to the Internet.

Therefore, it can be stated that there is a need to develop a non-imaging surveillance system capable of effective fire and intrusion detection using a variety of sensors with further

enhancements such as real-time integration with cloud technologies. Also, the system should include more than one communication protocol preventing data loss due to network overload. This work, therefore, presents an ANR system for indoor surveillance with the following features:

- i. A two-tier network WSN system comprising of a ZigBee network connecting deployed sensor nodes to the nodes on the MRS, and a Wireless Serial (UART – sub-1GHz) network connecting the robots and the base station.
- ii. Sensor nodes that contain non-imaging sensors for both fire and intruder detections with an effective logic system for event classification
- iii. MRS comprising robots that navigates an environment that has divided into zones, with each zone assigned to a specific robot.
- iv. A base station visualization system with cloud integration enabling IoT applications such as remote monitoring, visualization, and analytics.

Chapter 3

SYSTEM REQUIREMENTS AND RESEARCH METHODOLOGY

This chapter presents the requirements and methodology for the development of the ANR system. The ANR system features actions such as sensing by the static and mobile nodes alike, localization of both nodes, exploration, task distribution, and path planning of the mobile nodes, which are common to most WSNs/MRS systems [111]. Hence, the scope of the system is presented, covering all possible cases and extensions of the system, followed by the operational scenario description, with the capabilities of the various system components presented. The functional requirements present the types of functions to be executed by each system level and agent, while the research methodology outlines the steps involved in the system development, testing, and evaluation.

3.1 Scope of Research

There is a growing need to develop systems that can fulfill task requirements covering large operational spaces of complex applications with possibility to have dispersed and distributed spaces and facilities. One of such applications but not limited to are the surveillance (warehouses, factory floors, storage areas) for fire and intruder detection. Such coverage will demand to have effective functioning systems that are featured by the following:

- a. Effective structuring of task operational space that enables the work at local operational space of the task with the ability to integrate the overall functions and activities of these subspaces at the global level of the task.
- b. Have sensing and functional capabilities dedicated to each part of the local space.
- c. An ad hoc network of sensor nodes can be formed within each local space with the capability to sense, process, communicate, and relay the information to the other levels of the task space.
- d. Mobile devices assigned to each local space that has sensing capabilities supporting its navigation and also sensors related to the task requirements.

- e. Mobile devices assigned to the global task space that has sensing capabilities supporting its navigation and sensors related to the task requirements covering areas not associated with any of the local spaces.
- f. Effective communication protocol that facilitates communications and actions between local spaces, mobile devices, and the global task space.
- g. Effective data aggregation, storage, visualization, and analytics managed at the global level and shared with cloud platforms to allow remote access by entities such as mobile devices.
- h. Effective system coordination and monitoring using visualization interfaces that support simulation environments, HIL, and physical implementation.

3.2 System Operational Scenario

An ANR system that features MRS and WSNs is therefore developed to fulfill the above research scope for a given hybrid task space (indoor and/or outdoor) such as a warehouse, which is referred to as a global space. The operational task space is divided into local spaces called zones. Each zone can be part of the same physical local task space or be remotely accessible. The activities within each zone are interacting with the relevant mobile device and higher levels of the task hierarchy as follows:

A. At the Zone Level

- i. Deploy at each zone the necessary sensor nodes to cover its space. Each sensor node constitutes single or multiple sensors, a processor, communication and dedicated power supply. Each of these sensor nodes are static, hence, they are called static sensors nodes (SSNs). Also, each SSN has a known location within the zone with reference to the global space.
- ii. Each SSNs have the following capabilities:
 - a. Sensing capabilities to detect one or more parameters within the zone.
 - b. Processing sensor data.
 - c. Relay and communicate the processed data.
- iii. The interrelation of the sensor data from a single or multiple sensor nodes can formulate an action list. An action capability that affects the status of a part or all the operational space within the zone can be triggered when a relevant situation has been detected. Such action can be, for example in case of fire monitoring and detection, an activation of suppression mechanisms (such as sprinklers, alarms) by

a relevant SSN within the operational zone or requiring a mobile device to navigate to the zone for further analysis.

B. At the Mobile Device Level

- i. Deploy mobile devices such as a mobile robot to each zone.
- ii. The mobile robot should have sensors to support motion control, navigation and obstacle avoidance.
- iii. The mobile devices have an onboard static sensor node (O-SSN) similar to SSNs, for event detection within an assigned zone. The mobile robot with its O-SSN is called Mobile Robot Sensor Node (MRSN), and it has the following capabilities
 - a. Detect events around the robot while moving within its task space using its O-SSNs and communicating them to the base station.
 - b. Receive data from the SSNs within its assigned zone.
 - c. Relay its detected data and SSN data to higher levels of the task hierarchy through its O-SSN.
 - d. Navigate to SSN locations to assess events detected.
 - e. Request the G-MRSN to navigate to an SSN location when it is not within the reference navigation distance required to navigate to the SSN location.
 - f. Navigate continuously along the assigned paths within the relevant zone.
 - g. Perform other tasks such as item lifting and transportation, pick and place, etc.

C. At the Global Mobile Device Level

- i. Deploy a global mobile device to cover areas not covered by any of the zones with the possibility to extend support to MRSN associated with the zones. This device can be called Global-MRSN (G-MRSN) and it has the same sensing capabilities of MRSN to support motion control, navigation and obstacle avoidance, and have the following capabilities:
 - a. Navigate continuously paths of operational task space not associated with any zones detect and relay events within these areas to the IoTBS. While navigating these areas, the G-MRSN can navigate to zone SSN locations based on the zone MRSN request.
 - b. Communicate with MRSNs and relay any necessary information from MRSNs to the IoT BS.

D. Higher Level of Task Hierarchy

- i. The higher level of task hierarchy is considered to be an IoT base station (BS) with the following capabilities:
 - a. A base station gateway (BSG) to receive data from the O-SSNs.
 - b. Aggregate, analyze and visualize data received through the BSG.
 - c. Store the data locally within the IoT BS and globally for accessibility by other entities using cloud platforms.
 - d. Decision making based on the aggregated data.
- ii. Coordinating and monitoring all the zones within the task space using visualization interfaces that support simulation environments, HIL, and physical implementations.

Fig. 3.1 shows the hierarchical system levels, while Fig. 3.2 illustrates the overall system scenario, with event detections in zone 3 and 4. Fig. 3.3 shows the operational flow chart.

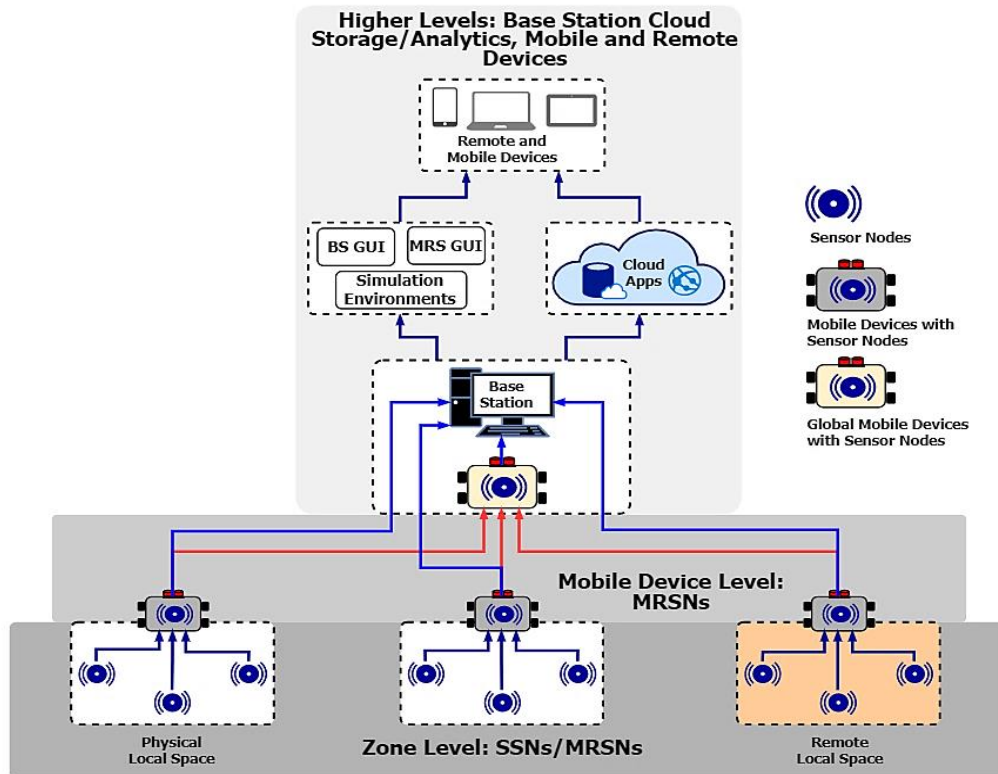


Fig. 3.1. System Scope and Hierarchy

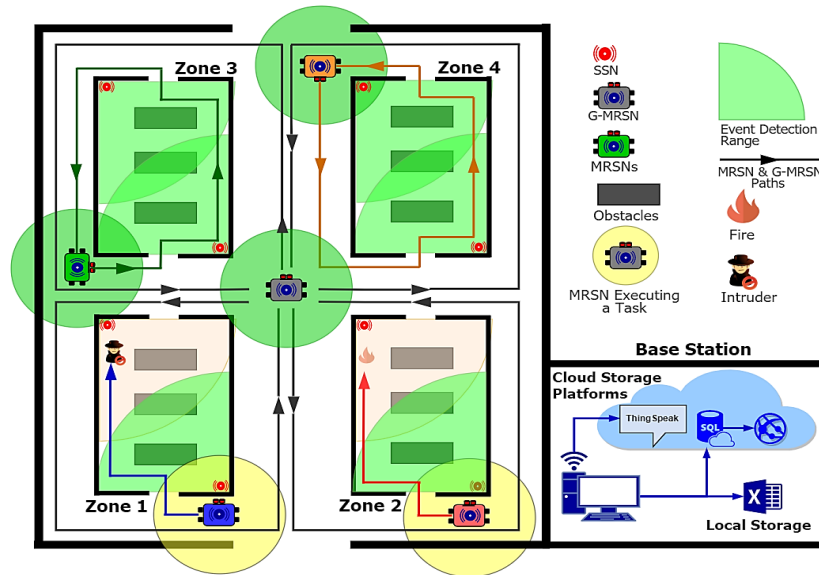


Fig. 3.2. System scenario

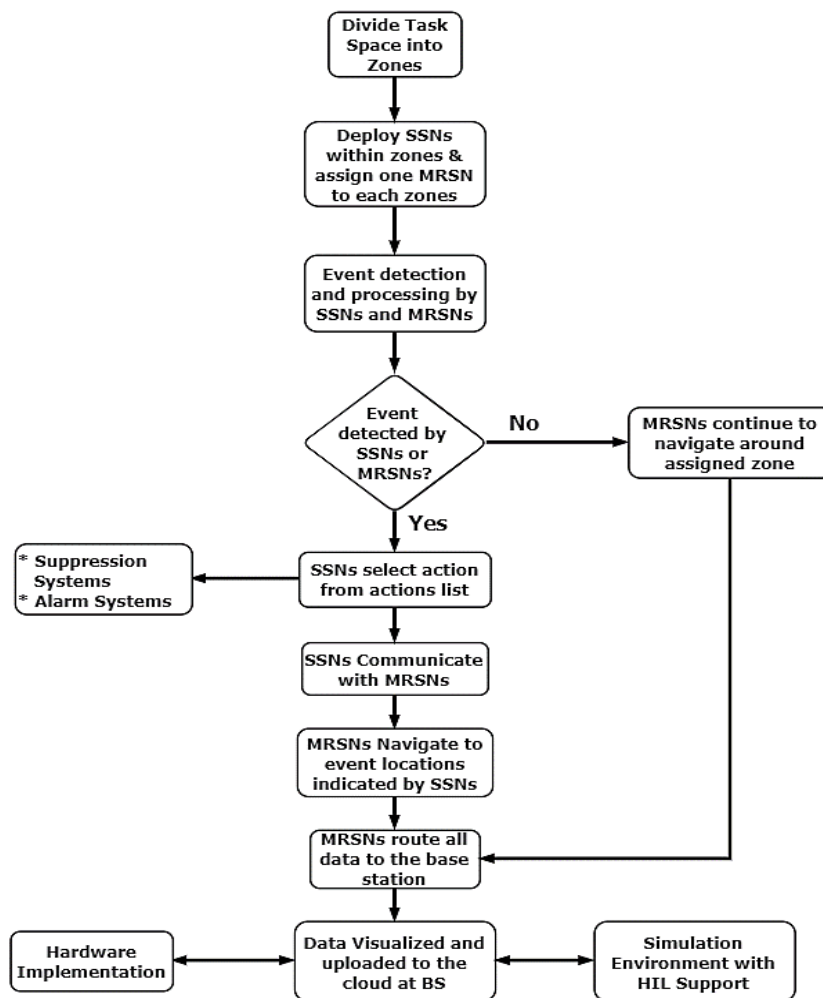


Fig. 3.3. System operational flow chart

3.3 Functional Requirements, Assumptions and Constraints

Considering the application of warehouse fire/intruder detection, the developed ANR system has the following functional requirements, assumptions, and constraints.

3.3.1 Functional Requirements

The functional requirements of the system at each level of operation are as follows:

A. At the Zone level

1. Detection of fire and/or intruder events.
2. Process and classify events detected in normal, high or critical alert events.
3. Chose an action (task) from the action lists.
4. Communicate event detection data to the assigned zone mobile (MRSNs).

B. At the MRSNs Level

1. Detection of fire and/or intruder events using onboard sensors.
2. Communicate event data from SSNs and O-SSN to the global task space level (BS) through the O-SSN.
3. Navigate to SSN locations to assess events detected.
4. Coordinate with G-MRSN, when MRSN is not within the reference distance to the alerting SSN.
5. Continuously navigate assigned paths within/around its zone.

C. At the Global Level (G-MRSN and IoT BS)

1. G-MRSN navigates, detects and relay events to the IoT BS around non-zoned areas of the global operational task space.
2. G-MRSNs navigate to zone SSNs upon request of zone assigned MRSNs.
3. IoT BS aggregates and visualizes all zone data from the O-SSNs communicated through the BSG.
4. IoT BS stores data both locally and/or upload to cloud storage platforms.
5. Make data accessible for remote aggregation, visualization and processing.
6. Provides a platform for the system simulation, HIL based simulation and implementation featuring the following:
 - a. WSN simulation featuring SSNs, O-SSN and BSG models.
 - b. MRS simulation featuring navigation environments, MRSNs and G-MRSN models, obstacles, walls, etc.

- c. Network emulator for facilitating data exchange between the various simulation environments.
- d. HIL support for the testing and validation of hardware modules before implementation.

D. Communication Requirements

- 1. A network protocol to facilitate communication between the zone SSNs and the MRSN O-SSNs.
- 2. A separate network protocol to facilitate communication between the MRSN O-SSNs and the BSG, thereby reducing network interference.

3.3.2 Assumptions

- 1. All MRSNs have similar characteristics, i.e., maximum speed, and physical abilities such as wheels, motors, motion control, and navigation sensors.
- 2. All MRSNs perform similar tasks involving navigating to SSN locations and navigating around their assigned zones.
- 3. The SSNs are connected to external power that supports continuous and sustained operation.
- 4. The robots are continually surveilling the area of the zone except when they need to charge up their batteries at the charging stations.
- 5. The obstacles in the operational setup, i.e. the storage items were assumed to be static, making the operational environment to be considered static.
- 6. The task environment floor is flat, and the zones can be of different dimensions.
- 7. The operational environment is considered to be large, for example, a warehouse, storage facility, based on warehouse size analysis [113].

3.3.3 Constraints

- 1. The reference distance between any SSN and MRSN associated with its zone required for the MRSN to navigate to the SSN is the effective detection range of one SSN, which corresponds to the effective detection range of the motion sensor, i.e. 7m [114].
- 2. Communication protocols constraints such as end-to-end delay, throughput, packet data payload sizes are applicable.
- 3. Attenuations of the RF signals from walls, equipment, furniture, etc. are applicable.

3.4 Research Methodology

The development of the ANR system involves the following steps, which are now presented.

1. The ANR system concept formulation.
2. The ANR system design scope and operational scenario.
3. The ANR system development, involving the following stages:
 - a. The design of the overall system architecture and coordination approach
 - b. The development of simulation models and simulation environments.
 - c. The system simulation and HIL based simulation using simulation models, simulation environments and hardware components, and
 - d. The hardware tests and assessments.

3.4.1 ANR System Conceptualization

The ANR system features MRS collaborating with WSNs. The system introduction and review show the benefits, challenges, and applicability of the system, which has been covered in chapters 1 and 2.

3.4.2 ANR System Design Overview

The ANR system design overview presents the system's research scope and operational scenario, whereby a general view of the system is presented, to which several applications with different types of task spaces can be developed. The research scope and operational scenario can vary based on the type and scale of application, the application task space, etc. The scope and operational scenario have been presented in sections 3.1 and 3.2. For this work, the chosen application is fire/intruder detection in storage task spaces such as warehouses using non-imaging sensors and mobile robots. With a defined application, the rest of the system framework can be formulated and developed.

3.4.3 ANR System Development

The ANR system development involves the design structures that support the ANR system development based on the scope, operational scenario, and application. They include the following:

A. ANR System Architecture Design

The ANR system design involves the formulation of the MRS/WSN architecture which is the framework upon which the whole system is developed. The ANR system has a hybrid architecture

that features a two-tier network system that enables the system simulation, hardware-in-the-loop (HIL) based simulation and real-time operation. The architecture constitutes the WSNs for event detection and communication, MRS for navigation within operational environment and IoT BS for data aggregation, visualization, and connection to cloud platforms

B. ANR System Simulation Development Environment

The ANR system simulation development environment design involves the development of simulation models and simulation environments, their integration, which enables the testing and validation of the system.

C. ANR System Simulation Tests and Evaluations

The ANR system simulation involves:

- a. **The WSNs Simulation** consists of the simulation of the event detection and communication through the SSN-to-MRSN O-SSN and the MRSN O-SSN-to-BSG networks.
- b. **The MRS Navigation Simulation:** involves the simulation of the MRSNs and G-MRSN navigation within the given environment involving path planning and coordination to alerting SSN locations and navigation of assigned paths within the given environment.
- c. **IoT BS Simulation** involves the aggregation, visualization of data at the IoT BS and the connection to cloud platforms for storage and analytics.
- d. **HIL Simulation** involves testing and validating of hardware components of the system through HIL simulations.

D. ANR System Hardware Implementation

The ANR system hardware implementation involves the operation of the developed hardware system components in a lab-based environment.

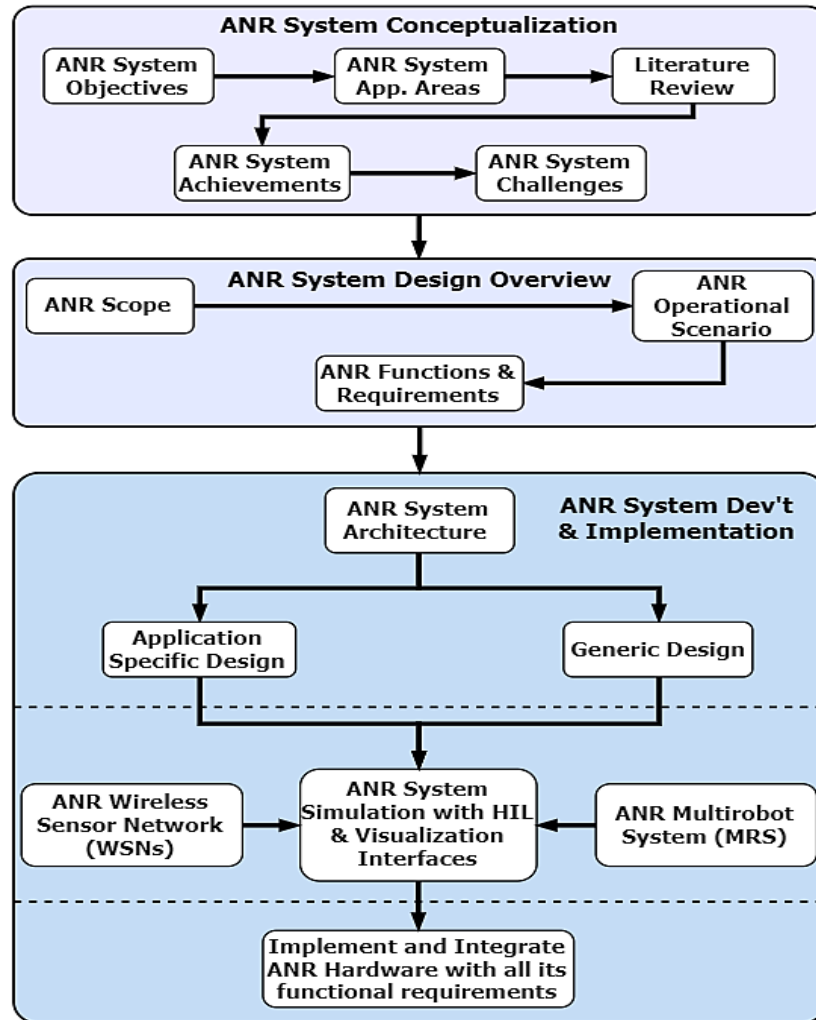


Fig. 3.4. ANR System Development Methodology

Chapter 4

ANR SYSTEM ARCHITECTURE

The ANR system architecture is the framework upon which the principle conceptual design of the system with its various components are developed and interacted to meet the system requirements. Each major component of the ANR architecture, including the WSNs, the MRS and the IoT BS are presented. For the WSNs, the sensor node architectures, decision system, and operation pseudocodes are presented. The MRS presents the overall structure, coordination and navigation algorithms of the MRSNs and G-MRSN. Finally, the IoT BS interfaces for sensor data and MRS navigation data visualization supported by cloud integration technologies are introduced.

4.1 ANR System Architecture Overview

The developed ANR system is based on a hybrid architecture composed of the WSNs, MRS and the IoT BS. The WSNs constituting the SSNs represent the primary component that detects task-related events within its operational zone, triggers the proper actions and communicates all sensing information to the relevant robots associated with the zone. The MRS constitutes of the MRSNs and the G-MRSN; both types of robots contain sensors used for control and navigation, and onboard static sensor nodes (O-SSNs) used to detect task-related events within the operational zones. While navigating within and around their assigned zones to SSN locations, the MRSNs O-SSNs receive sensing information from the relevant zone sensor nodes and relay them together with their own detections to the IoT BS. The G-MRSN mainly detects events and relay them to the BS while navigating areas not associated with the zones. However, it sometimes navigates to SSN locations based on request by the MRSN. Navigation data from the MRSN and G-MRSN are also communicated to the IoT BS. The IoT BS receives information from the robots, aggregate, visualize and upload the information to cloud platforms for storage and analysis. This hybrid architecture facilitates communication and coordination of WSNs, MRS and the IoT BS, making the system robust, scalable and reliable. The developed ANR architecture is supported by the development of a simulation development environment (SDE) with HIL that facilitates the ANR system testing and validation.

The block diagram for the developed ANR system architecture is shown in Fig. 4.1. At the center of the diagram is the core platform comprising of BS interfaces representing the sensor node data visualization and MRS navigation GUIs, and also the IoT/Cloud integration platforms. These

GUIs can exchange data either with components from the simulation environments only, the physical environments only, or having HIL merging components from the simulation and physical environments. The architecture layout is illustrated in Fig 4.2, in which the SSN detect events within the zones. The event data is then communicated to the MRSNs, which in turn relays it to the IoT BS together with its own detected events. The G-MRSN also relay detected events by its onboard sensor node to the IoT BS. The SSNs-to- MRSNs communication protocol is separate from the MRSN-to-BS/G-MRSN-to-BS/MRSNs-to-G-MRSN communication protocol. Hence, the architecture operates on a two-tier network system.

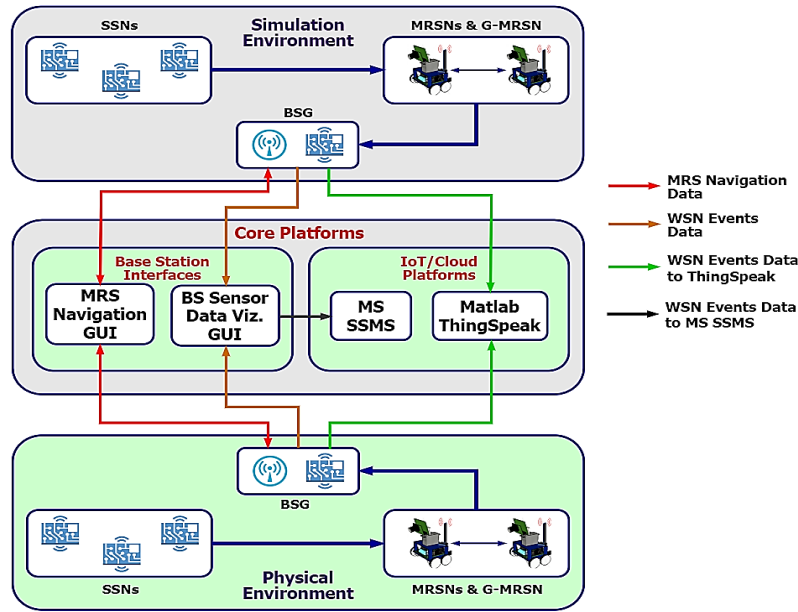


Fig. 4.1. ANR system development block diagram

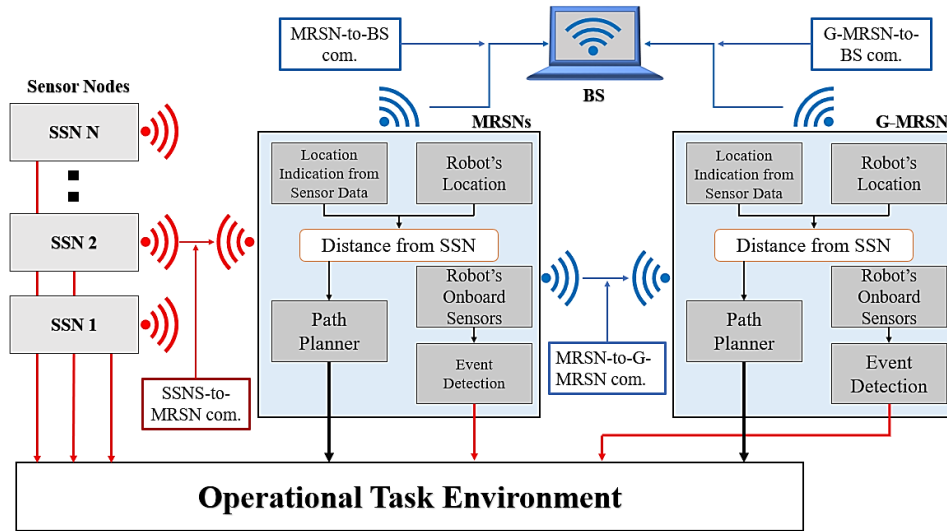


Fig. 4.2. ANR system architecture layout.

The two-tier network system involves two separate communication protocols facilitating data exchange between the SSNs, MRSNs, G-MRSN and the BS. The first network involves the ZigBee/IEEE 802.15.4 (2.4 GHz) protocol connecting the SSNs to their respective MRSN. The star (centralized) network topology is deployed with the SSNs as the end modes and the MRSN as the coordinator node. Also, within the ZigBee star network, Zone1 SSNs used separate channels of communication from Zone2 SSNs. The second network connects the MRSNs, G-MRSN and the BS and it uses the Wireless Serial (UART) (434 MHz) protocol. The overall system network topology could be considered as a tree topology, with the BS as the sink, as shown in Fig. 4.3. The use of separate communication protocols, channels and frequency bands for system networks ensured that traffic data loss due to interference was reduced [115], [116].

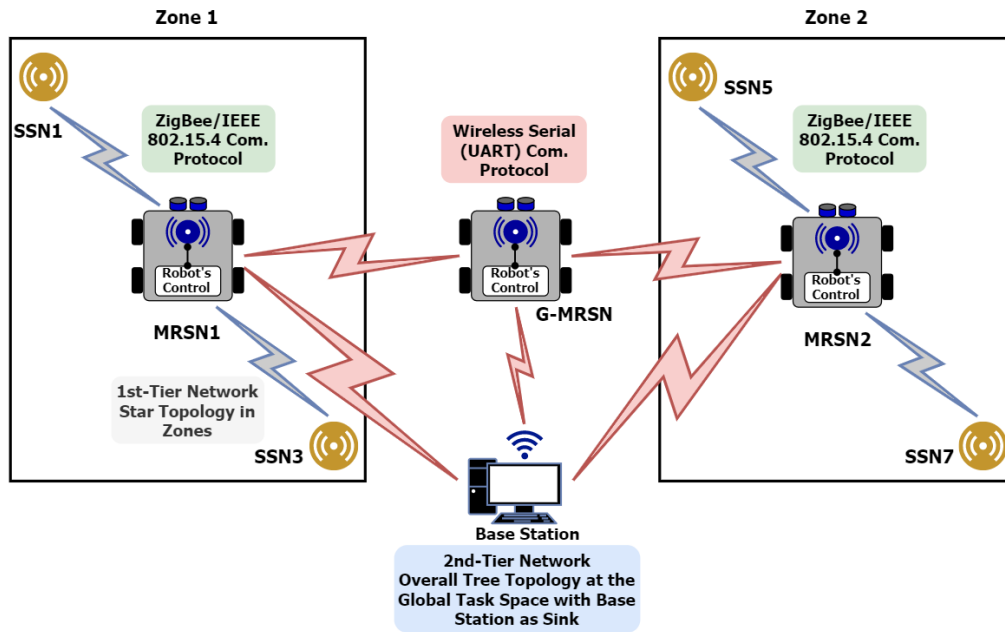


Fig. 4.3. The two-tier network system, forming an overall tree topology network

4.2 SSNs Architecture

The SSN architecture comprised the sensor node structure, the event characterization system and the sensor node pseudo algorithm. These features of the SSNs facilitate the detection, data processing and communication of events from the zones to the MRSNs. The SSNs and O-SSNs on the MRSNs and G-MRSNs are similar in architecture and operation except for differences in types of sensors, and communication modules and protocols.

4.2.1 SSN Structure

The SSN structure is based on the generic sensor node structure and it comprises gas, temperature/humidity, flame, motion sensors, a ZigBee communication module, LEDs and LCD indicators, microprocessor and power source. The flame sensor has a maximum range of 1m while the motion sensor has an effective range between 7 – 12m. Each SSN has 6 LEDs – 3 Green and 3 Red. The green LEDs indicate normal conditions, while the red LEDs indicate high or critical conditions. The LCD displays a text message reflecting the overall outcome of the detected event, either “Normal,” “High” or “Critical.” Also, it shows the corresponding temperature and humidity values, and reports the presence of motion, visible flame and combustion gas (Carbon-monoxide – CO gas and Smoke) using text messages such as “YES” or “NO.” The SSN structure is shown in Fig. 4.4(a). The operational process diagram of each SSN is shown in Fig. 4.4(b), and the process starts from the operational environment where the various sensors sense the occurrence visible flame, CO gas, smoke or motion from intruders. Sensed analog data are obtained from the gas sensors and temperature/humidity sensor, while the output of the flame and motion sensors are digital. The sensory information is then inputted into the data processing block comprising of the event characterization system, which combines a fuzzy logic rule system (for analog data) and the digital outputs for event characterization. Detected events can be characterized as “Critical” or “High Alert” whereby the SSN triggers an action from the action list, or “Normal,” and are communicated to the MRSN O-SSNs via the ZigBee RF protocol. The details of the event characterization system are presented in section 4.2.2.

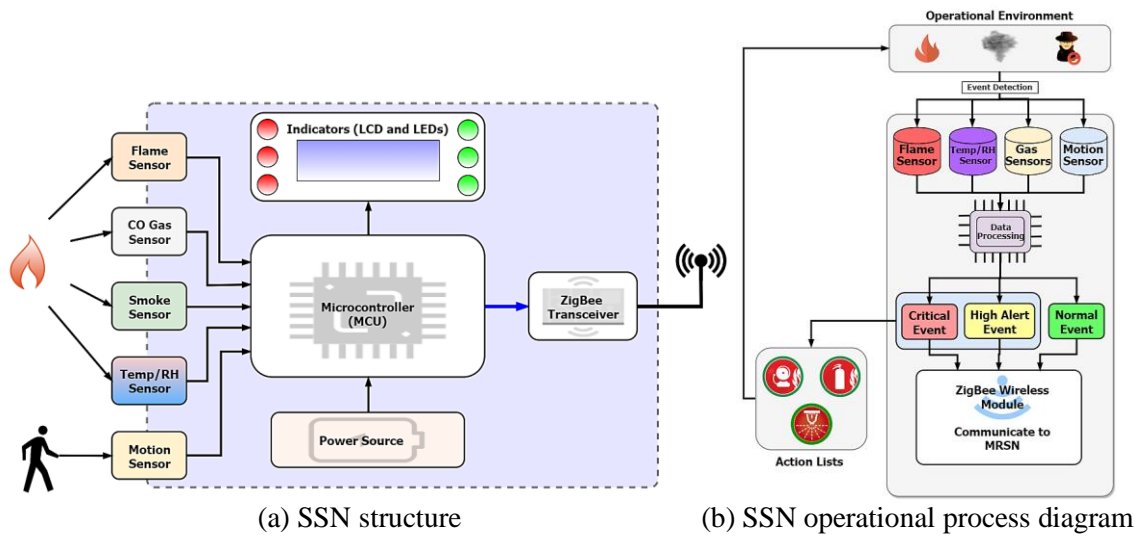


Fig. 4.4. SSN architecture.

4.2.2 Event Characterization System

Within each SSN, there are different sources of sensory information, and to specify the relevant event using this information, there is a need to develop a sensor data fusion technique that receives such information for the purpose of characterizing the event as necessary. In addition, the use of sensor data fusion technique can contribute to reducing the possibilities of errors ensuring accuracy in the event characterization [98].

There exist many sensor fusion algorithms such as Bayesian estimation technique, Dempster-Shafer (D-S) evidence theory, neural networks, fuzzy logic rule systems, neuro-fuzzy, etc., [96], [98], [117], [118], with the probability-based belief system D-S evidence theory and the fuzzy logic rule system, being the most commonly used [96], [98]. The D-S algorithm is mostly used in sensor data fusion problems involving uncertainties where all the possible outcomes Θ are contained in a power set (2^Θ) called the frame of discernment [119]. From the power set, subsets representing the various types of possibilities are formed and assigned a probability between 0 and 1, called the mass value (m) as expressed in equation (4.1).

$$m: 2^\Theta \rightarrow [0,1] \quad (4.1)$$

Where equation (4.1) is also called the mass function or basic probability assignment for:

- The mass of an empty set, i.e. $m(\emptyset) = 0$
- The mass of members of the power set equal to 1, i.e. $\sum_{A \subseteq \Theta} m(A) = 1$.

With the probability assignment on the power set, conclusions are drawn on the outcome based on the trade-off between the *degree of belief (bel)* and the *degree of plausibility (pl)* of a given subset. For example, a proposition A, with non-empty sets B has the following degree of belief and plausibility as expressed in equation (4.2a and b).

$$bel(A) = \sum_{B \subseteq A} m(B) \quad (4.2a)$$

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B) \quad (4.2b)$$

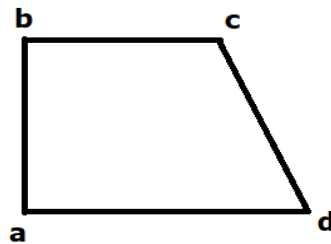
The D-S evidence theory has two significant limitations [119], [120]:

- i. An exponential increase (2^Θ) in computational complexity with an increasing number of elements in the frame of discernment. The number of elements is directly proportional to number of sensor data input.

- ii. Management of conflicting sensor input data, which can be caused by defects in sensor measurement. This leads to counterintuitive results and classification of data.

The fuzzy logic rule system can also be used in sensor data fusion problems involving uncertainties and imprecise data, but it has been reported to perform better at solving the conflicting sensor input data problem through data similarity measurement [121], [122]. However, it shares a similar computational complexity problem as the D-S evidence theory. But the computational complexity can be reduced using techniques such as rule interpolation, singular value based decomposition, etc. [123]. For these reasons, the fuzzy logic rule system is used as the sensor data fusion technique to classify events from the temperature/humidity and gas sensors.

Each sensor variable had three membership functions (MFs): normal, high (low for RH) and critical, and are all trapezoidal as expressed in equation (4.3).

$$\mu = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c < x \leq d \\ 0 & x > d \end{cases} \quad (4.3)$$


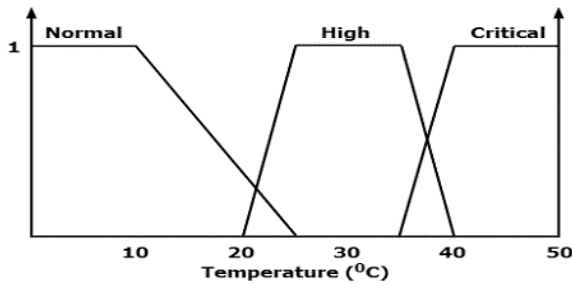
The membership functions and fuzzy sets are expressed as follows:

- The DHT11 temperature/humidity sensor is used for temperature/humidity measurement and has a temperature range of 20 – 50°C and a relative humidity (RH) range of 20 – 90% [124]. Therefore, the temperature MFs had fuzzy sets: ‘Normal’ – (0, 0, 10, 25), ‘High’ – (20, 25, 35, 40) and ‘Critical’ – (35, 40, 50, 50); while the RH MFs had fuzzy sets: ‘Normal’ – (35, 45, 60, 60), ‘Low’ – (23, 25, 35, 40) and ‘Critical’ – (20, 20, 23, 25). These fuzzy sets are selected based on recommended temperature/humidity ranges for industrial storage operational environments such as warehouses [125], [126].
- The gas concentration sensing range is 100 – 10,000ppm for the gas sensors [127], with considerable fire conditions having a value of 200ppm and above [128]. Hence, both the CO gas and smoke MFs had fuzzy sets given as ‘Normal’ – (0, 0, 200, 300), ‘High’ – (200, 300, 400, 500) and ‘Critical’ – (400, 500, 10000, 10000).

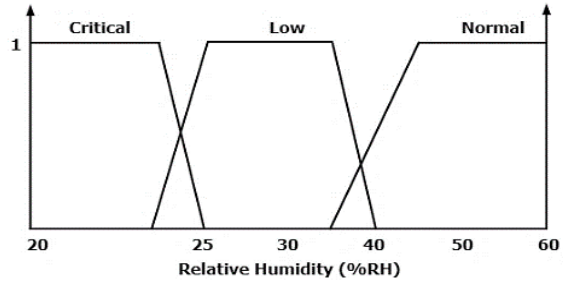
All fuzzy sets are outlined in Table 4.1 and the fuzzy sets diagrams for the Temperature/relative humidity, CO Gas, Smoke sensors are shown in Fig. 4.5(a) – (d) respectively.

Table 4.1. Fuzzy sets for temperature, humidity and gas sensors

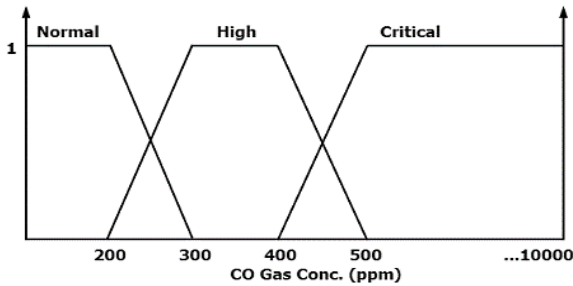
Membership Functions	Sensor Variables/ Fuzzy Sets			
	Temp. ($^{\circ}\text{C}$)	RH (%)	CO Gas Conc. (ppm)	Smoke Conc. (ppm)
Normal	(0,0,10,25)	(35,45,60,60)	(0,0,200,300)	(0,0,200,300)
High (Low for RH)	(20,25,35,40)	(23,25,35,40)	(200,300,400,500)	(200,300,400,500)
Critical	(35,40,50,50)	(20,20,23,25)	(400,500,10000,10000)	(400,500,10000,10000)



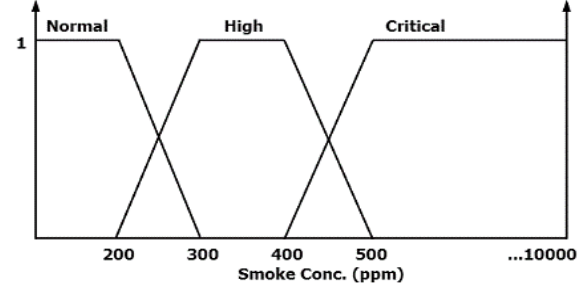
(a) Temperature



(b) Relative humidity



(c) CO gas concentration (conc.)



(d) Smoke concentration (conc.)

Fig. 4.5. Fuzzy set diagrams

The fuzzy inference system involves three processes, namely: fuzzification, fuzzy rules, and inference and defuzzification. The sensor input data are mapped to the fuzzy sets of the corresponding sensors' MFs during the fuzzification process to obtain the fuzzy rules. To reduce the number of rules (and thereby the computational complexity), rules that did not contribute to the overall output of the system are discarded. This truncation of rules follows after the Singular Value Decomposition (SVD) fuzzy rule-based reduction technique developed by Yam et al. [129]. Also, the similarity and proportionality-based relations between temperature and relative humidity reduced the number of rules and the conflicts in input sensor data. For example, an increase in temperature can result in a corresponding decrease in relative humidity and vice versa [130]. Hence, the following fuzzy rules and their inferences are obtained:

Critical Alert Events

1. If temperature is critical AND humidity is critical, THEN event alert is critical.
2. If CO conc. is critical AND smoke conc. is critical, THEN event alert is critical.

High Alert Events

1. If temperature is critical OR humidity is critical, THEN event alert is high.
2. If temperature is high AND humidity is high, THEN event alert is high.
3. If CO conc. is critical OR smoke conc. is critical, THEN event alert is high.
4. If CO conc. is high AND smoke conc. is high, THEN event alert is high.

Normal Alert Events

7. If temperature is normal AND humidity is normal AND CO conc. is normal AND smoke conc. is normal, THEN event alert is normal.

As for the digital sensors, i.e., the flame and motion sensors, their inputs are combined with the output of the fuzzy logic rule system to obtain the overall event characterization of all the sensor data (both analog and digital). Table 4.2 shows how the digital sensors' inputs are combined with fuzzy rule system output.

Table 4.2. Digital sensors' input combination with fuzzy rules output

Fuzzy Rules Output	Flame Sensor	Motion Sensor	Event Classification
Critical	High	High	Critical Alert
Critical	High	Low	Critical Alert
High	Low	High	High Alert
Low	Low	Low	Normal Alert

4.2.3 SSN Operation Pseudocode

The SSN operation involves sensing, data processing and communication with MRSN O-SSNs. The processed data communicated by each SSN is in the format of an integer array. The integer array contains the SSN node number, temperature, RH, CO gas concentration (conc.), smoke concentration (conc.) values and the characterized event (0 – 'High Alert' or 1 – 'Critical Alert,' etc.) respectively. The array has markers to indicate the beginning and end of the data being sent, and it is expressed as follows *<node number, temperature, RH, CO gas concentration, smoke concentration, event alert>*. For example, a data sent from SSN1 is encoded as *<3, 25, 35, 10, 20, 0>* where '<' and '>' are the beginning and end markers respectively, '3' is the SSN node number, temperature is 25°C, humidity is 35%, CO gas conc. 10ppm, smoke conc. 20ppm and event criticality is '0' (High Alert). This encoding format is also used by the O-SSNs of the MRSNs and G-MRSN to communicate their event characterization data to the BS. The SSN operation pseudocode is presented below.

Inputs:	Temperature and Humidity sensor data, odor and smoke sensors' data, motion and flame sensor data.
Outputs:	Critical alert events (fire and intruder detected) and high alert events (fire or intruder detected)

define sensors' input ports, indicator (red and green) output ports, LCD ports.

initialize SSN node number, list of event alert types, fuzzy sets and sample time

setup serial communication port and baud rate

setup LCD to display default case ("NORMAL")

setup the LEDs output to default case (Green: **on**, Red: **off**)

setup fuzzy input, output and list of rules

while loop (infinite)

if (time interval *is true*)

get all sensor's data

display sensors' data on the LCD screen

use temperature, humidity, odor and smoke data to evaluate fuzzy rule system

Evaluate fuzzy rules output

if ((event alert is critical & flame sensor is **HIGH** & motion sensor is **HIGH**) OR (event alert is critical & (flame sensor is **HIGH** OR motion sensor is **HIGH**))

display "CRITICAL" on LCD

 LEDs: Green: **off**, Red: **on**

send node number, temperature, humidity, odor, smoke data and event alert to MRSN

else if (event alert is high OR flame sensor **HIGH** OR motion sensor **HIGH**)

display "HIGH" event on LCD

 LEDs: Green: **off**, Red: **on**

send node number, temperature, humidity, odor, smoke data and event alert to MRSN

else (event alert is **NORMAL**)

display "NORMAL" on LCD

 LEDs: Green: **on**, Red: **off**

send node number, temperature, humidity, odor, smoke data and event alert to MRSN

end if

end while loop

4.3 MRS (MRSNs and G-MRSN) Architecture

The Multirobot system (MRS) architecture involves the MRSNs and G-MRSN. The structure and operational processes of the robots are presented. Also, the coordination algorithm between the MRSNs and G-MRSN enabling the G-MRSN to navigate to SSN locations based on MRSNs' request is elaborated. Finally, this subsection introduces the navigation algorithm enabling the robots navigating to the SSNs' locations and also the other preassigned paths within the zones and the operational task environment.

4.3.1 MRS Structure

The MRS structure is constituted of the robots' O-SSNs and navigation sensors with processors for decision making and motion control. The robots' O-SSNs and the zone SSNs have the same sensors and indicators except for the absence of the LCD in the MRSNs' O-SSNs. Hence, the sensing, event characterization and communication data encoding are the same as discussed in section 4.2. Other unique details of each robot's O-SSN are presented in this subsection.

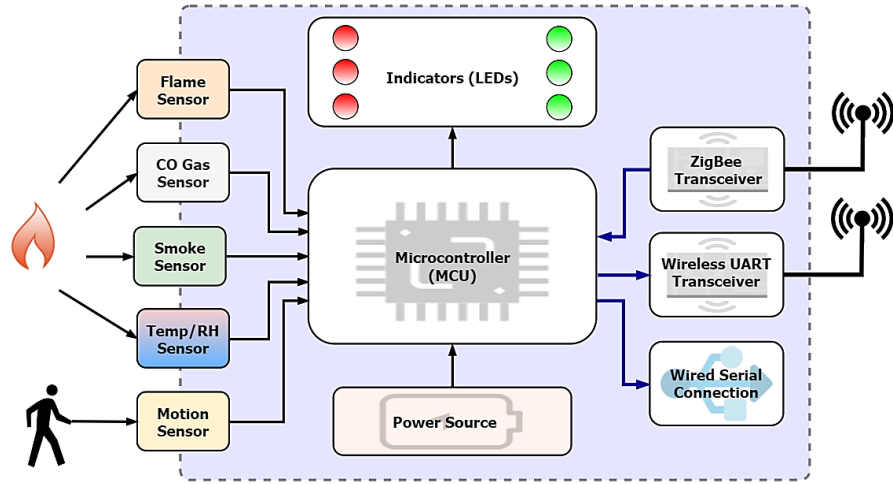
The navigation sensors for each robot includes an ultrasonic sound sensor for range measurement and optical encoders attached to the wheels for distance measurement. Only the optical encoders were used in navigation for distance (coordinate tracking) since all paths navigated were generated from previously known maps and locations.

A. MRSN Structure and Operation

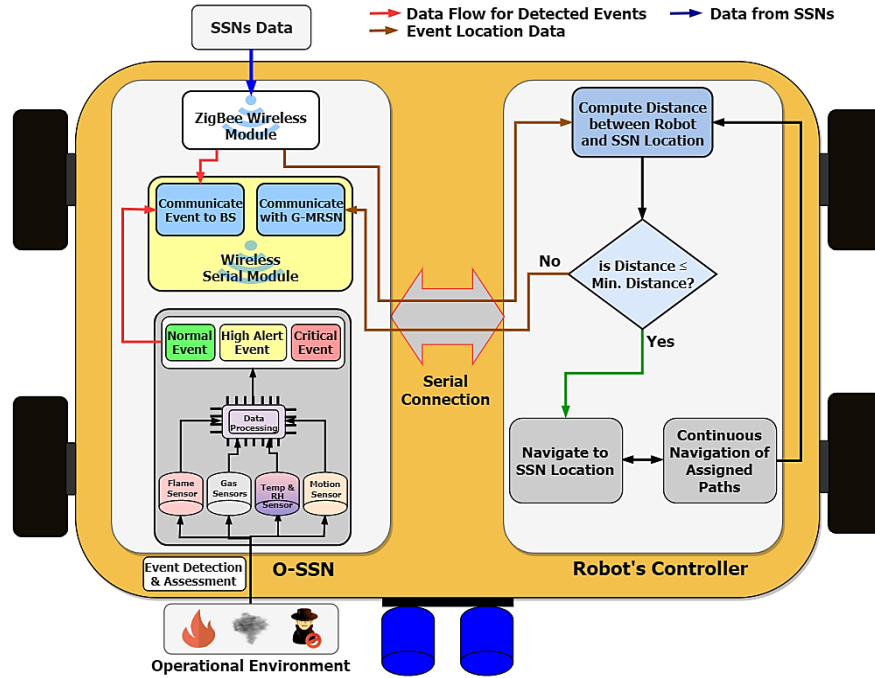
The MRSN's O-SSNs in addition to its sensors contains three communication protocols:

- i. The ZigBee protocol for communication with the SSNs.
- ii. The RF serial Wireless protocol for communication with G-MRSN and the BS.
- iii. The wired serial protocol for SSN location data transfers into the robots' motion control.

The MRSN's O-SSN structure is shown in Fig. 4.6(a). The MRSN operational process starts from its O-SSN, which senses, processes (by the event characterization system in section 4.2.2) and communicate (via the Wireless Serial module) the characterized events to the BS, while continuously navigating assigned paths within and around its zone. But when an event data is received by the O-SSN via the ZigBee protocol from an SSN, the event data is immediately communicated to the BS via the Wireless Serial protocol while the SSN location is sent into the robot control via the wired serial protocol, where the distance to the SSN location is computed. If this distance is less than the reference navigation distance, the MRSN navigates to the SSN location. If not, the MRSN requests the G-MRSN to navigate to the SSN location. The operational process diagram is shown in Fig. 4.6(b).



(a) MRSN O-SSN structure



(b) MRSN process diagram

Fig. 4.6. MRSN architecture

The MRSN O-SSNs has the same data encoding for communication to the BS as the SSNs. Hence, their pseudocode is similar for the event detection and communication part, except sections involving the LCD, since the MRSN lack this component. In addition to the event detection and communication sections, there is the navigation section to zone SSNs or communication with G-MRSN as the case may require. The MRSN pseudocode is presented below.

Inputs:	Temperature and Humidity sensor data, odor and smoke sensors' data, and flame and motion sensors data.
Outputs:	Critical alert and high alert events

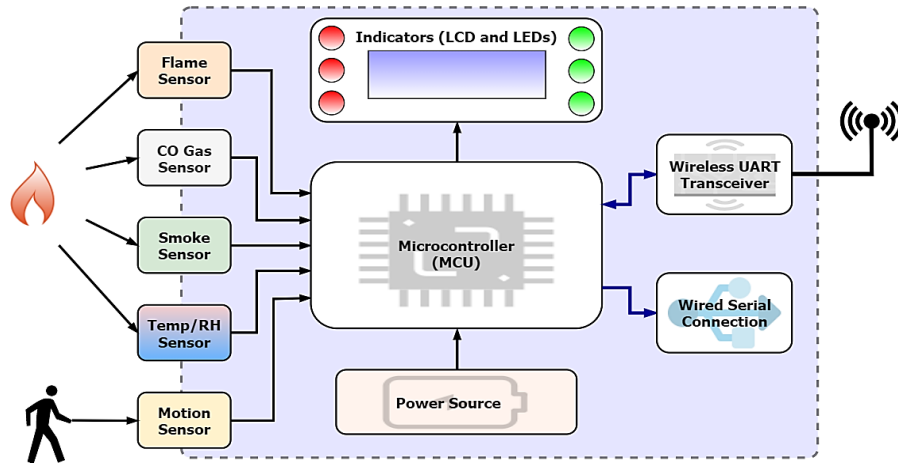
define sensors' input ports, indicator (red and green) output ports.
initialize MRSN node number, list of event alert types, fuzzy sets and sample time
setup serial communication port and baud rate
setup the indicator output (green LEDs on)
setup fuzzy input, output and list of rules

while loop (infinite)
 if (time interval *is true*)
 get all sensor's data
 use temperature, humidity, odor and smoke data to evaluate fuzzy rule system.
 Evaluate fuzzy rules output
 if ((event alert is critical & flame sensor is **HIGH** & motion sensor is **HIGH**) OR (event alert is critical & (flame sensor is **HIGH** OR motion sensor is **HIGH**))
 LEDs: Green: **off**, Red: **on**
 Send node number, temperature, humidity, odor, smoke data and event alert to BS
 else if (event alert is high OR flame sensor **HIGH** OR motion sensor is **HIGH**)
 LEDs: Green: **off**, Red: **on**
 Send node number, temperature, humidity, odor, smoke data and event alert to BS
 else (event alert is **NORMAL**)
 LEDs: Green: **on**, Red: **off**
 Send node number, temperature, humidity, odor, smoke data and event alert to MRSN
 end if
 poll data from zone related SSNs
 send SSN data to BS
 if (event alert)
 send SSN location and event alert into robot control
 compute distance to SSN
 if (distance \leq min. distance)
 Generate path to SSN location
 MRSN navigates generated path
 else (distance \geq min. distance)
 MRSN communicate with G-MRSN
 else (no event alert)
 MRSN navigates preassigned paths within/around the zone
 end while loop

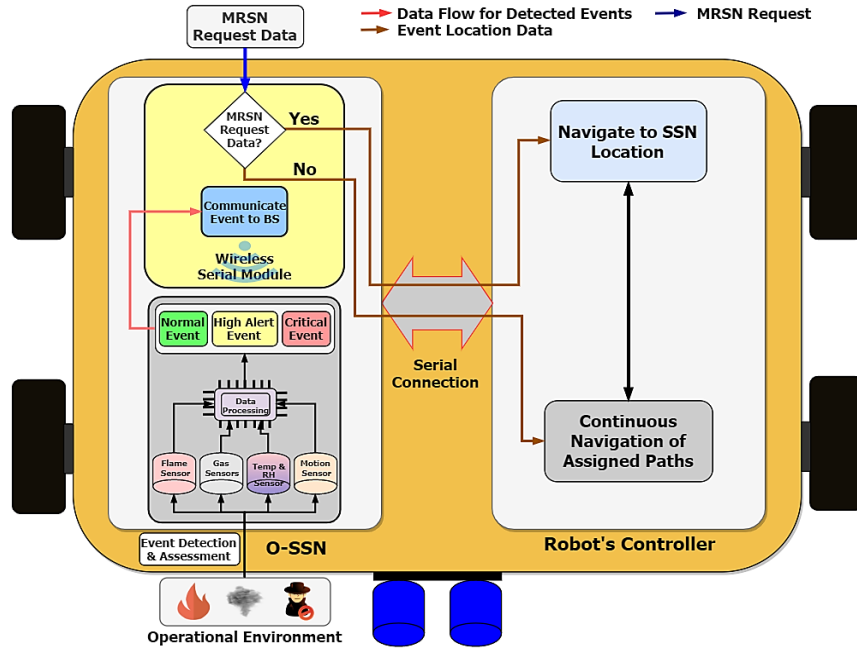
A. G-MRSN Structure and Operation

The G-MRSN's O-SSN in addition to its sensors has the RF serial Wireless protocol for communication with MRSNs and the BS, and the wired serial protocol for SSN location data transfer to robot's motion control. The O-SSN structure is shown in Fig. 4.7(a). The main

operational task of the G-MRSN is to navigate given paths around the task environment that are not covered by any MRSN with its zone, detect events and select the relevant actions, and relay detected events to the BS. While performing this task, the G-MRSN can navigate to SSNs within a zone based on a request made by any of the MRSNs. The G-MRSN operational flow chart is shown in Fig. 4.7(b).



(a) G-MRSN's O-SSN structure



(b) G-MRSN process diagram
Fig. 4.7. G-MRSN architecture

The G-MRSN and MRSNs have the same operation pseudocode except for the LCD and navigation instances where the G-MRSN navigates its preassigned paths until a request is received

from the MRSNs to navigate to a zone SSN location. Upon receiving this request, a path is generated whereby the G-MRSN navigates to the zone SSN.

4.3.2 MRSN/G-MRSN Coordination Algorithm

Each MRSN can navigate an assigned path within/around its zone. When an SSN within the same zone issues an alert, the MRSN receives it and calculates its distance to the alerting SSN. If the distance is less than the reference navigation distance, the MRSN navigates to the alerting SSN. Otherwise, the MRSN announces the task to all the G-MRSNs and select the suitable one to navigate to the alerting SSN location. The coordination approach between the MRSN and the selection of a suitable G-MRSN is based on the Contract Net Protocol (CNP) proposed by Reid G. Smith in 1980 [131]. In simplified terms, it is a market-based task distribution algorithm in multi-agent systems made up of a central task auctioneer/coordinator and multiple task bidders with different capabilities [132], [67]. The algorithm is summarized in algorithm one and illustrated in Fig. 4.8:

Algorithm 1: Contract Net Protocol (CNP)

1. Task Announcement: the coordinator/auctioneer
 - a. Receives a task(s) request(s).
 - b. Announces a set of tasks with specific requirements (objective function(s)) such as distance to task, speed of task execution, etc.
 - c. Only the auctioneer can announce tasks.
 2. Task Bidding: each agent
 - a. Analyses the task requirements based on its capabilities and functions.
 - b. Calculates its utilities with respect to the task requirements/objectives.
 - c. The calculated utilities are communicated as a bid to the auctioneer.
 3. Best Bidder Selection: the coordinator/auctioneer
 - a. Evaluates all agent submitted bids with respect to the task requirement.
 - b. The agent with best bid is selected.
 4. Task Allocation: the selected bidder is contracted and assigned the task to execute the task.
-

For this system, the MRSN announcing the task becomes the coordinator/auctioneer while the available G-MRSNs become the bidders. The task announcement includes the alerting SSN location and the event alert. The MRSN broadcasts the task to the G-MRSNs, and each G-MRSN calculates its navigation distance to the alerting SSN. The calculated navigation distances are communicated to the MRSN by the G-MRSNs, and the G-MRSN closest to the alerting SSN is selected to navigate the distance to the alerting SSN location. If no G-MRSN is within the reference navigation distance, the MRSN then navigates itself to the alerting SSN. The reference navigation

distance is set at 7m, corresponding to the effective detection range of the PIR motion sensor. The coordination approach is illustrated in Fig. 4.9.

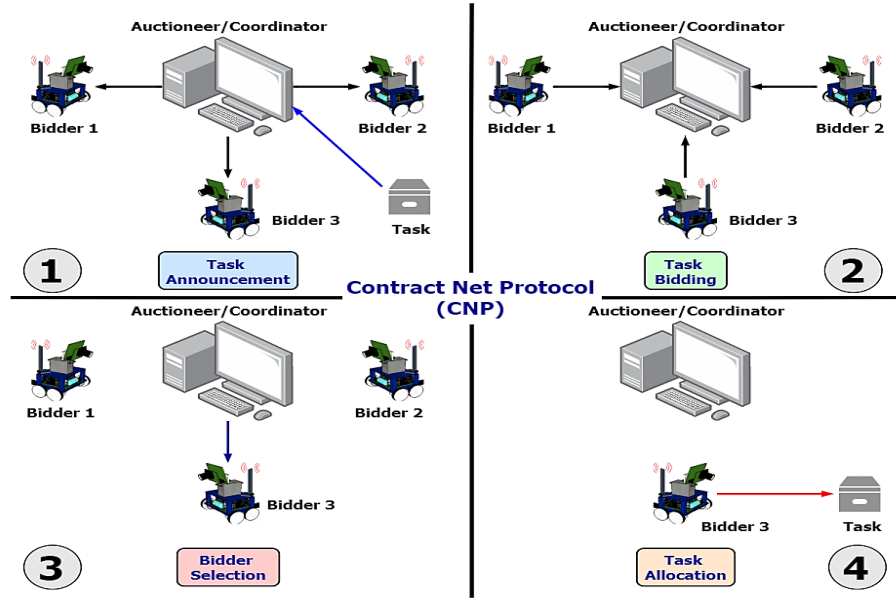


Fig. 4.8. Contract Net Protocol (CNP)

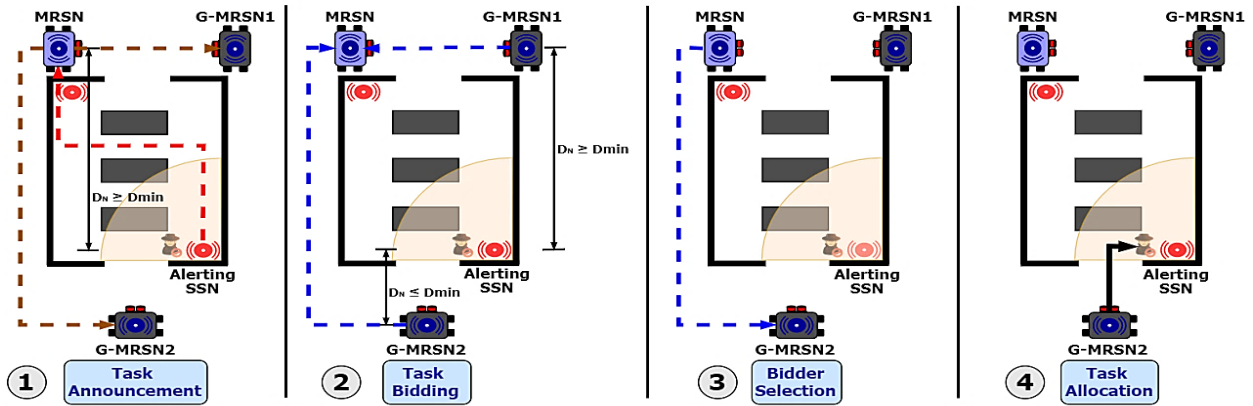


Fig. 4.9. ANR system CNP

4.3.3 MRSNs and G-MRSN Navigation Algorithm

Once a robot is selected to navigate to the alerting SSN location (either MRSN or G-MRSN), there is a need to plan a path from its current location to the alerting SSN location. A* path planning is deployed for this purpose to generate the required path using the map of the operational environment. This algorithm is chosen because of its simplicity and heuristics-based approach to finding the least costly navigation path [133].

A. The A* Path Planning Algorithm and Environment Mapping

The algorithm operates on a 2D grid-based mapped environment, containing cells that are either free spaces or occupied by objects/obstacles based on the given known map. Hence, the shortest path to the goal is determined by evaluating the propagated cost from the starting cell to the goal cell. The cost function is expressed in equation 4.4 [134]:

$$F = G + H \quad (4.4)$$

- G is the cost of moving from the starting cell to the next possible adjacent cell in the grid. The adjacent cells' horizontal or vertical movements are assigned the same cost say 'a' values while diagonal movements have cost equal to the square root of 2 times the of horizontal or vertical movements cost i.e. $G_{h/v} = a$ and $G_d = \sqrt{2}a$.
- H is the heuristics which provides information on the cost of reaching the goal cell from the current cell on the path. Heuristics functions usually include distance to target based on horizontal and vertical movements (Manhattan distance), diagonal distance or Euclidean distance [135], [136].

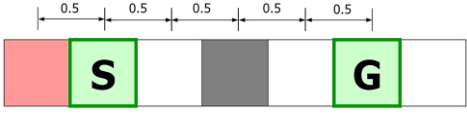
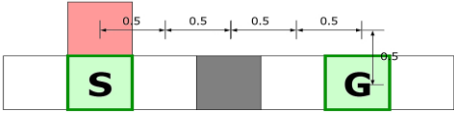
For the A* path planning algorithm a 2D occupancy grid map is used with equal cell sizes. Considering a large operational environment such as warehouses:

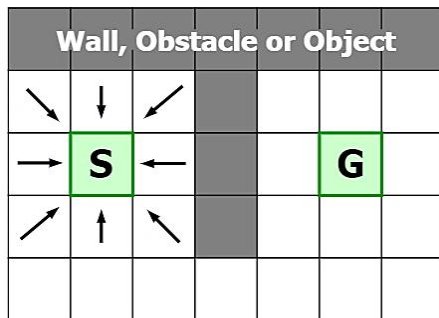
1. The cell size was chosen to be **0.5×0.5m** in association with the dimension of the used robot, which is equal to **0.45×0.35m**.
2. The zones' operational environment is considered to be **20×15m** in dimension, resulting in **40×30 cells** 2D occupancy grid map for MRSNs and G-MRSN navigation simulation.

The following steps use an example to illustrate the A* algorithm,

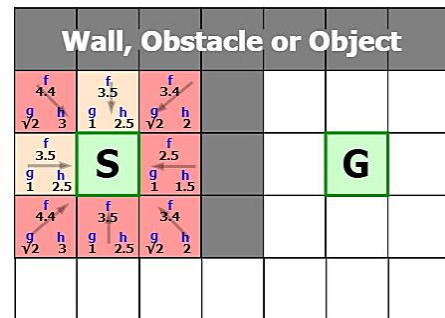
1. Consider the current position of the robot as the start cell (S), while the alerting SSN location is the goal cell (G). Around the start cell, there are eight possible movements to the next cell on the path as shown in 4.10 (a).
2. A free cell (empty space) is assigned a movement cost of $G = 1$, while a cell with obstacles (walls, storage racks, objects) has a movement cost of $G = 100$. Hence, for each of the adjacent cells, moving vertically or horizontally from the start cell S has a cost of $G = 1$, while a diagonal movement has a cost of $G = \sqrt{2}$.
3. The heuristics (H) to move from any of the adjacent cells to the goal cell involves adding the vertical/horizontal cell distances from the center of the adjacent cell to the center of the

goal cell including the obstacle cells. The cell distance is equal to 0.5m. Considering the cell to the left of S and the cell above S, the heuristics (H) from the mentioned cell to the goal cell (G) is calculated as:

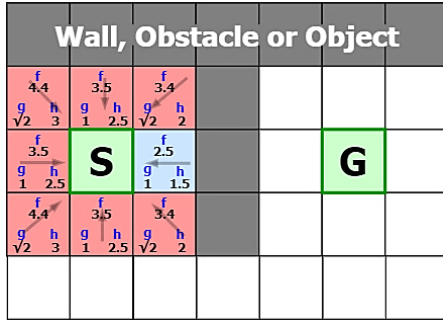
- a. Left hand cell: , $H = 5 \times 0.5 = 2.5m$
- b. Cell above S: , $H = 5 \times 0.5 = 2.5m$.
- a. The total cost F (G + H) for the selected cells is therefore:
- Left hand cell: $G = 1$ and $H = 2.5 \rightarrow F = G + H = 1 + 2.5 = 3.5m$
 - Cell above S: $G = 1$ and $H = 2.5 \rightarrow F = G + H = 1 + 2.5 = 3.5m$
5. Hence, the movement cost (G), heuristics cost (H), and total cost (F) for each of the adjacent cell around the start cell is shown Fig. 4.10(b).
6. From the cost evaluation of all the adjacent cells, the adjacent cell with the least cost of F is selected and considered for the next move from the start cell, as shown in Fig. 4.10(c). The costs of the adjacent cells to the cell under consideration (CU) are also evaluated using steps 2 – 4, and the cell with the least cost of F is selected for the next possible move as shown in Fig. 4.10(d).
7. This process is repeated as shown in Fig. 4.10(e) – (f), until all the cells that can be part of the path are considered as shown in Fig. 4.10(g).
8. The shortest navigation path is the least cost path traced from the goal cell backward to the start cell using the F cost for each cell under consideration as shown in Fig 4.10(g). This becomes the navigation path for the robot as shown in Fig. 4.10(f).



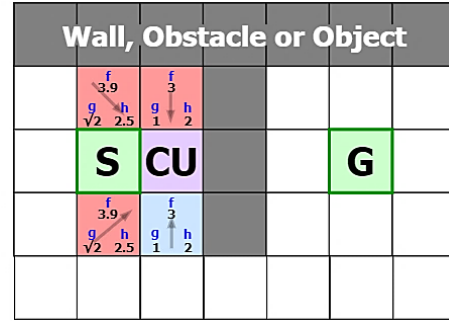
(a) Grid Initialization



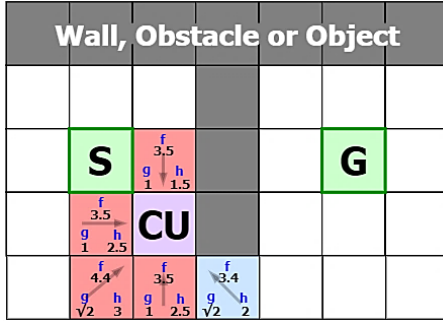
(b) Cost evaluation around start cell (S)



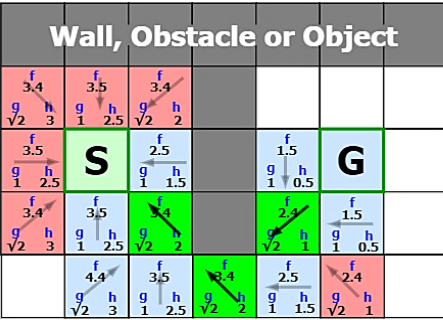
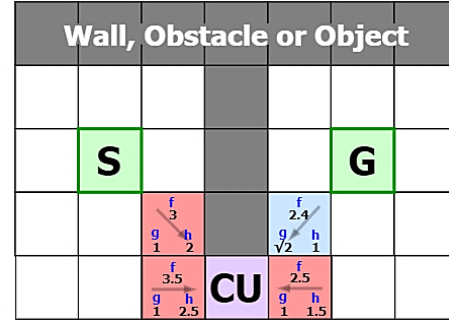
(c) Selection of Cell with least cost



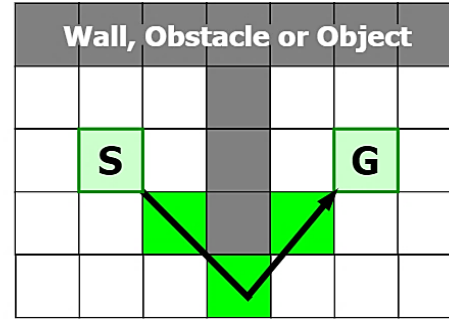
(d) Adjacent cells cost evaluation/new cell selection



(e) and (f). Adjacent cells cost evaluation/new cell selection



(g) Least Cost Path Evaluation



(h) Final Navigation Path

Fig. 4.10. A* Algorithm Example

An example representation of the zone environment mapping and the robots' A* planning in LabVIEW is shown in Fig. 4.11. The simulated SSNs (yellow shaded areas on the figures) have coordinates as shown in Table 4.3. The example shows an A* star planned path from the location (5, 4) to the alerting SSN1 (10, 16).

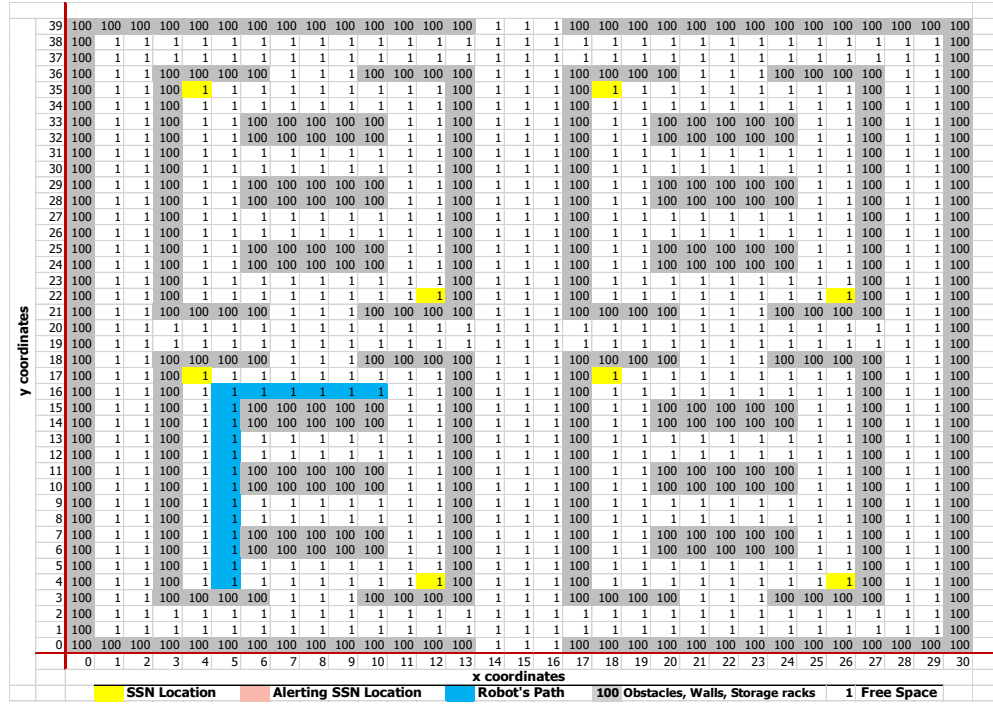


Fig. 4.11. Simulation Occupancy Grid Map

Table 4.3. SSN coordinates

SSN	LabVIEW Environment Map Coordinates	Actual Coordinates
SSN1	(4, 17)	(2.0, 8.5)
SSN2	(4, 35)	(2.0, 17.5)
SSN3	(12, 4)	(6.0, 2.0)
SSN4	(12, 22)	(6.0, 11.0)
SSN5	(18, 17)	(9.0, 8.5)
SSN6	(18, 35)	(9.0, 17.5)
SSN7	(26, 4)	(13.0, 2.0)
SSN8	(26, 22)	(13.0, 11.0)

4.4 ANR IoT Base Station Architecture

The base station contains the following components:

- The BS Gateway (BSG) to receive data from the O-SSNs of the MRSNs and G-MRSN.
- The base station GUIs for:
 - Sensor data aggregation, visualization, and local storage.
 - The visualization of the navigation environment, the generated paths and the robots navigating their paths.
- The local storage and two cloud platforms for cloud data aggregation, visualization, storage and analytics.

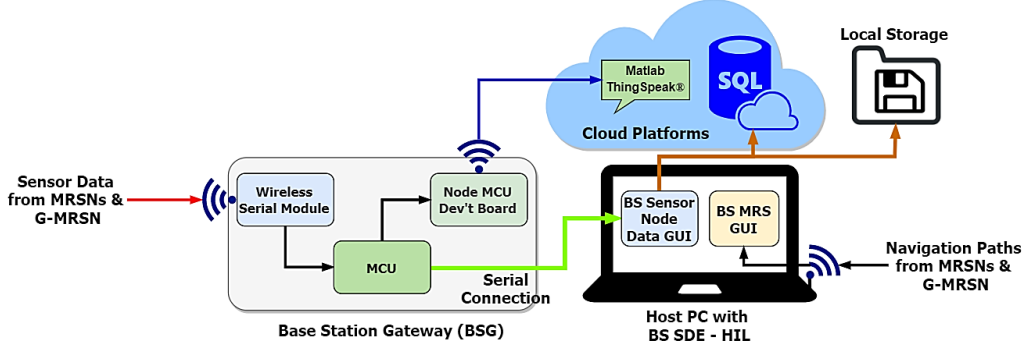


Fig. 4.12. BS architecture

4.4.1 Base Station Gateway (BSG):

The BSG is made up of the following components:

1. The RF Wireless Serial communication module which receives sensor node (SSNs and O-SSNs) data from the MRSNs and G-MRSNs,
2. The MCU that processes the sensor node data by parsing it according to the data encoding format, i.e. *<node number, temperature, RH, CO gas concentration, smoke concentration, event alert>*. The parsed data is then sent to Node MCU for cloud upload and serially to the BS sensor node data GUI.
3. The Node MCU development board for direct Wi-Fi connection and sensor node data upload to the ThingSpeak® cloud platform.

4.4.2 The BS GUIs

The BS GUIs constitute two separate GUI programs, namely the BS Sensor Node Data (SND) GUI for sensor data visualization and BS MRS GUI for visualizing MRS environment, paths, and navigation.

A. BS Sensor Node Data (SND) GUI

The BS sensor node data GUI is an interactive GUI that shows the various events and data being detected at the SSNs, MRSNs, and G-MRSNs. The SND GUI has a tab bar menu with tabs covering SND GUI guide information, MRSNs, G-MRSN, and the SSNs within each zone as shown in Fig. 4.13. Once the BS SND GUI information tab is selected, the screen of the SND GUI is going to display the interface shown in Fig. 4.13 under the bar menu, and it has the following structural content:

1. To the left side of the interface, there is a brief description of the GUI and a list of its operational instructions on how to run the SND GUI program.

2. To the right side of the operational instructions are the BS SND GUI controls for:
 - i. Interactively operating the SND GUI program (start and stop).
 - ii. Interactively specifying the serial communication port that connects the SND GUI to the BSG.
 - iii. Interactively selecting the storage platform (cloud or local system). Depending on the selected storage platform, an MS-SQL connection string or a local text file path is chosen from the “Connect to the Database” control. The local file path is obtained from file direction after clicking the folder icon on the “local file path” control.
 - iv. Indicating which sensor node’s data is being received. LED indicators are positioned at the sensor node locations, as shown in Fig. 4.13 where they turn “green” (MRSN1) indicating that data has been received from the relevant sensor node. The received data can then be visualized on the related sensor node Tab.

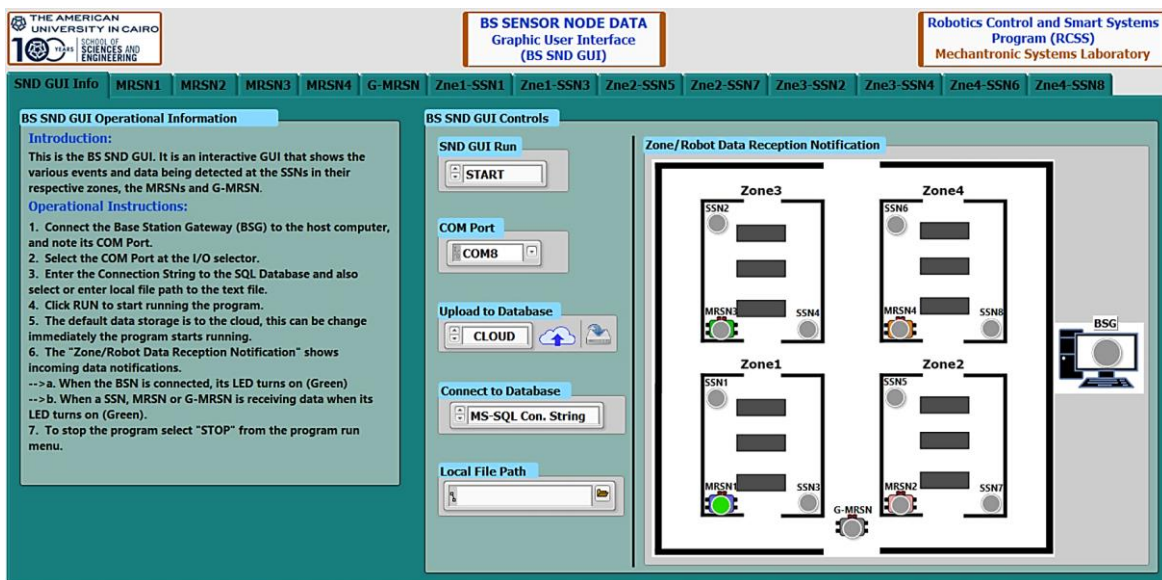


Fig. 4.13. BS SND GUI information tab

Each of the tabs for the SSNs, MRSNs and G-MRSNs has the following structural content as shown in the MRSN1 Tab in Fig. 4.14.

1. To the left of the interface, there is the MRSN1 surveillance data associated with indicators visualizing the temperature, humidity, odor (CO gas) and smoke data. There are also indicators for event alert characterization (critical – “red”, high alert – “yellow” and normal – “gray”).

- To the right of the interface are graph windows displaying the temperature, humidity, CO gas and smoke concentration values as plots with respect to time.

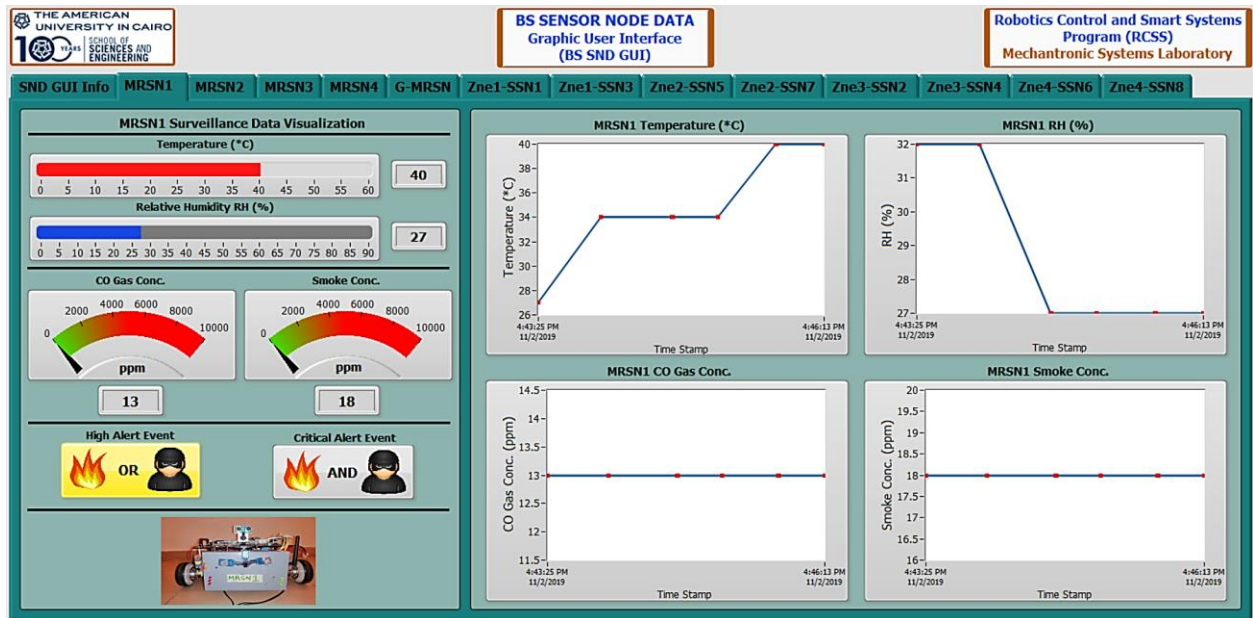


Fig. 4.14. MRSN1 tab interface

State machines are used to appropriately receive and display the data from the sensor nodes on their tab interfaces. Every sensor node has a state machine activated by an ID called the node number (NN). The NNs are integers corresponding to the MRSNs, SSNs, and G-MRSN respectively where 'NN' = 1 \rightarrow MRSN1, 'NN' = 2 \rightarrow MRSN2, 'NN' = 3 \rightarrow SSN1, ..., 'NN' = 10 \rightarrow SSN8, etc. The state machines operate as follows:

- When new sensor node data is received into the GUI, the state machine corresponding to the sensor node is activated based on its NN. This allows the received data to be visualized on the sensor node tab interface.
- The visualized data is then stored locally or uploaded to the cloud depending on the data storage selection mode
- If no data is received, the state machine stays on the default case which displays the SND GUI information Tab.

An example state machine operation for SSN1 (NN = 3) is shown in Fig. 4.15.

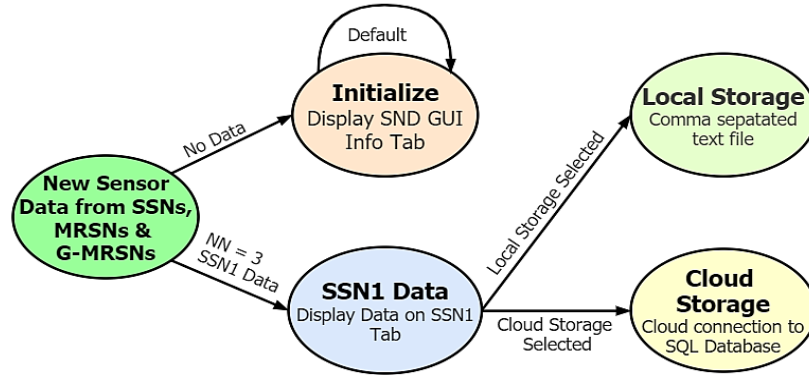


Fig. 4.15. SSN1(NN = 3) state machine operation.

From the SND GUI, the cloud connection string allows the sensor node data to be uploaded to the Microsoft (MS) cloud platform, where web applications for SSNs, MRSNs and G-MRSN are developed to visualize and analyze the data. In case the local storage is selected, the data is accumulated in a text file that can be viewed on any data processing software such as MS Excel.

B. BS MRS GUI

The BS MRS GUI program is designed to display the simulation environment, the generated paths, and the robots' navigation and location in the environment. This MRS GUI has a tab bar menu with tabs covering the MRS GUI information guide tab, and a tab for each MRSN and G-MRSN. Once the MRS GUI information guide tab is selected, the screen shown in Fig. 4.16 is going to be displayed, having the following structural contents:

1. To the left side of the interface, there is a brief description of the GUI and a list of operational instructional guidelines on how to run the MRS GUI.
2. To the right side of the interface are the MRS GUI controls for:
 - i. Interactively operating the GUI program and simulation service (start/stop).
 - ii. Interactively specifying the serial communication port through which an alerting SSN's location data is transferred from the O-SSN into the robot motion control.
 - iii. Displaying (on numerical indicators) the x, y coordinates, and the event alert data from the alerting SSN being sent from O-SSN.
 - iv. Indicating the alerting SSN. LED indicators are positioned at the SSN locations as shown in Fig. 4.16 where they turn "yellow" (SSN1) to indicate high alert events, "red" for critical alert events and "gray" for normal (or no) events detected.

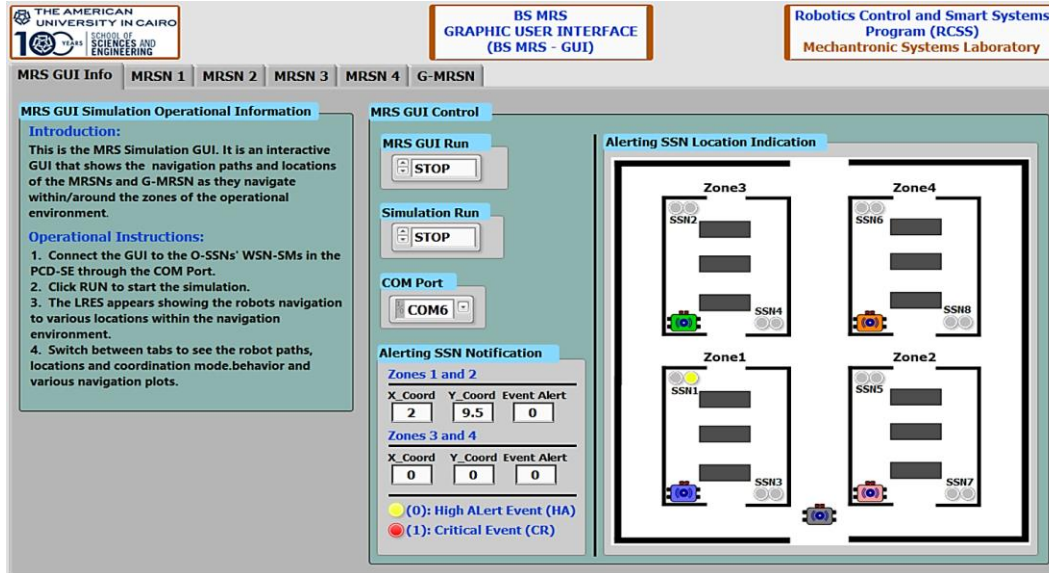


Fig. 4.16. MRS GUI information tab

Each of the tabs for the MRSNs and G-MRSN has the following structural content as shown in the MRSN1 Tab in Fig. 4.17.

1. To the left side of the interface, there are numeric controls to interactively enter the initial position of the robots and numeric indicators to show the goal and current positions of the robot.
2. To the right side of the interface are graph windows displaying the results of the A* algorithm, both as an intensity map and as an XY coordinate graph plot.
3. At the bottom part of the interface, are the coordination mode boolean indicators and status textbox showing the coordination mode of the robot. During operation,
 - i. If the distance to the alerting SSN is less than the reference navigation distance, the “True” LED indicator turns green and the MRSN1 Status textbox indicator displays the message “MRSN1 navigating to alerting SSN location”.
 - ii. If the distance to the alerting SSN is greater than the reference navigation distance, the “False” LED indicator turns red, and the MRSN1 Status textbox indicator displays the message “MRSN1 coordinating with the G-MRSN”.
 - iii. Else if no alert is issued from the SSNs, the “Neutral” LED turns yellow, and the MRSN1 Status textbox indicator displays the message “MRSN1 navigating assigned paths around its zone”.

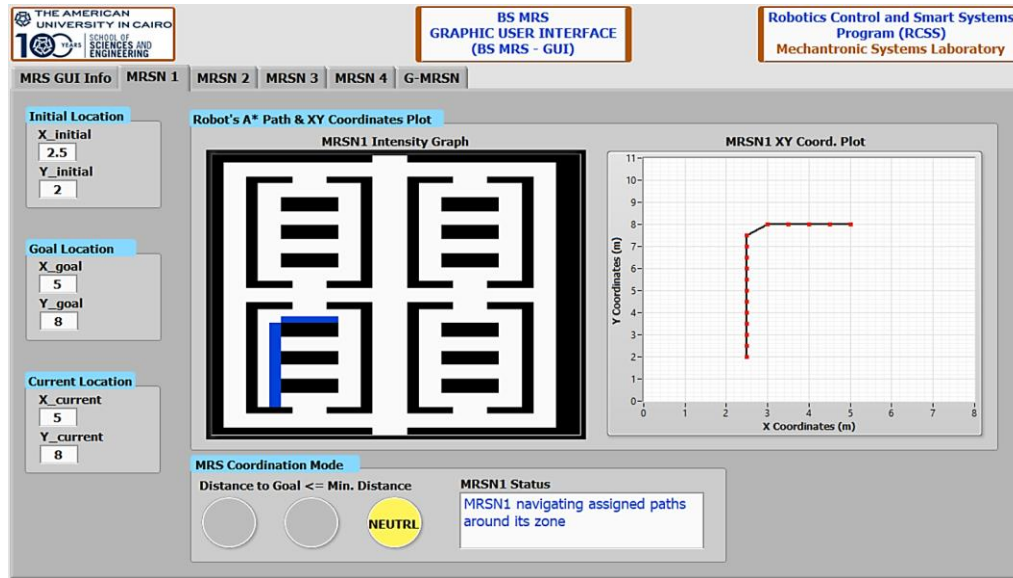


Fig. 4.17. MRSN1 tab interface.

4.4.3 Cloud Data Storage, Visualizations and Analysis

Sensor data sent from the MRSNs and G-MRSNs to the BS are either stored locally or in the cloud as necessary through two cloud platforms. Cloud data storage presents several advantages to the system despite the challenges associated with cloud technology. The benefits, challenges and suggested solutions to problems of cloud data storage are presented in Table 4.4 [137]–[139]:

Table 4.4. Advantages, Challenges and Proposed Solutions of Cloud Data Storage

S/N	Benefits	Explanation
1	Flexibility	With cloud storage, data can be accessed anytime and anywhere for any kind of analysis and processing.
2	Scalability	Application expansion that leads to increase data size to be stored can be easily accommodated in a cloud storage with less infrastructural change.
3	Agility	Cloud storage provides its users with the necessary toolkits to respond to fast-changing business environments.
4	Storage Capacity	Storage capacity in cloud systems is virtually unlimited.
5	Backup and Recovery	Cloud storage systems offer better backup and recovery options for data management.
6	App Integration, Deployment, and Upgrades	With cloud storage, applications in the cloud can be easily deployed and upgraded to accommodate the changes in the size of data being aggregated.
7	Cost Management	By avoiding the cost of operating hardware for storage purposes, cloud storage presents a cost-effective way of managing data.
	Challenges	
1	Security and Privacy	Security and privacy are significant challenges in cloud storage systems because data is mostly stored by third-party agents. There are possibilities of data compromise, attacks, data transfer to other parties, etc.

2	Cyber Attacks	Data stored in the cloud are susceptible to multiple kinds of attacks including data-stealing, malware injection, wrapping attacks, denial of service attack, etc.
3	Bandwidth Limitations	Cloud systems present on-demand, pay-as-you-go services, that are limited by bandwidth. An increase in demand also implies increased bandwidth, hence, requiring additional cost.
4	Technical Challenges	These are issues associated with the technicalities of the systems such as network latency and speed, internet connection, etc.
5	Data Management Flexibility	Depending on the provided cloud service, the users are mostly constrained to their style of data management. This makes migration from one platform to the next difficult.
6	Extended Use Costs	For small applications, cloud storage can be beneficial at first, but as the business expands, so does the quality cloud services increase in cost.
	Proposed Solutions	
1	Encryption before Upload.	Encryption before upload provides two-layer security to the data being accessed on the cloud [140].
2	Multiple Authentication	These ensure that a user who accesses the data is well known through the authentication process [141].
3	Hypervisor Security	The hypervisor is the layer that organizes, controls, monitors and maps physical resources to virtual resources in the cloud. Strengthening security at this layer increases the overall security of the system [142].
4	Cloud Storage Platform Interoperability	The interoperability of cloud storage platforms can reduce the problem data management flexibility, making migration easier from one platform to the next.

Generally, the storage of sensor nodes' data in cloud platforms facilitates the integration of IoT technologies such as remote data visualization, intelligent data analytics, application development for smart systems (such as logistics, transport), etc.

There exist multiple cloud platforms that can be used for data storage and IoT purposes. These platforms can be grouped based on their deployment models into private, public, hybrid cloud or community cloud platforms [143]–[145].

- a. **Private Cloud platforms** are provided by commercial companies and are described by dedicated hardware, no resource sharing with third-parties, enhanced security, etc. The cloud infrastructures can be remote or within the user premises with complete control by the user. Examples include Microsoft platforms (SQL servers, Azure, Web applications, etc.), Amazon Cloud Services (AWS), IBM cloud services, etc.
- b. **Public Cloud platforms** are provided by public companies and are described by free and opened source service development, resource sharing, reduced security, etc. The cloud infrastructures are remote and controlled by hosting companies that offer limited services. Examples include Google cloud services, Kaa, Device Hive, Zetta, etc. [146].

- c. **Hybrid Cloud platforms** combine aspects of private and public services depending on the service provider and the user's choice. Examples include the Microsoft platforms (SQL Servers, Azure, etc.) Google cloud services, Matlab ThingSpeak®, ThingsBoard, etc.
- d. **Community Cloud Platforms** are platforms for groups or organizations with common goals, whereby all resources and infrastructures are shared. Examples include Google Apps for Government, Microsoft Government Community Cloud, etc.

The hybrid cloud platforms are used for cloud data storage and IoT based technologies since they provide both public and private services. The public services can be used to test and validate the cloud storage and IoT technologies of the platform before a full migration to the private services can be completed. The following hybrid cloud platforms are used:

- a. **The MS Cloud Platform** where sensor node data is uploaded directly from the SND GUI.
- b. **The ThingSpeak Cloud Platform** where sensor node data is uploaded directly from the BSG.

Both cloud platforms are used for the following purposes:

- a. Data Visualization: the cloud platforms enable remote data visualization with multiple visualization formats and interfaces.
- b. Data Analytics: the cloud platforms enable remote and intelligent data analytics through the development of cloud applications.

A. The MS Cloud Storage Platform

The MS cloud storage platform is constituted using:

- a. SQL Server Management Studio (SSMS) and
- b. Web applications,

These can be deployed both offline and online.

a. MS SSMS

The MS SSMS is a software application used to create, manage and configure objects on an SQL server. These objects include SQL databases, database engines, backup devices, management objects, security objects and structures, etc. The software can be combined with web applications developed in MS Visual Studio® to visualize SQL databases, analyze their data, trigger actions based on the data, connect to other websites, etc. Fig. 4.18(a) shows the MS cloud storage platform. Due to its interoperability with the SND GUI, the MS SSMS:

- i. Can be used directly for offline sensor node data storage, visualization, and analytics using web applications. This offline public service facilitates the testing and validation of the platform.
- ii. Can be connected online to the MS Azure cloud, which provides enhanced services (web applications, analytics, APIs, etc.) at various prices.

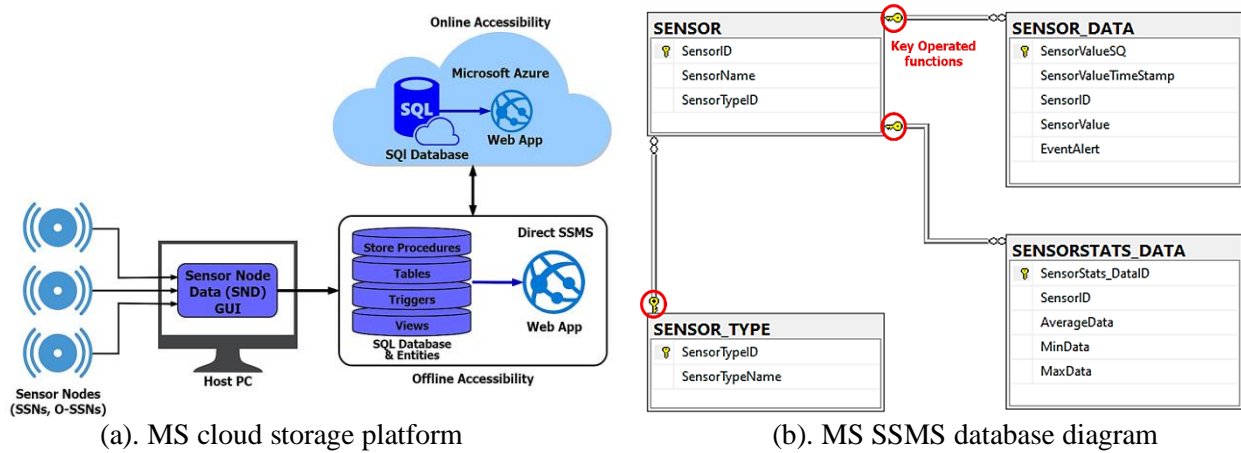


Fig. 4.18. MS cloud storage platform

Fig. 4.18(a) shows the SQL database with its entities i.e. store procedures, tables, triggers, and views. The store procedures entity is used to create the tables, while the table entity is used to store the sensor node data. The triggers entity acts on data in the tables for evaluation and analytics purposes. As for views entity, they are used to display the data contained in the tables. Four tables namely, “*SENSOR*” “*SENSOR_DATA*,” “*SENSOR_TYPE*” and “*SENSORSTATS_DATA*” are created as shown Fig. 4.18(b) using the stored procedures entity. During the creation of these tables, the “*SensorID*,” “*SensorName*,” “*SensorTypeID*” and “*SensorTypeName*” are written into “*SENSOR*” and “*SENSOR_TYPE*” tables as relevant. Data entries in the tables can be obtained through ‘key’ (encircled in Fig. 4.18(b) operated functions. These functions allow data to be read from one table to another table or from table to web application. Data storage and visualization of the “*SENSOR_DATA*,” table are performed as in the following steps:

- i. Each sensor node data is stored against the following labels as shown in the “*SENSOR_DATA*” table in Fig. 4.18(b)
 - a. the “*SensorValueSQ*” representing the sequence of data entry.
 - b. The timestamp of the data – “*SensorValueTimeStamp*.”

- c. The sensor ID – “*SensorID*,” i.e. the ID of each sensor (temperature, relative humidity, gas and smoke) within a sensor node accessed through ‘key’ operations.
 - d. The data itself – “*SensorValue*.”
 - e. Finally, the event alert – “*EventAlert*.”
- ii. Data visualization is performed by running query functions contained in the view entity that display the data in a table format from the queried table.

For example, a badge of data starting with SSN1 (with ID = 3) data <**3, 25, 35, 10, 20, 0**> and ending with SSN4 (with ID = 6) data <**6, 28, 20, 40, 30, 0**> can be viewed as shown in Table 4.5. This table is displayed by the command of a view entity function called “*GetSensorData*”. The function outputs a table with the “*SensorID*”, “*SensorName*”, “*SensorValueSQ*”, “*SensorValueTimeStamp*”, “*SensorValue*” and the relevant “*EventAlert*”.

Table 4.5. Example sensor node data visualization on MS SSMS

SensorID	SensorName	SensorValueSQ	SensorValueTimeStamp	SensorValue	EventAlert
26	SSN1-Temp	1	2019-09-11 09:30:03.220	25	
27	SSN1-RH	2	2019-09-11 09:30:04.980	35	
28	SSN1-Gas	3	2019-09-11 09:30:06.080	10	
29	SSN1-Smk	4	2019-09-11 09:30:07.890	20	
30	SSN1-EA	5	2019-09-11 09:30:09.009	0	High Alert
:	:	:	:	:	:
:	:	:	:	:	:
41	SSN4-Temp	16	2019-09-11 09:34:04.220	28	
42	SSN4-RH	17	2019-09-11 09:34:05.980	20	
43	SSN4-Gas	18	2019-09-11 09:34:06.980	40	
44	SSN4-Smk	19	2019-09-11 09:34:07.100	30	
45	SSN4-EA	20	2019-09-11 09:34:08.010	0	High Alert

Data analytics and visualization of the “*SENSORSTATS_DATA*” table are performed as in the following steps:

- i. Basic statistical analytics (mean, maximum and minimum data entries) are performed on the data and stored in the table against the following labels as shown in Fig. 18(b):
 1. “*SensorStats_DataID*” corresponding to the identification of statistical data.
 2. “*SensorID*” i.e. the ID of each sensor (temperature, relative humidity, gas, and smoke) within a sensor node accessed through ‘key’ operations.
 3. “*AverageData*” corresponding to the mean value of sensor data.
 4. “*MinData*” corresponding to the minimum data value of the sensor.
 5. “*MaxData*” corresponding to the maximum data value of the sensor.

- ii. Data visualization is performed by running query functions contained in the view entity that display the data in a table format from the queried table.

For example, Table 4.6 shows the visualization of data from the “*SENSORSTATS_DATA*” table for aggregated data of SSN1.

Table 4.6. example sensor node statistics data visualization on MS SSMS

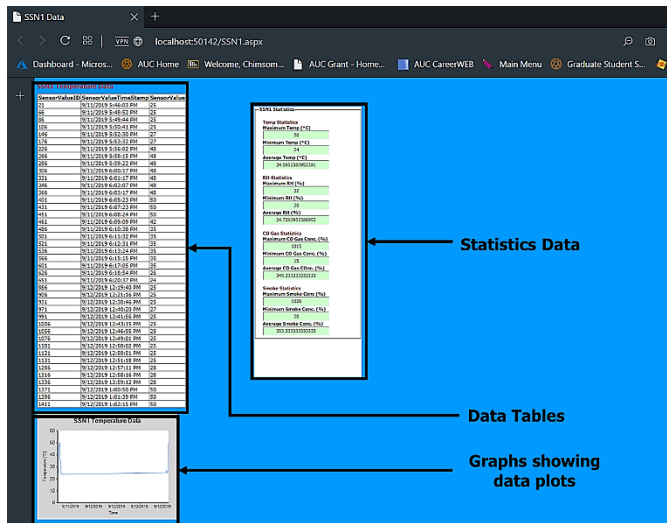
SensorStats_DataID	SensorID	AverageData	MinData	MaxData
11	11	37.68	24	50
12	12	24.92	20	32
13	13	415	15	1015
14	14	420	20	1020
15	15	0.4	0	1

b. Web Application

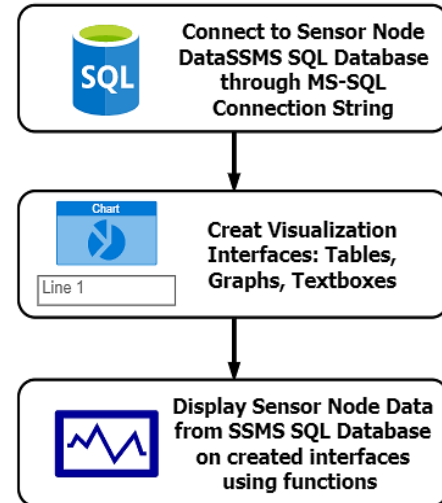
The web application visualization is designed using MS Visual Studio®, and it is made of tables, charts (graphs), textboxes to visualize the data from the SQL database. It uses functions to access data from the SQL database tables for each sensor node and display them in a table containing the “*SensorValueID*,” “*SensorValueTimeStamp*,” and “*SensorValue*.” The graphs show plots of the “*SensorValue*” data against the “*SensorValueTimeStamp*,” and finally, a group of textboxes displays the statistics, i.e. mean, maximum, minimum of the data. Fig. 4.19(a) shows example MRSN1 temperature table, plots, and statistics.

The web applications access and display data from the sensor node data SSMS SQL database through the following steps, as shown Fig. 4.19(b):

- i. The web application connects to the sensor node SSMS SQL database using the MS-SQL connection string.
- ii. Tables, charts (graphs) and other interfaces are created using web forms to display sensor node data and statistical analytics.
- iii. Functions are then developed that access the sensor node SSMS SQL database to read and display the data on the created interfaces.



(a) Web application interfaces



(b) Web app. visualization process

Fig. 4.19. Web application visualization interfaces

B. Matlab ThingSpeak Cloud Platform

Matlab's ThingSpeak is an online IoT-based cloud platform for data aggregation, visualization, and analytics. It is featured with the ease of interoperability with many development hardware and facilitates online analytics using applications developed through Matlab Toolboxes. Fig. 4.20 illustrates the Matlab ThingSpeak cloud integration.

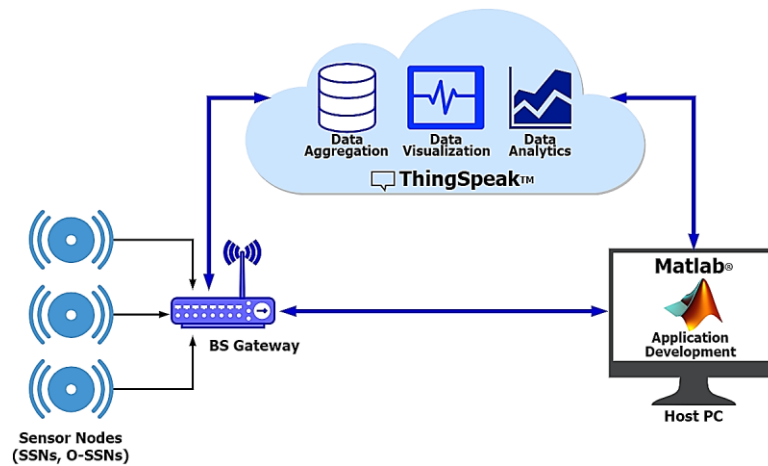
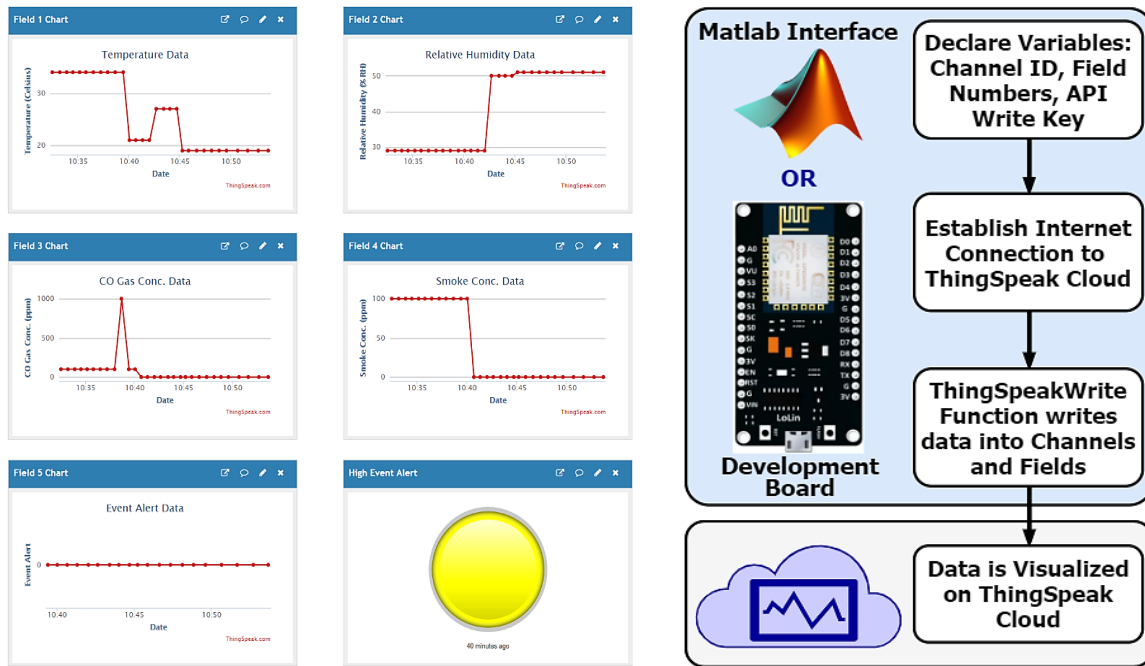


Fig. 4.20. ThingSpeak cloud platform

a. ThingSpeak Visualization

ThingSpeak cloud platform visualizes data in channels, with each channel having a maximum of 8 fields to hold incoming sensor node data. Sensor node data are visualized in real-time as each sensor data stream is plotted in its corresponding field. Also, each channel can have additional

fields to display the results of visualization applications, widgets, etc. For the developed ANR system, each sensor node channel has five fields for temperature, humidity, CO gas concentration, smoke concentration, and event alert data, respectively. An LED widget was added to each channel to indicate the detected events: “yellow” for high alert events and “red” for critical alert events and ‘gray’ for normal events. Every channel field can aggregate and display data ranging from a few data points (<5), to many data points (>100) over an extended time. Fig. 4.21(a) shows a ThingSpeak channel view.



(a). ThingSpeak channel data fields

(b). ThingSpeak data upload process

Fig. 4.21. ThingSpeak Channel Data Fields and Data Upload Process

Each ThingSpeak channel has an ID where the channel fields are assigned ‘field numbers,’ and application program interface (API) read and write keys used to access the channel data. The following steps write and visualize data on a channel either through Matlab interface or through a development board as shown in Fig. 4.21(b):

- i. The channel ID, channel ‘fields numbers’ and API ‘write’ key are declared as long integers, integers, and characters respectively.
- ii. Establish an internet connection between the Matlab interface or the development board and ThingSpeak cloud.
- iii. ThingSpeak functions are used to write the sensor node data into the corresponding channel and fields using their declared field numbers, ID and API write character.

- iv. At the ThingSpeak online interface, the written data are plotted on the corresponding channel fields in real-time.

The API ‘read’ keys are used similarly to reading data from the various fields and channels during data analytics.

b. ThingSpeak Analytics

Statistical and preservation metrics analytics can be performed on the data stored in the ThingSpeak cloud platform. The results of these analytics can be used to make decisions, such as maintaining a specific temperature/humidity within the zones and the type of materials to be stored within the zones.

i. Statistical Analytics:

Statistical analytics provides information on the nature of events and involves the development of applications to calculate and display essential statistical data such as the maximum, minimum, and mean value for the temperature, humidity, CO gas, and smoke concentration detected within each zone of the operational environment. These essential statistical data are displayed alongside their respective timestamps. Other in-depth statistical analytics performed includes regression analysis on the temperature and humidity data from the zones of the operational environment.

ii. Preservation Metrics Analytics

Preservation metrics analytics deals with the prediction of metrics that evaluate the storage qualities of indoor environments [147]. The metric evaluation is very important for storage facilities such as warehouses as they represent the degree of risk for multiple forms of material decay. Material decay involves environmentally induced chemical decay in organic materials, corrosion in metallic materials, molds (biological decay), etc. Hence, performing preservation metrics analytics on the zones’ sensor node data provides information on the type of materials that can be stored within the zones and their estimated storage duration.

Therefore, two preservation metrics are evaluated, namely the equilibrium moisture content (%EMC) and preservation index (PI) [147]. The %EMC metric represents the risk of environmentally induced metal corrosion while the PI metric represents the risk of environmentally induced chemical degradation in organic materials that leads to aging. The %EMC and PI metrics are dependent on the zones’ temperature and humidity data. Hence, they are compared to the dewpoint (DP) analysis, which measures the moisture content within the

zones. The comparisons are used to observe the trends and relations between the metrics and the dewpoint (moisture content) over time. Brief details on the dewpoint and the preservation metrics with their method of prediction from the zones' temperature and humidity data are hereby presented.

1. Dewpoint

Dewpoint offers insight into how much moisture (water vapor) is contained within an environment. It is calculated from temperature (T) – °C and relative humidity (RH) values from within the zone environment. It is expressed in equation (4.5), called the Magnus formula [148]:

$$T_{DP} = \frac{c \left(\log \left(\frac{RH}{100} \right) + \frac{bT}{c+T} \right)}{b - \left(\log \left(\frac{RH}{100} \right) + \frac{bT}{c+T} \right)} \quad (4.5)$$

Where b is the water vapor constant (b = 243.5°C) and c is the barometric pressure constant (c = 17.67). Air moisture content calculated from dewpoint can be used to determine the extent of rust in metals and mold in organic materials.

2. Equilibrium Moisture Content (% EMC)

A material attains EMC when it is neither losing nor gaining moisture from its environment [149]. It is a function of the relative humidity (RH) and temperature (T) – °C of the environment. % EMC can be used to predict the risk of environmentally induced metal corrosion. It is expressed in equation (4.6(a) – (e)) [150]:

$$\% EMC = \frac{1800}{W} \left(\frac{K \cdot RH}{1 - K \cdot RH} + \frac{K_1 \cdot K \cdot RH + 2 \cdot K_1 \cdot K_2 \cdot K^2 \cdot RH^2}{1 + K_1 \cdot K \cdot RH + 2 \cdot K_1 \cdot K_2 \cdot K^2 \cdot RH^2} \right) \quad (4.6a)$$

Where W, K, K1, K2 are the absorption model coefficients, expressed as:

$$W = 349 + 1.29T + 0.0135T^2 \quad (4.6b)$$

$$K = 0.805 + 0.000736T - 0.00000273T^2 \quad (4.6c)$$

$$K_1 = 6.27 - 0.00938T - 0.000303T^2 \quad (4.6d)$$

$$K_2 = 1.91 - 0.0407T - 0.000293T^2 \quad (4.6e)$$

According to the preservation metrics standards from Image Permanence Institute (IPI), metal corrosion can be estimated based on the maximum %EMC value for a 30-day running average of temperature/RH [151]. This principle is used and applied in a reduced time scale (the duration of

the system simulation/demonstration) to predict the %EMC at the various zones of the environment. The level of damage based on the maximum %EMC is shown in Table 4.7.

Table 4.7. Metal corrosion risk based on %EMC values

Max. %EMC Value	Metal Corrosion Risk
Max. %EMC ≤ 7	GOOD : minimal risk of corrosion
$7.1 \leq \text{Max. \%EMC} \leq 10.5$	OK : limited risk of corrosion except for highly sensitive materials
Max. %EMC ≥ 10.5	RISK : high risk of corrosion

3. Preservation Index (PI)

PI estimates the rate of environmentally induced chemical decay in organic materials that causes aging. It is based on the storage environment's temperature (T in Kelvin – K) and relative humidity (RH) [147]. This index is evaluated based on a careful study of the rate of the hydrolysis of cellulose acetate contained in organic materials [152]. From multiple study results, the IPI has formulated the reaction kinetics for the duration (in years) of the deterioration of organic materials expressed in equation (4.7).

$$PI = \frac{e^{\left(\frac{E-134.9RH}{8.314T} + 0.0284RH - 28.023\right)}}{365} \quad (4.7)$$

where E is the activation energy, E = 95220 J/mol

This equation outputs the time for organic materials to deteriorate based on a single entry of temperature and relative humidity. But the IPI developed a more accurate estimation called the Time Weighted PI (TWPI) which aggregates the temperature and relative humidity data over a range of time. The rate of chemical decay for organic materials based on PI values are given in Table 4.8 [151].

Table 4.8. Time ranges for organic material decay based on PI values

PI Value (years)	Rate of Organic Material Decay
PI > 75	GOOD : slow organic decay
$45 \leq \text{PI} \leq 75$	OK : acceptable organic decay except for highly decaying materials
PI < 45	RISK : accelerated rate of organic decay

4. Preservation Analytics Approach

To accurately predict the %EMC and PI within the different zones, a neural network (NN) is trained on the temperature and RH data and used to predict the %EMC and PI preservation metrics. The temperature and RH data are the inputs, while the generated %EMC and PI are targets of the

NN. The NN has two inputs and two outputs, 10 hidden neurons minimizing the errors from either undertraining or overtraining the network. The network is designed to be trained on any range of data depending the user. The NN diagram is shown in Fig. 4.22.

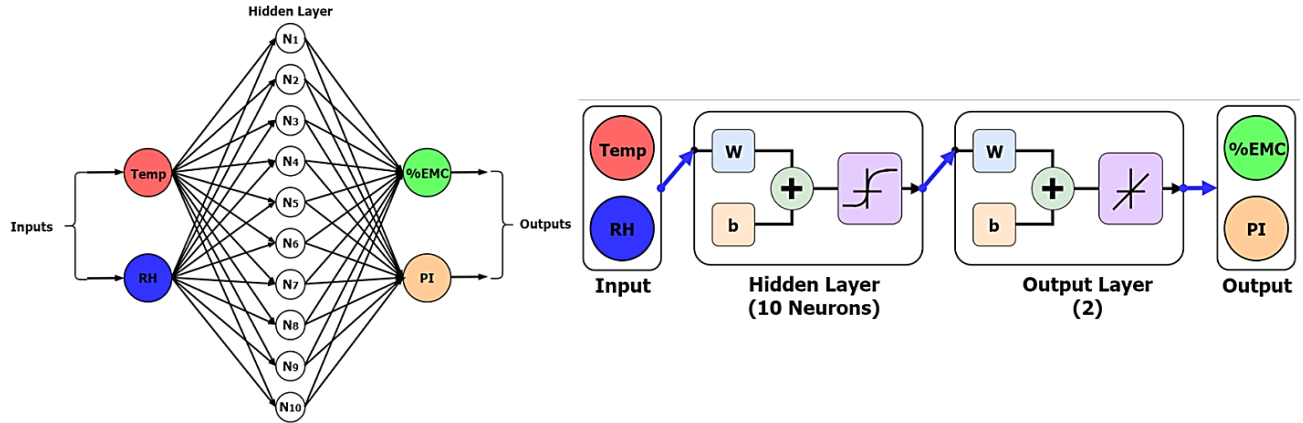


Fig. 4.22. Neural network for storage conditions prediction

From Fig. 4.22, the neural network has the following properties:

- A feedforward network with a mean squared error (MSE) performance function.
- The hidden layer is evaluated using a non-linear sigmoid function to learn the non-linear relationships between inputs and the targets.
- The output layer is evaluated using a linear function to perform the regression analysis, between the outputs and the targets of the network.
- The input data is divided into 80% training data, 10% testing, and 10% validation. The Levenberg-Marquardt (LM) method is used to train the network [153]. The LM algorithm has an incremental learning rate, i.e. the learning rate increases after each iteration of the algorithm, thereby increasing the training accuracy [154]. It has been shown to perform better than algorithms based on variable learning rates [155]. The LM algorithm is expressed in equation (4.8) [156]:

$$W_{i+1} = W_i - [J^T(x)J(x) + \mu_k I]^{-1} J^T(x) \cdot e \quad (4.8)$$

Where W is the network weights, J is the Jacobian matrix of the first derivatives of network error with respect to the weights and biases, e is the network error vector, I is the identity matrix and μ_k is the value responsible for the speed of convergence and stability of the LM algorithm.

Chapter 5

SIMULATION DEVELOPMENT ENVIRONMENT (SDE) WITH HIL

The ANR system simulation development environment (SDE) constitutes the integrated development of the WSNs and MRS simulation models (WSNs-SMs & MRS-SMs) that are featured by the support of hardware-in-the-loop (HIL) and the integration of cloud technologies. The WSNs simulation models comprise the sensor nodes (SSNs, O-SSNs) and the gateway (BSG), while the MRS simulation models include the mobile robots – MRSNs & G-MRSNs. The WSNs-SMs are developed and simulated using the Proteus® Circuit Design Simulation Environment (PCD-SE) integrated with Arduino®, while MRS-SMs are developed and simulated using LabVIEW® Robotics Environment Simulator (LRES). The MS SSMS and Matlab software facilitate cloud support.

5.1 Design Structure of the SDE

Referring to the developed ANR system architecture shown in Fig. 4.1 of chapter 4 that constitutes both the hardware and simulation components, the SDE facilitates the development and integration of the SMs, SEs and GUIs featured by the use of HIL and cloud technologies, as illustrated in Fig. 5.1. The SDE has two primary roles:

- a. Enables the development of standalone system simulation with no HIL.
- b. Enables module based HIL testing and simulation during the system development to support hardware implementation.

In addition, the SDE is featured by four main development capabilities:

- a. The WSNs-SMs development using the PCD-SE.
- b. The MRS-SMs development using the LRES and MRS GUI.
- c. The BS data aggregation and visualization with the SND GUI and cloud integration.
- d. The HIL module-based testing.

The SDE facilitates the system simulation whereby events data is communicated from the WSNs-SMs to the MRS-SMs and BS GUIs, and finally from the BS GUIs to the cloud platforms

as shown in Fig. 5.1. Hardware system modules such as sensor nodes can be incorporated into the SDE for HIL based simulation.

Fig. 5.1. Simulation Development Environment

The WSNs-SMs are developed and simulated in the PCD-SE. The PCD-SE is a propriety software suite for electronic design automation [157]. It has the following features and advantages:

Due to the availability of many device libraries and the ease of MCU simulation in the PCD-SE, Arduino-based MCUs and sensors libraries were used to develop and simulate the WSNs-SMs as described in subsections 5.2.1 and 5.2.2.

The WSNs-SMs constitute all static and onboard sensor nodes (SSNs, O-SSNs and BSG). Each of the sensor nodes comprises the following PCD-SE libraries representing the structural contents of the sensor nodes presented in chapter 4:

A. SSN – SM

The SSN-SM constitutes of the following PCD-SE part libraries, as shown in Fig. 5.2(a):

- a. Microcontroller (MCU): The MCU is used for data processing and it is represented using Arduino Uno MCU part libraries. The data processing programs are written in the C++ based Arduino IDE; then Hex files are generated and integrated during simulation into the Arduino MCU part libraries in the PCD-SE.
- b. Sensors: Sensor part libraries are used to generate data representing various types of events. The temperature/humidity sensor are represented by the Digital Temperature Humidity (DTH – 11) part library, motion and flame sensors are represented by customized part libraries supporting their respective sensors, while the gas sensors (CO Gas and Smoke) are represented by potentiometer part libraries that generate analog data.
- c. Indicators: LEDs and LCD indicators are represented by part libraries of the same nature. The Green LEDs are used to indicate normal events, while the red LEDs are used to indicate high and critical events alerts. The LCDs shows text characters representing the various events detected.
- d. Communication Modules: Serial com port part libraries are used to represent the communication modules since a serial emulator software is used to simulate the system networks. Image overlays on the com port part libraries are used to differentiate the network protocols. They include “XBee,” “HC” and “RS232” image overlays for ZigBee, Wireless serial and serial protocols respectively.
The SSN-SM contains the “XBee” com port part library representing the ZigBee protocol.
- e. Virtual Terminals: are used to visualize the processed data being sent from each node.

B. MRSNs’ O-SSN – SM

The MRSNs’ O-SSN-SMs are similar to the SSN-SM, as shown in Fig. 5.2(b), except for the following part libraries:

- a. Microcontroller (MCU): the MCU is represented by the Arduino Mega part library because its multiple hardware serial ports facilitate connection to three different networks.
- b. Communication Modules: The MRSNs’ O-SSN-SM contains the following network protocols:
 - i. “X-Bee com port library representing the ZigBee protocol for communication with the SSN-SMs,

- ii. “HC” com port part library representing the Wireless Serial protocol for communication with the BSG and G-MRSNs and,
- iii. “RS232” com port part library for serial communication with the robot’s motion control.

C. G-MRSNs’ O-SSN – SM

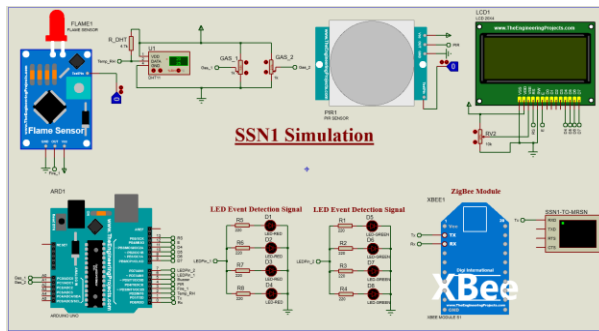
The G-MRSN O-SSN-SM is similar to the SSN-SM as shown in Fig. 5.2(c), except for the following libraries:

- a. Microcontroller (MCU): the MCU is represented by the Arduino Mega part library because its multiple hardware serial ports facilitate connection to three different networks.
- b. Communication Modules: The G-MRSN’s O-SSN-SM contains the “HC” com port part library representing the Wireless Serial protocol for communication with the MRSNs and BSG.

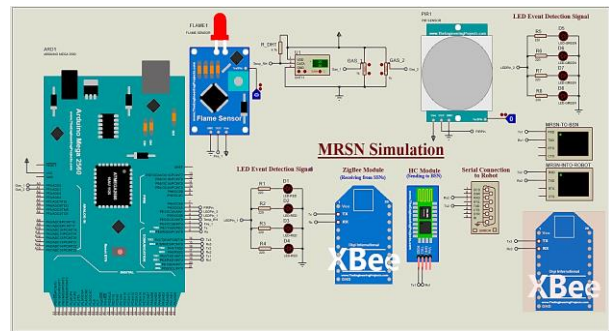
D. BSG – SM

The BSG-SM has no sensors or indicators since it operates as a gateway to communicate detected data from the WSNs-SMs to the SND GUI and ThingSpeak cloud platform. Hence, it constitutes the following communication modules as shown in Fig. 5.2(d):

- a. The “HC” com port part library representing the Wireless Serial protocol for communication with the MRSNs and G-MRSN.
- b. Two “RS232” serial com port part libraries connecting the BSG to the SND GUI and ThingSpeak cloud platform through Matlab.



(a). SSN



(b). MRSN O-SSN

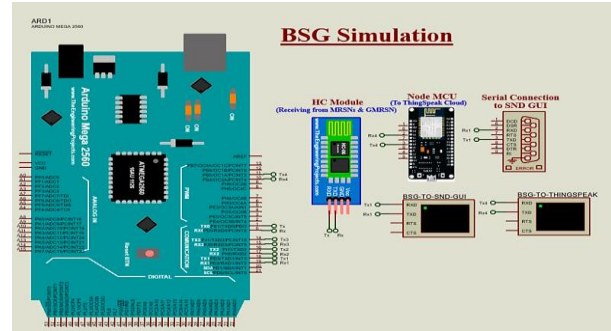
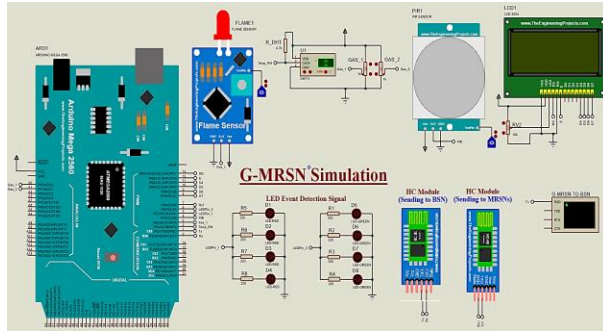
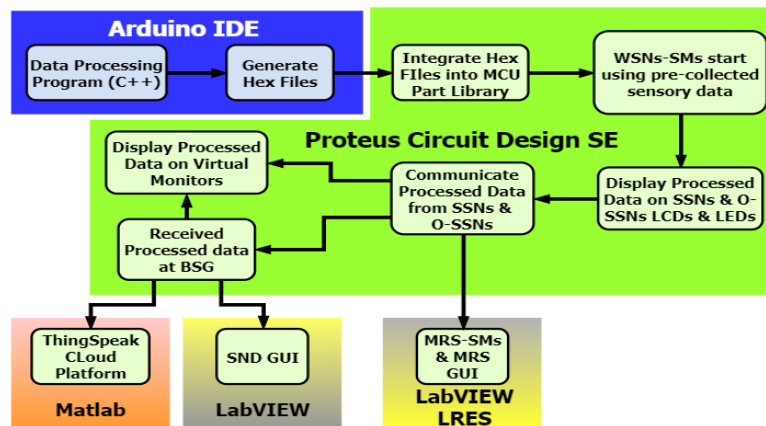


Fig. 5.2. WSNs-SMs in PCD-SE.

5.2.2 WSNs Simulation: The Operation and Test

The circuit components of the WSNs-SMs are simulated to generate and communicate the various type of events in association with the sensory information. The simulation involves the following steps as illustrated in Fig. 5.3:



- a. A data processing program for each WSNs-SM reflecting the sensor node operation algorithm presented in section 4.2.3 and 4.3.1. This program is written in C++ in the Arduino IDE and its corresponding Hex files are generated and integrated into the MCU part library.
- b. The simulation starts when the user presses the simulation start button on the PCD-SE for each WSN-SM, and accordingly the WSN-SMs start running using the sensory data obtained from the various sensor part libraries based on pre-collected sensory data and enabled by user interaction with the PCD-SE interface. Fig. 5.4 shows one of the WSN-SM.

- c. The sensory data is processed by the MCU part library, and the results are displayed through the LEDs and text message on the LCD of each WSN-SM as relevant as shown in SSN1-SM and the global robot sensor node (G-MRSN O-SSN) SM examples of Fig. 5.5 (a) and (b) respectively where the LCD is displaying the following text messages for SSN1-SM:

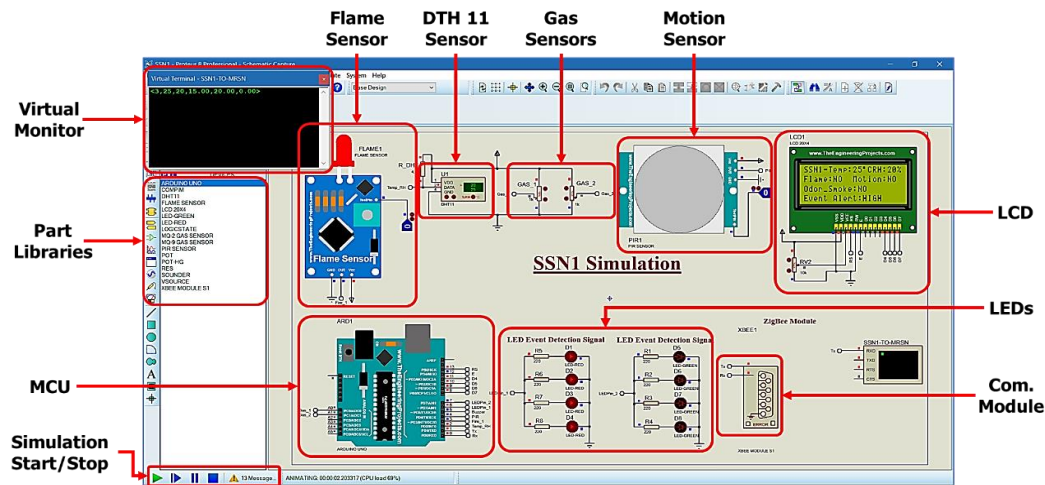
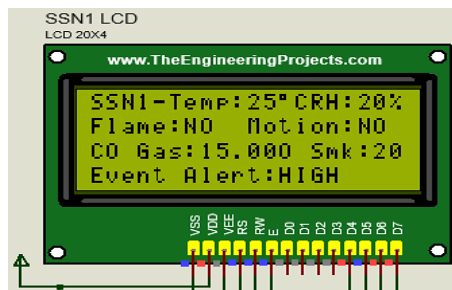
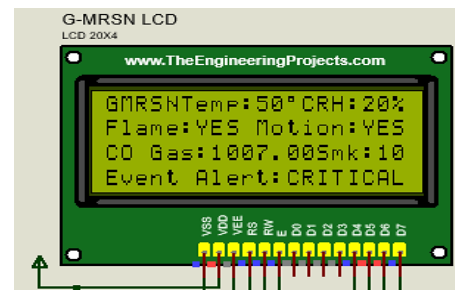


Fig. 5.4. WSN-SM (SSN) during simulation

- The text message “SSN1 – Temp: 25⁰C RH: 20%” where ‘SSN1’ is sensor node, ‘Temp’ is the temperature (25⁰C) and ‘RH’ is the relative humidity (20%).
- The text message “NO” against the “Flame” and “Motion” labels to indicate no visible flame, and no human motion is detected.
- The text message “CO Gas: 15.000 Smk: 20” where ‘CO Gas’ is CO gas concentration (15ppm) and ‘Smk’ is the smoke concentration (20ppm).
- The text message “HIGH” against, the “Event Alert” label, to indicate the event characterization output of the fuzzy logic decision system.



(a) SSN1 LCD Display



(b) G-MRSN O-SSN LCD Display

Fig. 5.5. SSN1 and G-MRSN O-SSN LCD.

- d. The obtained results can be visualized on the virtual terminal available at each of the sensor nodes as shown in Fig. 5.6(a) – (c) respectively. For example, SSN1, G-MRSN O-SSN and MRSN O-SSN have the following displayed on their virtual monitors:
- i. For the SSN1-SM of Fig. 5.6(a), the virtual terminal is displaying the data $\langle 3, 25, 20, 15.00, 20.00, 0.00 \rangle$, where the NN = 3, temperature = 25°C, RH = 20%, CO Gas concentration = 15ppm, Smoke concentration = 20ppm and Event Alert = 0 (High Alert).
 - ii. For the MRSN O-SSN SM (Fig. 5.6(b)), the following data is displayed:
 1. The location data of the alerting SSN-SMs communicated into the robot control system through the communication modules (serial com port part libraries) in the following format:
 - $2.00, 8.50, 0.00 \rightarrow x = 2.00\text{m}, y = 8.50\text{m}, \text{event alert} = 0.00$ from SSN1
 - $6.00, 2.00, 0.00 \rightarrow x = 6.00\text{m}, y = 2.00\text{m}, \text{event alert} = 0.00$ from SSN2
 2. The received results from the each related SSN SM and the results from of each MRSN O-SSN SM are communicated to the BSG following the same format as in Fig. 5.6(a) and shown in Fig. 5.6(c):
 - $\langle 3, 25, 20, 15.00, 10.00, 0.00 \rangle$ from SSN1 to BSG
 - $\langle 1, 40, 27, 13.00, 18.00, 0.00 \rangle$ from MRSN1 to BSG
 - $\langle 2, 38, 28, 14.00, 20.00, 0.00 \rangle$ from MRSN2 to BSG
 - $\langle 5, 23, 21, 30.00, 40.00, 0.00 \rangle$ from SSN3 to BSG
 - iii. For the G-MRSN of Fig. 5.6(c), the results of its O-SSN are communicated to the BSG in the same format as SSN1 and MRSN O-SSN SMs:
 - $\langle 11, 50, 20, 1007.00, 10013.00, 1.00 \rangle$ from G-MRSN to BSG
- e. In addition, the obtained result is communicated through the communication modules (serial com port part libraries) of the BSG to the SND GUI and ThingSpeak cloud platform in the following format:
- $1, 40, 27, 13.00, 18.00, 0.00$
 - $2, 38, 28, 14.00, 20.00, 0.00$
 - $5, 23, 21, 30.00, 40.00, 0.00$
 - $3, 25, 20, 15.00, 10.00, 0.00$

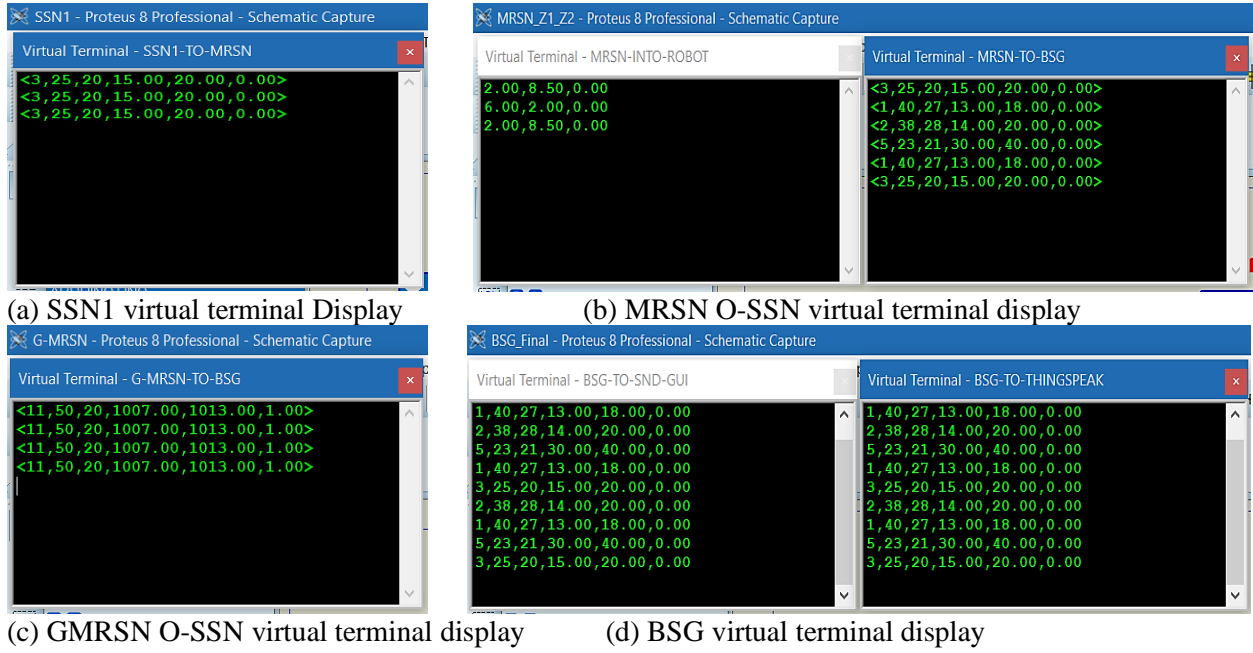


Fig. 5.6. WSNs-SMs virtual terminal displays.

5.2.3 The Network Emulator

The network emulator enables the connection between the various sensor nodes, robots, GUIs and cloud platforms across the different simulation environments (SEs). These connections are facilitated by the creation of virtual com ports pairs on the network simulator interface as shown in Fig. 5.7. The following virtual connection links are created:

- The connection between the SSNs, O-SSNs, BSG simulation models to simulate the WSN two-tier network i.e. SSNs-to-O-SSNs ZigBee network and O-SSNs-to-BSG wireless serial network.
- The serial connection between the O-SSNs and the robots' whereby alerting SSN locations from the O-SSNs are communicated to the robots' motion control to enable them to plan and navigate paths to the alerting SSNs.
- The connection between the ThingSpeak cloud platform and the BSG through Matlab SE to simulate the wireless connection of the BSG to the ThingSpeak cloud.

The virtual connections are established by pairing the com ports linking the various simulation models. For example, the virtual com port pair COM1↔COM2 connected SSN1 and MRSN1 O-SSN simulation models as shown in Fig. 5.7.

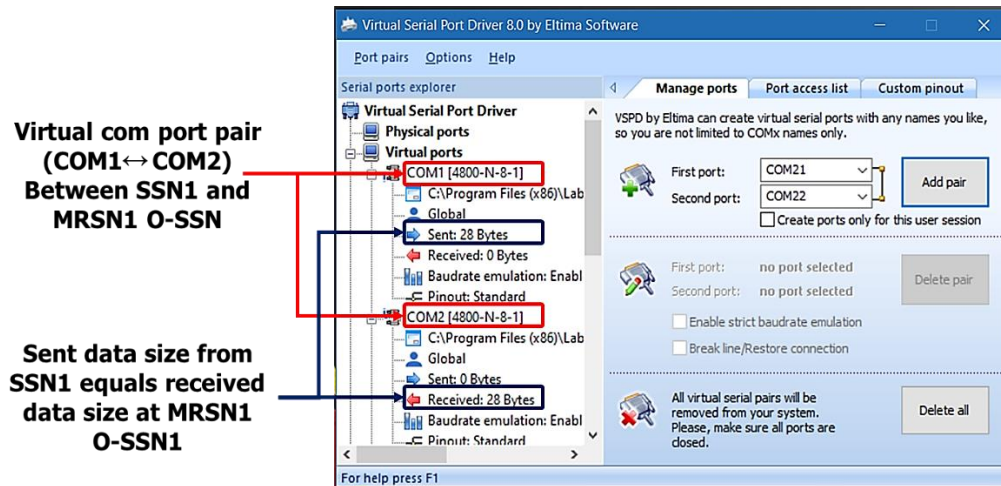


Fig. 5.7. Network emulator interface

5.3 MRS-SMs Development

The MRS-SMs are developed using the LRES integrated with the MRS GUI as presented:

- a. The LabVIEW Robotics Environment Simulator (LRES) Environment: the LRES is used to develop the physical models of the MRS-SMs and their navigation environment (NE).

The LRES has the following features:

- i. It is based on the Open Dynamics Engine software development kit (SDK) [158] that facilitates the simulation of rigid bodies.
- ii. Allows the importation and customization of 3D CAD (in Collada format) models for robot and environment design, and
- iii. Facilitates the visualization of the movement of robot models within NE during the simulation.

- b. The MRS GUI: the MRS GUI is used to develop the control models of the MRS-SMs. The MRS GUI has the following features:

- i. It is based on the virtual instruments (VIs) of the LabVIEW Robotics Module and the LabVIEW VISA hardware drivers.
- ii. Enables the development, testing and validation of navigation algorithms on the MRS-SMs before real-time deployment.

5.3.1 MRS-SMs and Navigation Environment

The MRS-SMs are CAD models of the actual LabVIEW DaNI 1.0 robots [159] available in the LRES, while the navigation environment (NE) is designed using SolidWorks® CAD software

and imported as a Collada file format into the LRES. Both models are developed in the LRES environment while the navigation algorithms are developed using the MRS GUI.

A. MRS-SMs and NE CAD Models

The MRS-SMs physical models are based on the National Instruments (NI) DaNI 1.0 starter kit robot [160], while the NE was designed in CAD software and imported into the LRES.

a. The MRS-SMs (MRSNs and G-MRSNs) CAD Models

The MRS-SM physical models constitute of the parts of the robots that enable the robot to move and sense its environment. They include the following part models as shown in Fig. 5.7 of MRSN1-SM:

- i. The Physical Frame: the physical frame is based on the Pitsco educational frame structures for easy integration of add-on hardware.
- ii. Drive System: The drive system model is a differential drive model of two 12V DC motors, each connected to two wheels on the left and right side of the robot through a gear system as shown in Fig. 5.8.
- iii. Sensors: There are two sensor models, the ultrasonic sensor model for distance/range measurement which is mounted on a servo motor as shown in Fig. 5.8. Also, there the optical encoders for tracking the distance covered by the robot over time.

The zone-based MRSNs and the G-MRSN SMs are distinguished within the NE by their customized colors, for example, MRSN1 is blue, MRSN2 is red, MRSN3 is green, MRSN4 is brown and G-MRSN is gray. Fig. 5.9 shows the blue customization of MRSN1-SM.

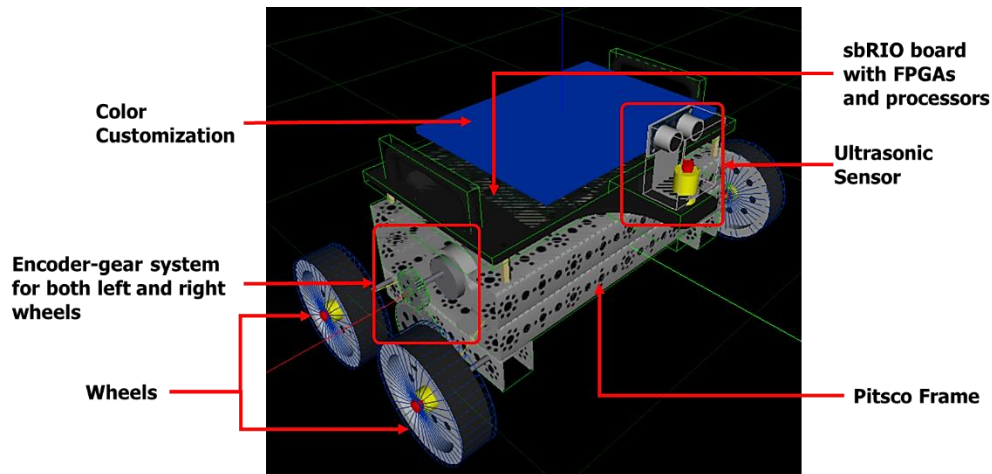
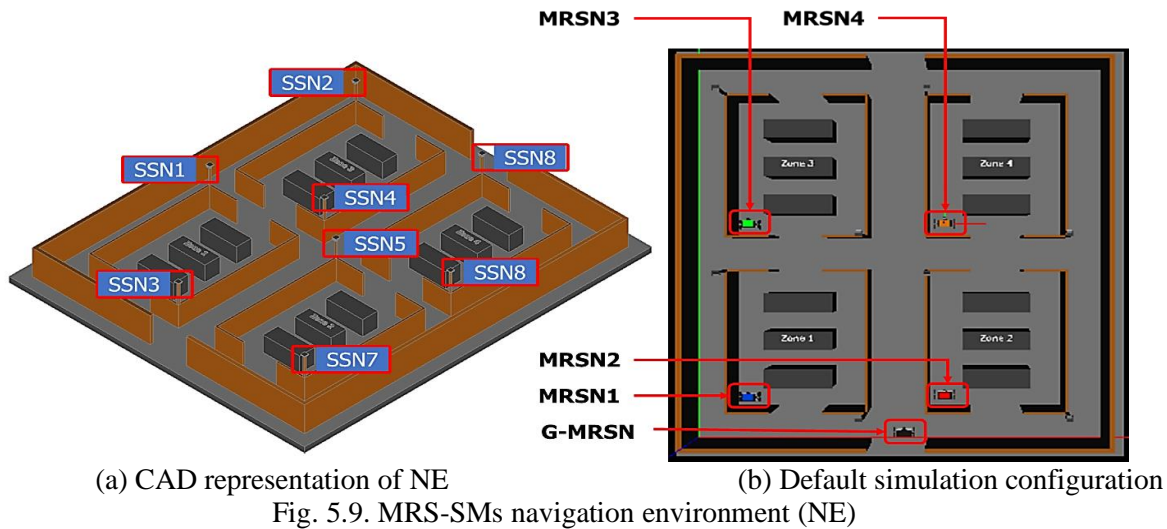


Fig. 5.8. MRSN1-SM: DaNI 1.0 CAD model.

b. The Navigation Environment

The navigation environment is designed to reflect the task space and it is based on the occupancy grid map as described in section 4.3.3 of chapter 4. The designed environment contained four zones, with two SSNs in each zone: SSN1 & SSN3 – zone1, SSN5 & SSN7 – zone2, SSN2 & SSN4 – zone3, SSN6 & SSN8 – zone4. Static obstacles representing storage racks are placed within each zone. Fig. 5.9(a) shows the CAD representation of the NE, while Fig. 5.9(b) shows the default configuration for the simulation.



B. MRS-SMs Navigation Control Models

The navigation models constitute the physical parts on which the navigation algorithms are developed using the MRS GUI. They constitute the following part models based on the robot's circuit diagram as shown in Fig. 5.10:

- i. The Controller: The controller model is based on the integrated single-board reconfigurable input/output (sbRIO) controller with reconfigurable FPGA, real-time processors and other supporting hardware. The FPGA facilitates the programming of the robot while the real-time processor facilitates the operation of the robot in real-time. Data from connected sensors and other hardware are obtained through the input/output ports. LabVIEW robotics module and LabVIEW VISA hardware drivers virtual instruments. (VIs) blocks are used to develop the required programs that enable the robots to function. The controller model of the MRS-SM connects through a VI comport to the O-SSNs to receive the location of alerting SSNs.

- ii. Communication and Connection Ports: they include an Ethernet port for connection and communication with host PC and serial com port for connection with O-SSNs.
- iii. Breakout I/O boards for connection to other hardware, and
- iv. Switches for switching on the motors and robot controller.

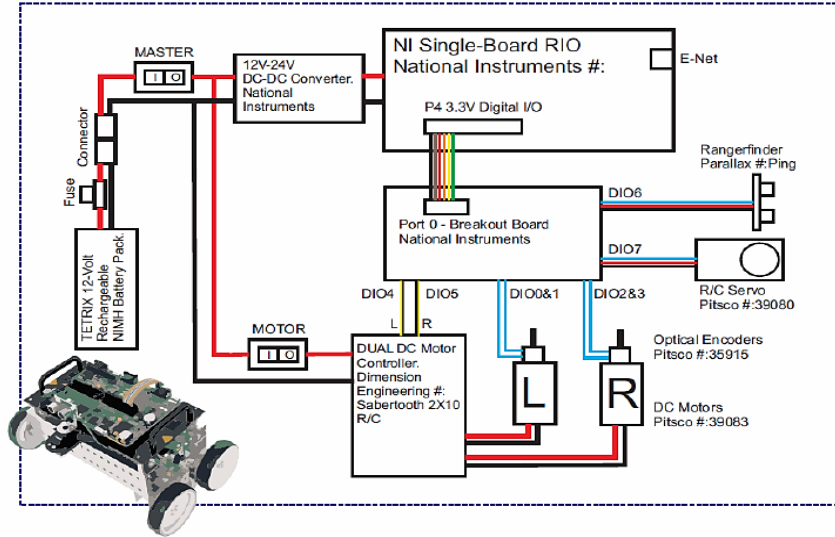


Fig. 5.10. DaNI 1.0 Starter kit circuit block diagram

5.3.2 MRS Simulation: The Operation and Test

Each MRS-SM navigates the assigned zone environment and can navigate preassigned paths within it while the WSN-SMs of that zone are communicating sensory information with the MRS-SMs through their O-SSNs. The simulation process of MRS-SMs navigation is illustrated in Fig. 5.11.

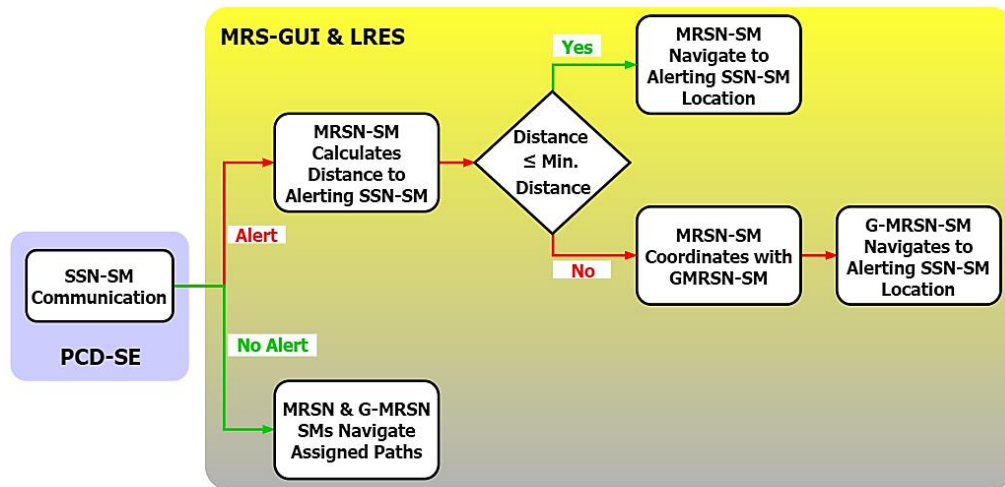


Fig. 5.11. MRS-SMs navigation simulation operation process

MRS simulation involves the following steps:

- a. The run button is pressed to start the MRS GUI, which launches the LRES with the robots at their default positions.
- b. If no alert is issued from the SSNs, each MRS SM navigates assigned paths within/around their zones, with their navigation paths displayed on their respective tab interfaces on the MRS GUI.
- c. When an alert is issued from an SSN, the relevant robot calculates its distance to the alerting SSN. If the distance is less than the navigation distance, the relevant robot navigates to the alerting SSN location, else, it coordinates with the G-MRSN, which navigates to the alerting SSN location.

Based on the MRS simulation process, there three test cases for the MRS SM navigation:

- a. The robots navigating assigned paths within/around the zones of the NE when no SSN alert is received.
- b. The robots (MRSNs) navigating to alerting SSN locations while the G-MRSN continues navigation of its assigned paths, and
- c. An MRSN coordinating with the G-MRSN based on the CNP algorithm, whereby the G-MRSN navigates to the alerting SSN location.

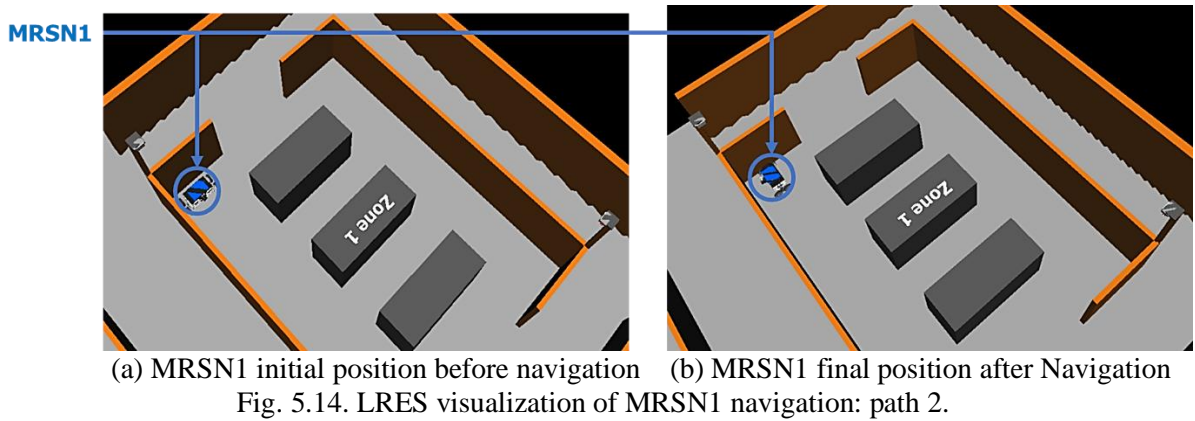
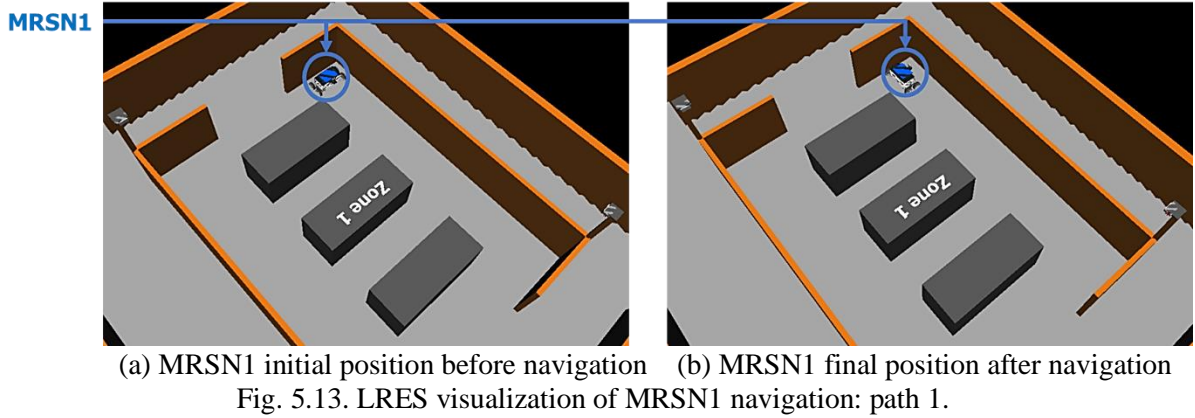
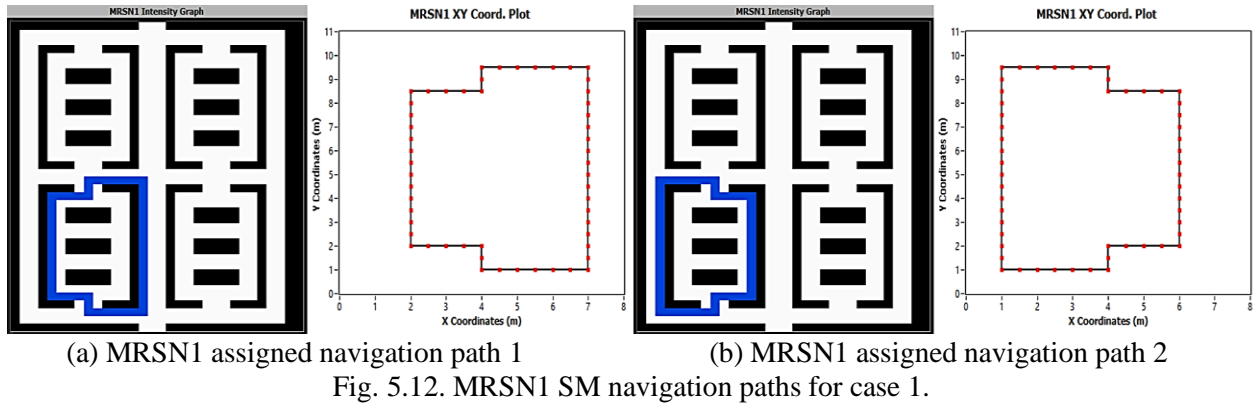
5.3.3 MRS Simulation: Case Examples

As illustrated in Fig. 5.11 there are three case examples for the MRSN and G-MRSN navigation, as presented in the following subsections.

Case 1: MRSNs and G-MRSN navigate assigned paths within/around the zones of the NE

This case occurs when no alert is issued from the SSNs. The MRSNs can then navigate assigned paths within or around the zones while the G-MRSN navigate assigned paths in NE not covered by the zones. Example assigned paths for MRSN1 is shown in Fig. 5.12(a) & (b).

The navigation of MRSN1 within the NE as visualized in the LRES for both paths is shown in the annotated Fig. 5.13 and Fig. 5.14 respectively. Fig. 5.13(a) and Fig. 5.14(a) show MRSN1 at its initial position before navigating the given paths while Fig. 5.13(b) and Fig. 5.14(b) show MRSN1 at its final positions after navigating the given paths.



Since MRSN1 is navigating assigned paths within/around zone, its coordination status is shown in Fig. 5.15.

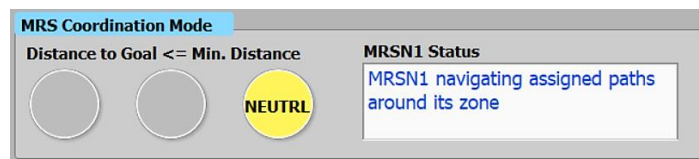


Fig. 5.15. MRSN1 coordination status for case 1

Case 2: MRSNs navigate to alerting SSN locations while G-MRSN continues navigation of its assigned paths.

This case occurs when the MRSNs receive alerts from their various SSNs within the zones (for example, the alert of the presence of an intruder or fire) whereby the MRSNs then navigate to their respective alerting SSN locations. Meanwhile, the G-MRSNs continue navigating assigned paths in NE not covered by the zones. Also, the CNP coordination algorithm between the MRSNs and G-MRSN is not in effect since all MRSNs are within their zones, and the effective distance to the alerting SSN is less than the minimum navigation distance of 7m. Table 5.1 shows the MRNs initial locations, the alerting SSN locations from within the zones, and the effective distance of each MRSN to its relevant alerting SSN.

Table 5.1. MRS-SMs Navigation Coordinates: Case 2

MRS-SMs	Initial Position		Alerting SSNs Location			Effective Dist. (m)
	NE Coord.	Actual Coord.	SSNs	NE Coord	Actual Coord.	
MRSN1	(4, 4)	(2.0, 2.0)	SSN1	(4, 17)	(2.0, 8.5)	6.5 (< 7)
MRSN2	(18, 4)	(9.0, 2.0)	SSN5	(18, 17)	(9.0, 8.5)	6.5 (< 7)
MRSN3	(4, 22)	(2.0, 11.0)	SSN4	(12, 22)	(6.0, 11.0)	4 (< 7)
MRSN4	(18, 22)	(9.0, 11.0)	SSN8	(26, 22)	(13.0, 11.0)	4 (< 7)

The MRS GUI control interface with the alerting SSNs, and their location coordinates and event alerts is shown in Fig. 5.16. A critical event (fire) was detected at SSN8 location in zone 4 while a high alert (intruder) event was detected at SSN5 location in zone 2. The location coordinates for these SSNs are shown in the “alerting SSN notification” label. The same notification indication was obtained for SSN4 (fire) and SSN1 (intruder) respectively.

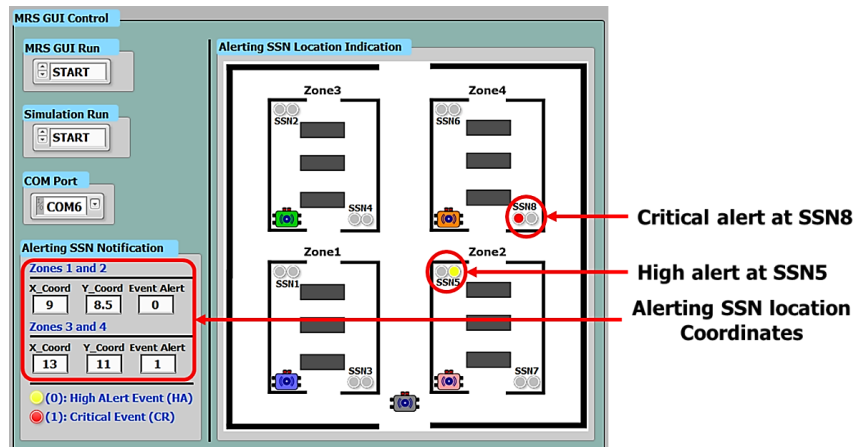


Fig. 5.16. MRS GUI control interface for alerting SSN notifications: case 2

The generated paths to the relevant alerting SSN locations for each MRSN, using the A* algorithm are shown in Fig. 5.17(a) – (d), while the assigned path for the G-MRSN is shown in Fig. 5.17(e).

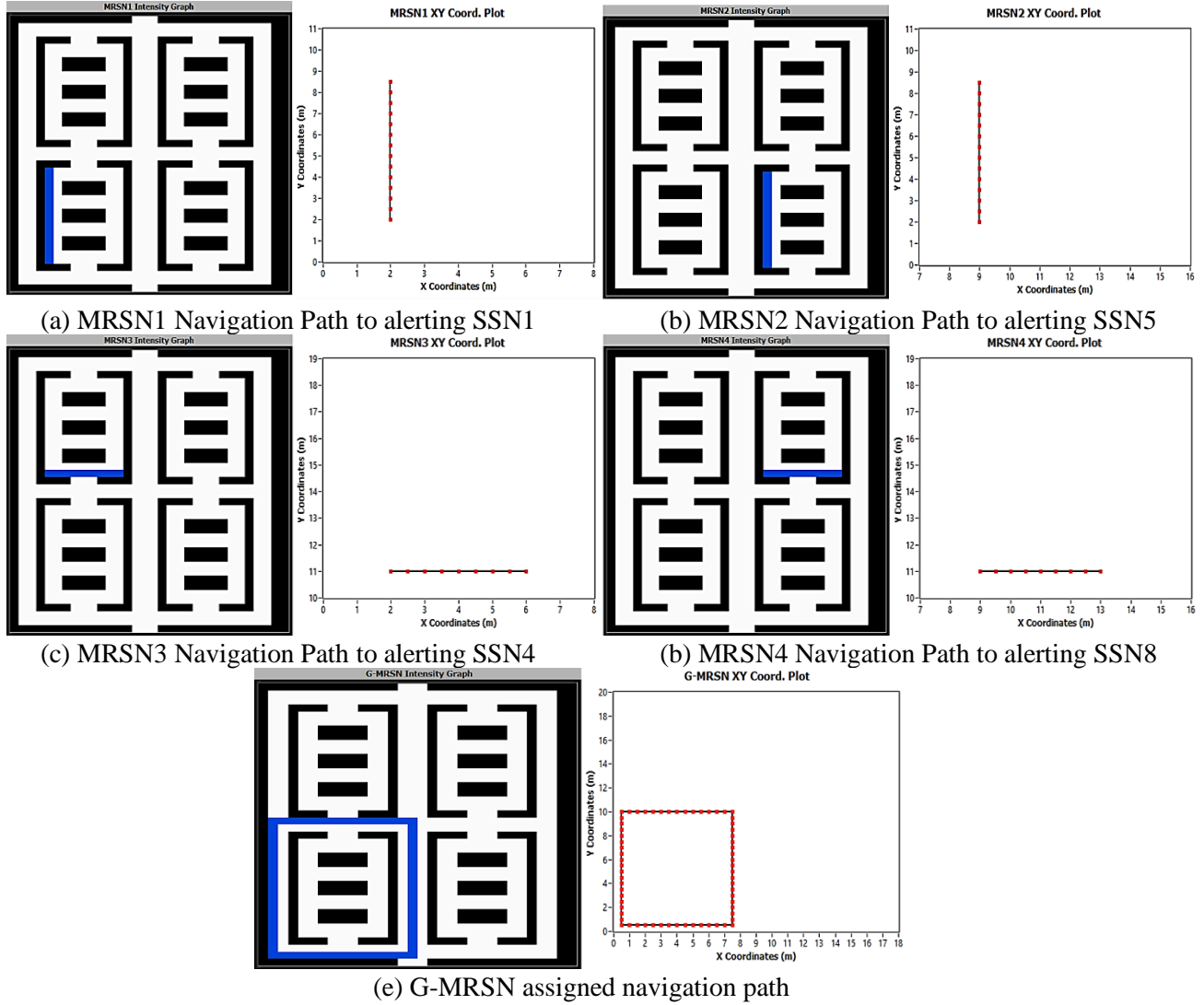


Fig. 5.17. MRS-SMs navigation paths for case 2.

The navigation of the MRS-SMs within the NE as visualized in the LRES is shown in Fig. 5.17, where each MRS-SM is annotated according to its custom color. The intruder and flame events detected within the zones are also annotated. Fig. 5.18(a) shows the robots at their initial positions before navigation while Fig. 5.18(b) – (f) shows them in their final positions after navigation.

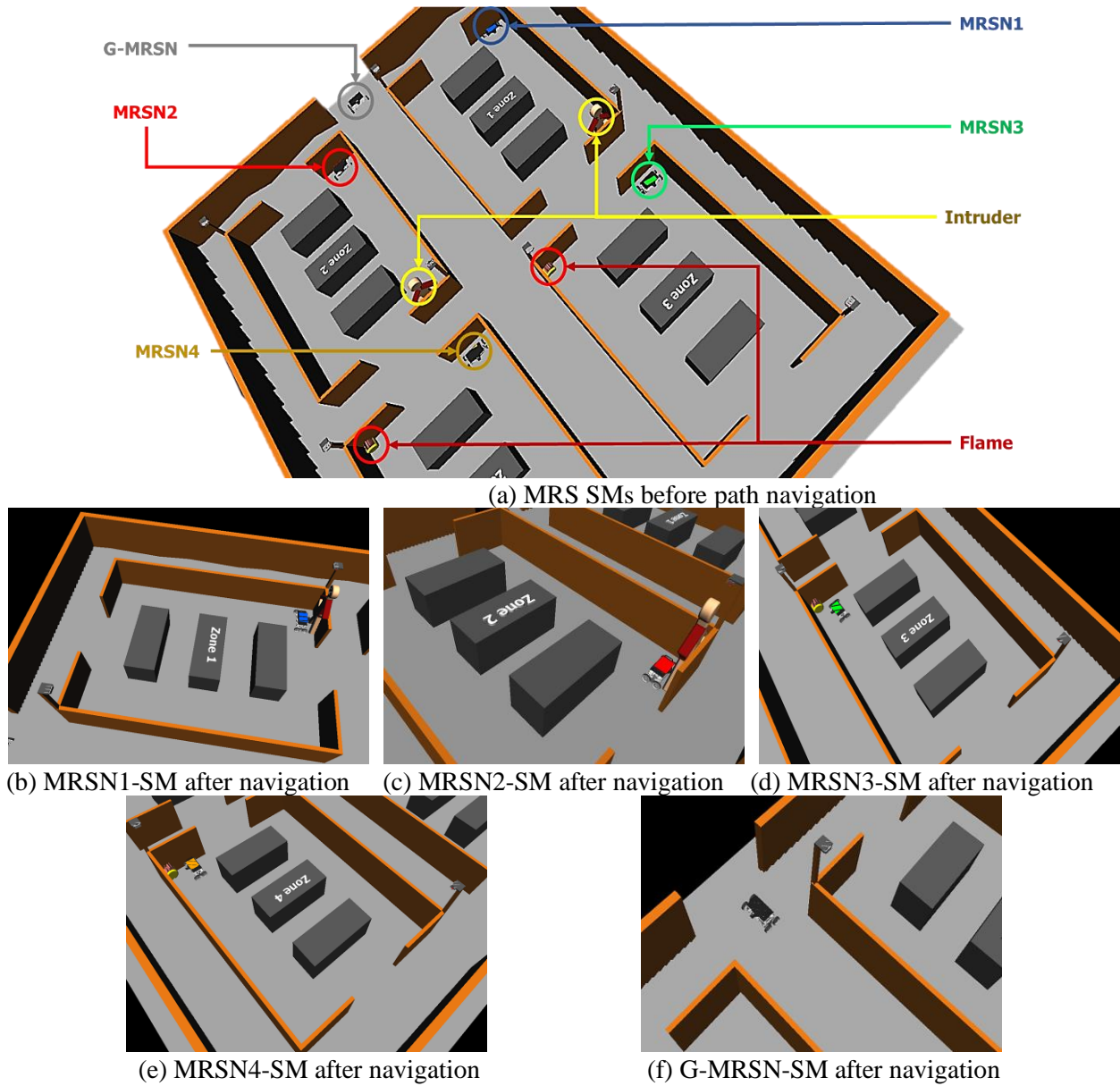
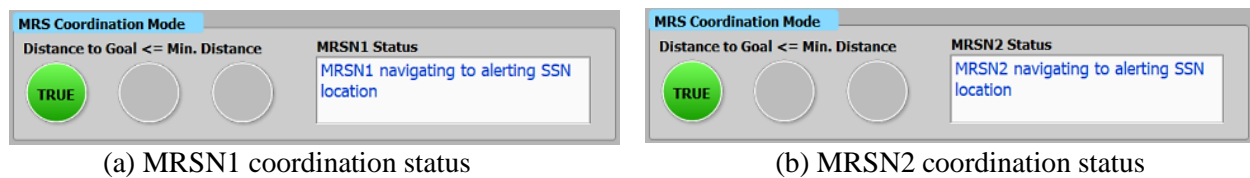


Fig. 5.18. LRES visualization of MRS-SMs navigation: Case 2

All the MRSN-SMs navigated to their respective alerting SSN location; hence, the CNP coordination algorithm was not in effect. The resulting MRS-SMs coordination statuses are shown in Fig. 5.19(a) – (e).



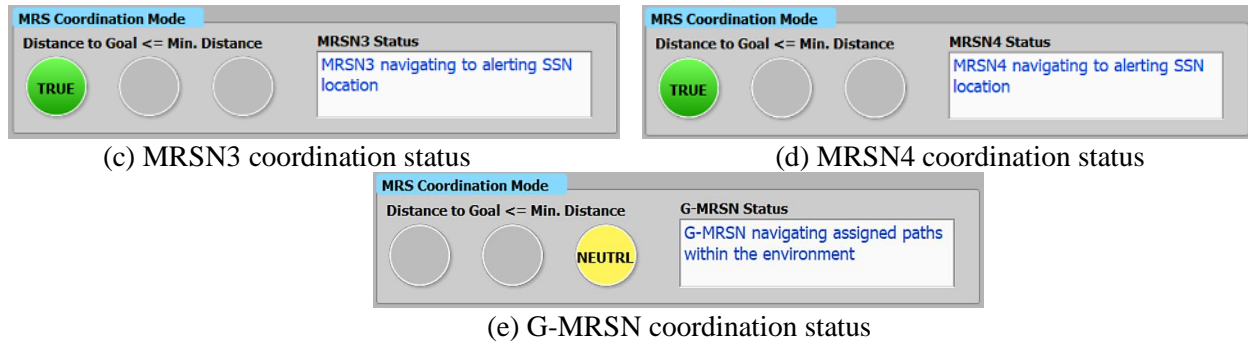


Fig. 5.19. MRS-SM coordination statuses for case 2

Case 3: MRSN-SM Coordination with G-MRSN based on CNP Coordination Algorithm

This case occurs when the distance between an alerting SSN and the relevant MRSN is greater than the minimum navigation distance. Then according to the CNP coordination algorithm, the MRSN coordinates with the G-MRSN, whereby the latter navigates to the SSN location. This case is illustrated with a “fire” alert from SSN4 in zone 3 to MRSN3, which is at a navigating position outside its zone. The G-MRSN is shown to navigate to SSN4 location based on its coordination with MRSN3. Table 5.2 shows the positions and distance analysis of SSN4, MRSN3 and G-MRSN.

Table 5.2. MRS-SMs navigation coordinates: case 3

MRS-SMs	Initial Position		Alerting SSNs Location			Effective Dist. (m)
	NE Coord.	Actual Coord.	SSNs	NE Coord	Actual Coord.	
MRSN3	(3, 37)	(1.5, 18.5)	SSN4	(12, 22)	(6.0, 11.0)	8.75 (> 7)
G-MRSN	(9, 20)	(4.5, 10)	SSN4	(12, 22)	(6.0, 11.0)	1.8 (< 7)

The MRS GUI control interface for this case is shown in Fig. 5.20. A critical event was detected at SSN4 location in zone 3, requiring MRSN3 to navigate to the location. Due to its distance to SSN4, i.e. 8.75m, greater than the navigation distance of 7m, it coordinated with the G-MRSN whose distance to SSN4, i.e. 1.8, is less than the navigation distance. Hence, the G-MRSN navigated to the alerting SSN4 location. The generated path to the SSN4 using CNP and A* algorithms are shown in Fig. 5.21(a) – (b), where MRSN3 is at the navigation position outside its zone (Fig. 5.21(a)) coordinating with the G-MRSN to navigate to SSN4 position as shown in Fig. 5.21(b).

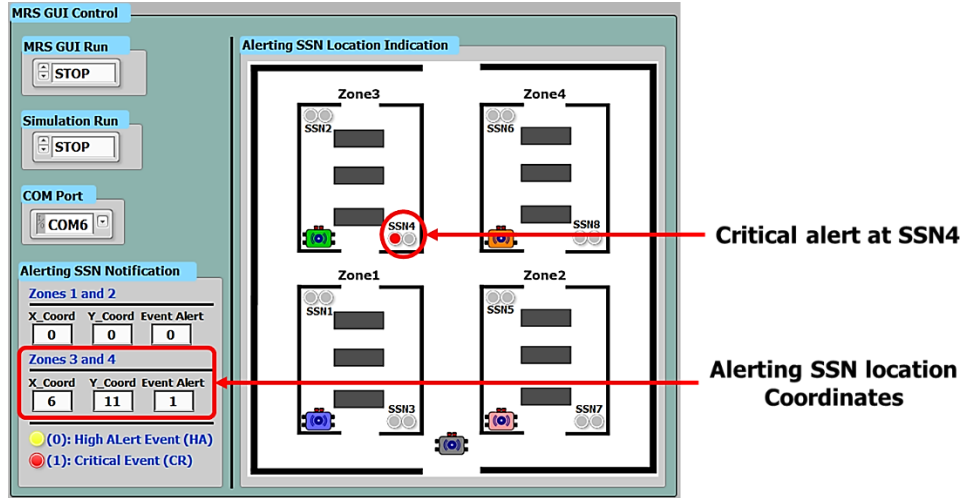


Fig. 5.20. MRS GUI control interface for alerting SSN notification: case 3

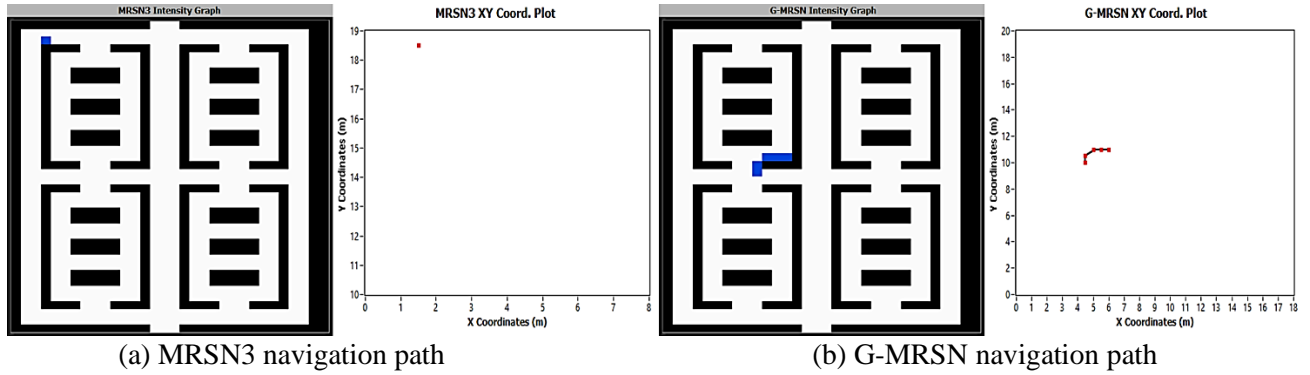


Fig. 5.21. MRS-SMs navigation paths for case 3.

The navigation of the MRS-SMs within the NE as visualized in the LRES is shown in Fig. 5.22(a) – (b), where each MRS-SM is annotated according to its custom color, including the detected event. Fig. 5.22(a) shows the robots at their initial positions before coordination and navigation while Fig. 5.22(b) shows them in their final positions after coordination and navigation.

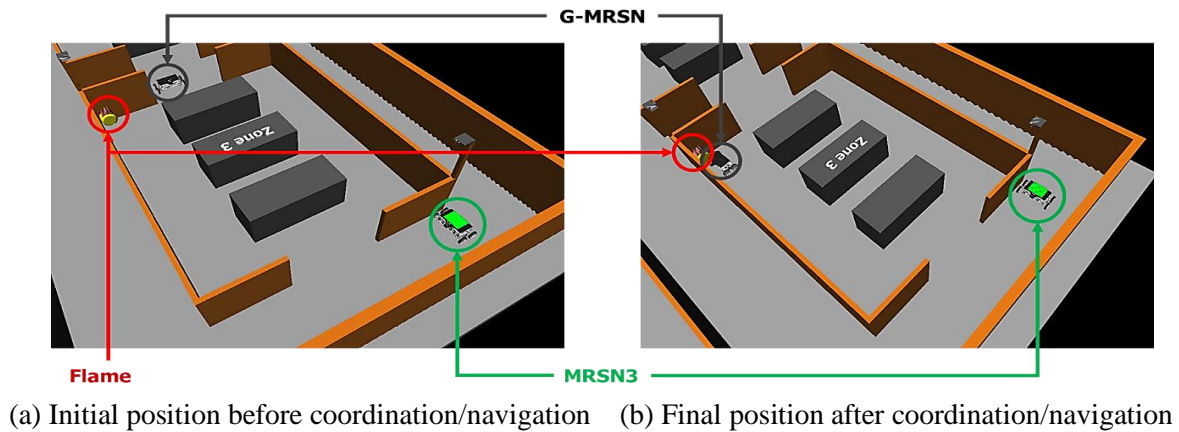


Fig. 5.22. LRES visualization of MRS-SMs navigation: case 3

With the CNP algorithm is in effect between MRSN3 and the G-MRSN, their coordination modes for each MRS-SM is as shown in Fig. 5.23(a) and (b).

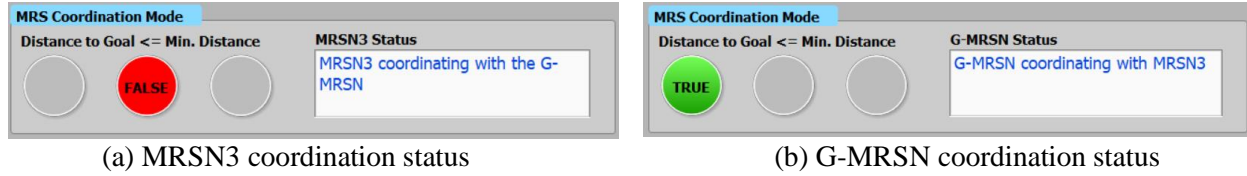


Fig. 5.23. MRS-SM coordination statuses for case 3

So far, the MRS-SMs navigation and visualization using the LRES and MRS GUI has been presented. Their navigation is based on the SSN alert and location, which is contained in the data communicated to them (MRS-SMs) by their O-SSNs in the PCD-SE. On the other hand, the results are communicated by the MRS O-SSNs SMs to the BSG-SM which in turn communicates the data to the SND GUI for local visualization, storage and upload to the MS cloud platform. The BSG also communicate the results to the Matlab ThingSpeak cloud platform.

5.4 IoT BS: Simulation, Results Aggregation, Visualization and Cloud Integration

The BS simulation involves the following components and their roles as illustrated in Fig. 5.23:

- The BSG-SM receive the results of the sensor nodes and communicates them to the SND GUI and ThingSpeak cloud platform. Its test and operation have been presented in section 5.2.
- The SND GUI that aggregates, visualizes, stores the received sensor node results from the BSG. The SND GUI uploads the aggregated results to the MS cloud platform.
- The Cloud Platforms: The MS and ThingSpeak cloud platforms aggregates, stores and analyzes the results data from the sensor nodes.

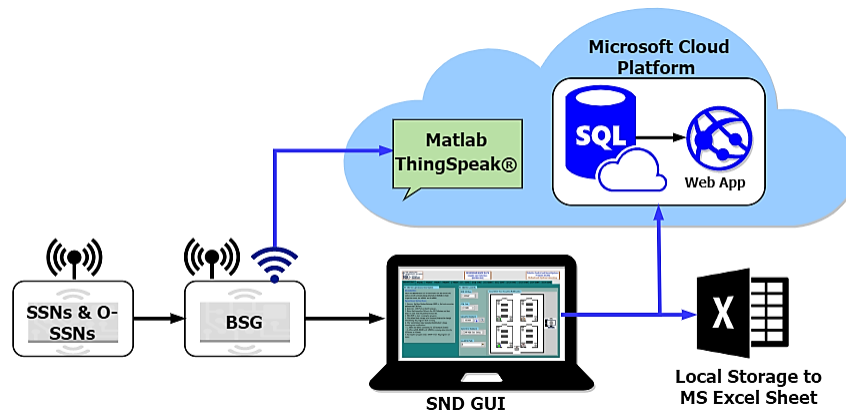


Fig. 5.24. BS simulation process

5.4.1 SND GUI Results Aggregation and Visualization

The SND GUI aggregates, visualizes and stores the sensory information and results from the sensor nodes (SSNs and O-SSNs). They are communicated by the O-SSNs to the BSG, which in turn sends the results to the SND GUI. The SND GUI has an information tab and tabs for each sensor node as described in section 4.4.2. The information tab during the system simulation is shown in Fig. 5.25, where all the sensor node indicators are turned “green” implying that data is being received from each sensor node.

For a sample simulation time of 30 minutes, Fig. 5.26(a) – (e) and Fig. 5.27(a) – (e) shows the visualization of the aggregated sensor information plots for MRSN1 and SSN1 respectively. The temperature, relative humidity (RH) and gas indicators in Fig. 5.26(a) shows the last data received from the sensor node, which is temperature = 24⁰C, RH = 23%, CO gas concentration = 13ppm and smoke concentration = 1018ppm. Based on these detected data the event characterization is “High Alert Event,” since both the smoke concentration, temperature and RH values are indicative of a possible fire scenario according to the fuzzy logic decision system.

Similar to the MRSN1 visualization, the event characterization obtained for the last received data from SSN1 as shown in Fig. 5.27(a) is “High Alert Event” since the temperature and RH values of 24⁰C and 23% respectively are indicative of high alert events.

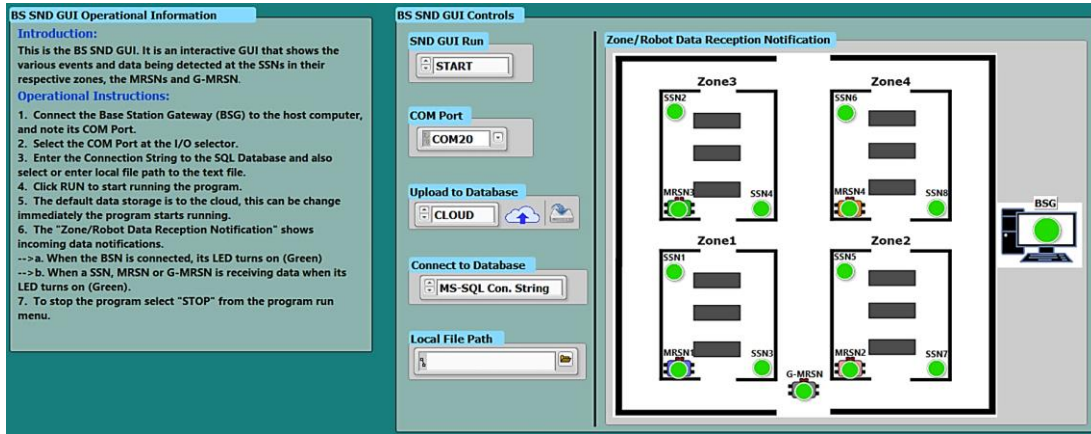
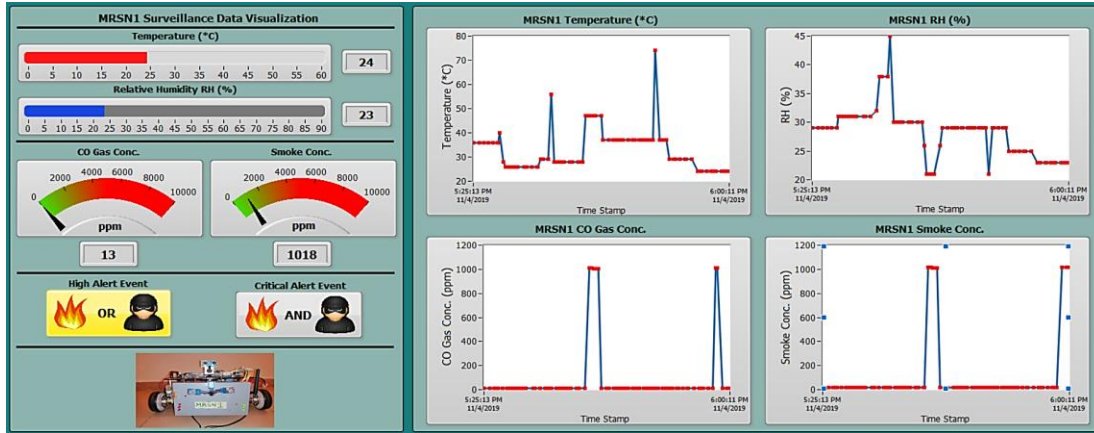
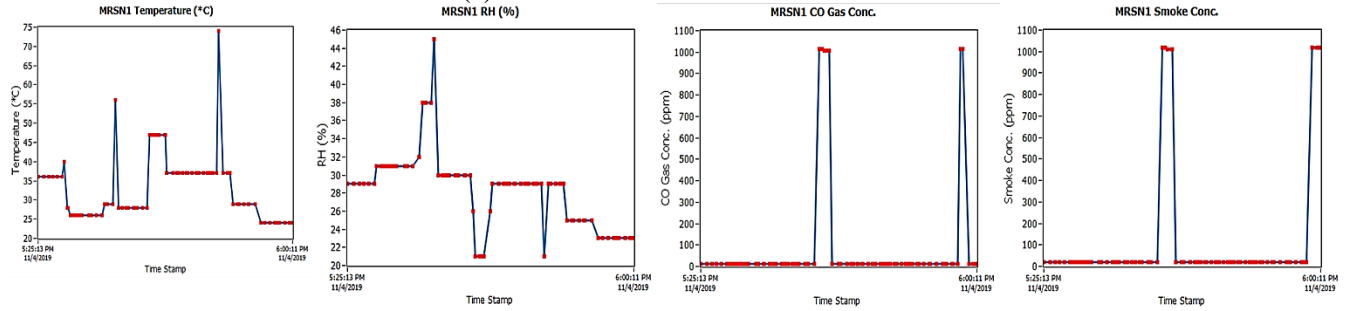


Fig. 5.25. BS SND GUI information tab during simulation



(a) MRSN1 tab visualization



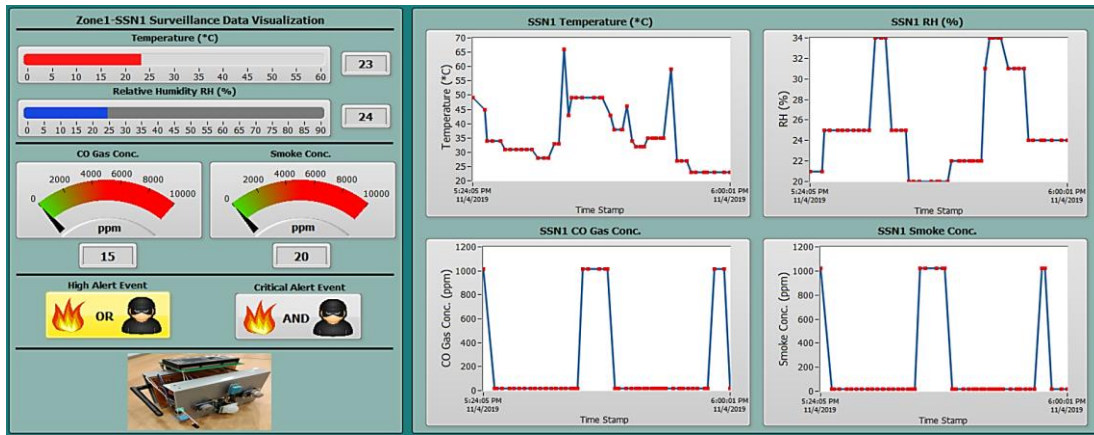
(b) MRSN1 temperature

(c) MRSN1 RH

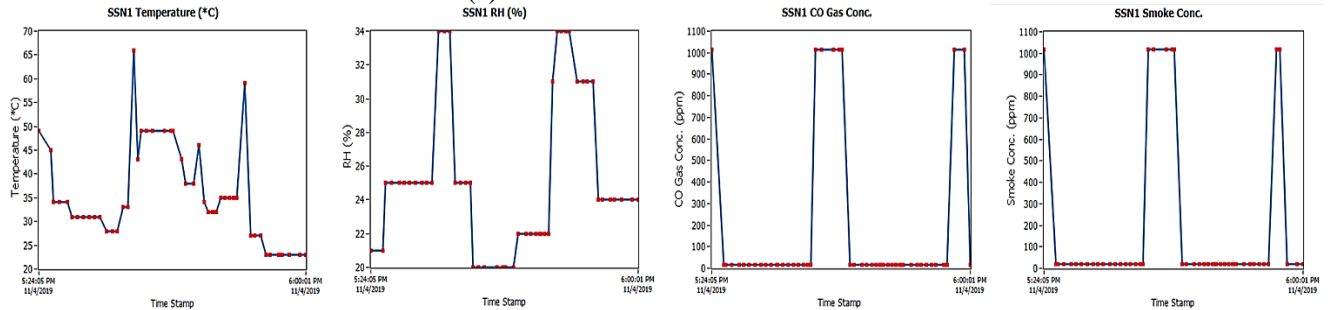
(d) MRSN1 CO gas conc.

(e) MRSN1 smoke conc.

Fig. 5.26. MRSN1 SND GUI visualization



(a) SSN1 tab visualization



(b) SSN1 Temperature

(c) SSN1 RH

(d) SSN1 CO Gas conc.

(e) SSN1 Smoke conc.

Fig. 5.27. SSN1 SND GUI visualization

Similar visualizations were obtained for all the other sensor nodes and robots. From the SND GUI, the aggregated results data is uploaded to the MS cloud platform, while the BSG directly uploads the results data to the ThingSpeak cloud platform.

5.4.2 Cloud Platforms Results Aggregation, Visualization and Analytics

The cloud platforms serve to aggregate, visualize and analyze the sensor node results remotely. There are two cloud platforms as presented in section 4.4.3, the MS cloud and ThingSpeak cloud platforms.

A. MS Cloud Platform Data Visualization and Analytics

The sensory information and results from the sensor nodes are uploaded to the MS cloud platform through the SND GUI. The data is received into a table structured SQL database in the SSMS software as presented in section 4.4.3. Fig. 5.28(a) shows the simulation data visualization in the SQL database table format based on Table 4.5 in section 4.4.3, while Fig. 28(b) shows the results of the simulation data analytics based on Table 4.6 of section 4.4.3.

Results Messages							Results Messages									
	SensorID	SensorName	SensorValueSQ	SensorValueTimeStamp	SensorValue	EventAlert		SensorID	SensorName	SensorTypeID	SensorStats_DataID	SensorID	AverageData	MnData	MaxData	
1	26	SSN1_Temp	1	2019-11-04 13:18:58.673	25	NULL	1	1	G-MRSN_Temp	1	1	1	33.8604651162791	21	49	
2	27	SSN1_RH	2	2019-11-04 13:19:00.560	20	NULL	2	2	G-MRSN_RH	2	2	2	28.5581395348837	21	41	
3	28	SSN1_Gas	3	2019-11-04 13:19:00.873	15	NULL	3	3	G-MRSN_Gas	3	3	3	239.558139534884	7	1007	
4	29	SSN1_Smk	4	2019-11-04 13:19:01.313	20	NULL	4	4	G-MRSN_Smk	4	4	4	199.046511627907	13	1013	
5	30	SSN1_EA	5	2019-11-04 13:19:01.710	0	High Alert	5	5	G-MRSN_EA	5	5	5	0.13953488372093	0	1	
6	51	SSN6_Temp	6	2019-11-04 13:19:08.807	50	NULL	6	6	MRSN1_Temp	6	6	6	35.5441176470588	24	74	
7	52	SSN6_RH	7	2019-11-04 13:19:09.400	20	NULL	7	7	MRSN1_RH	7	7	7	27.0294117647059	21	45	
8	53	SSN6_Gas	8	2019-11-04 13:19:09.973	1015	NULL	8	8	MRSN1_Gas	8	8	8	232.647058823529	13	1013	
9	54	SSN6_Smk	9	2019-11-04 13:19:10.483	1020	NULL	9	9	MRSN1_Smk	9	9	9	193.529411764706	18	1018	
10	55	SSN6_EA	10	2019-11-04 13:19:10.950	0	High Alert	10	10	MRSN1_EA	10	10	10	0.117647058823529	0	1	
11	61	SSN8_Temp	11	2019-11-04 13:19:24.493	48	NULL	11	11	MRSN2_Temp	11	11	11	30.7118644067797	22	46	
12	62	SSN8_RH	12	2019-11-04 13:19:24.897	21	NULL	12	12	MRSN2_RH	12	12	12	28.0677966101695	22	39	
13	63	SSN8_Gas	13	2019-11-04 13:19:25.233	1030	NULL	13	13	MRSN2_Gas	13	13	13	199.966101694915	14	1014	
14	64	SSN8_Smk	14	2019-11-04 13:19:25.623	1040	NULL	14	14	MRSN2_Smk	14	14	14	188.898305084746	20	1020	
15	65	SSN8_EA	15	2019-11-04 13:19:26.043	0	High Alert	15	15	MRSN2_EA	15	15	15	0.11864406779661	0	1	
16	6	MRSN1_Temp	16	2019-11-04 13:19:27.467	35	NULL	16	16	MRSN3_Temp	16	16	16	36.7258064516129	22	96	
17	7	MRSN1_RH	17	2019-11-04 13:19:27.980	26	NULL	17	17	MRSN3_RH	17	17	17	27.6774193548387	20	40	
18	8	MRSN1_Gas	18	2019-11-04 13:19:28.387	13	NULL	18	18	MRSN3_Gas	18	18	18	350.548387096774	13	1013	
19	9	MRSN1_Smk	19	2019-11-04 13:19:28.660	18	NULL	19	19	MRSN3_Smk	19	19	19	323.290322580645	18	1018	
20	10	MRSN1_EA	20	2019-11-04 13:19:29.157	0	High Alert	20	20	MRSN3_EA	20	20	20	0.145161290322581	0	1	
21	36	SSN3_Temp	21	2019-11-04 13:19:35.780	23	NULL	21	21	MRSN4_Temp	21	21	21	35.2162162162162	20	48	
22	37	SSN3_RH	22	2019-11-04 13:19:36.360	21	NULL	22	22	MRSN4_RH	22	22	22	26.7027027027027	21	41	
23	38	SSN3_Gas	23	2019-11-04 13:19:36.747	30	NULL	23	23	MRSN4_Gas	23	23	23	418.540540540541	14	1014	
24	39	SSN3_Smk	24	2019-11-04 13:19:37.170	40	NULL	24	24	MRSN4_Smk	24	24	24	370.27027027027	20	1020	
25	40	SSN3_EA	25	2019-11-04 13:19:37.580	0	High Alert	25	25	MRSN4_EA	25	25	25	0.216216216216216	0	1	

(a) MS SQL database visualization

Results Messages							Results Messages									
	SensorID	SensorName	SensorTypeID	SensorStats_DataID	SensorID	AverageData	MnData	MaxData								
1	1	G-MRSN_Temp	1	1	1	33.8604651162791	21	49								
2	2	G-MRSN_RH	2	2	2	28.5581395348837	21	41								
3	3	G-MRSN_Gas	3	3	3	239.558139534884	7	1007								
4	4	G-MRSN_Smk	4	4	4	199.046511627907	13	1013								
5	5	G-MRSN_EA	5	5	5	0.13953488372093	0	1								
6	6	MRSN1_Temp	6	6	6	35.5441176470588	24	74								
7	7	MRSN1_RH	7	7	7	27.0294117647059	21	45								
8	8	MRSN1_Gas	8	8	8	232.647058823529	13	1013								
9	9	MRSN1_Smk	9	9	9	193.529411764706	18	1018								
10	10	MRSN1_EA	10	10	10	0.117647058823529	0	1								
11	11	MRSN2_Temp	11	11	11	30.7118644067797	22	46								
12	12	MRSN2_RH	12	12	12	28.0677966101695	22	39								
13	13	MRSN2_Gas	13	13	13	199.966101694915	14	1014								
14	14	MRSN2_Smk	14	14	14	188.898305084746	20	1020								
15	15	MRSN2_EA	15	15	15	0.11864406779661	0	1								
16	16	MRSN3_Temp	16	16	16	36.7258064516129	22	96								
17	17	MRSN3_RH	17	17	17	27.6774193548387	20	40								
18	18	MRSN3_Gas	18	18	18	350.548387096774	13	1013								
19	19	MRSN3_Smk	19	19	19	323.290322580645	18	1018								
20	20	MRSN3_EA	20	20	20	0.145161290322581	0	1								
21	21	MRSN4_Temp	21	21	21	35.2162162162162	20	48								
22	22	MRSN4_RH	22	22	22	26.7027027027027	21	41								
23	23	MRSN4_Gas	23	23	23	418.540540540541	14	1014								
24	24	MRSN4_Smk	24	24	24	370.27027027027	20	1020								
25	25	MRSN4_EA	25	25	25	0.216216216216216	0	1								

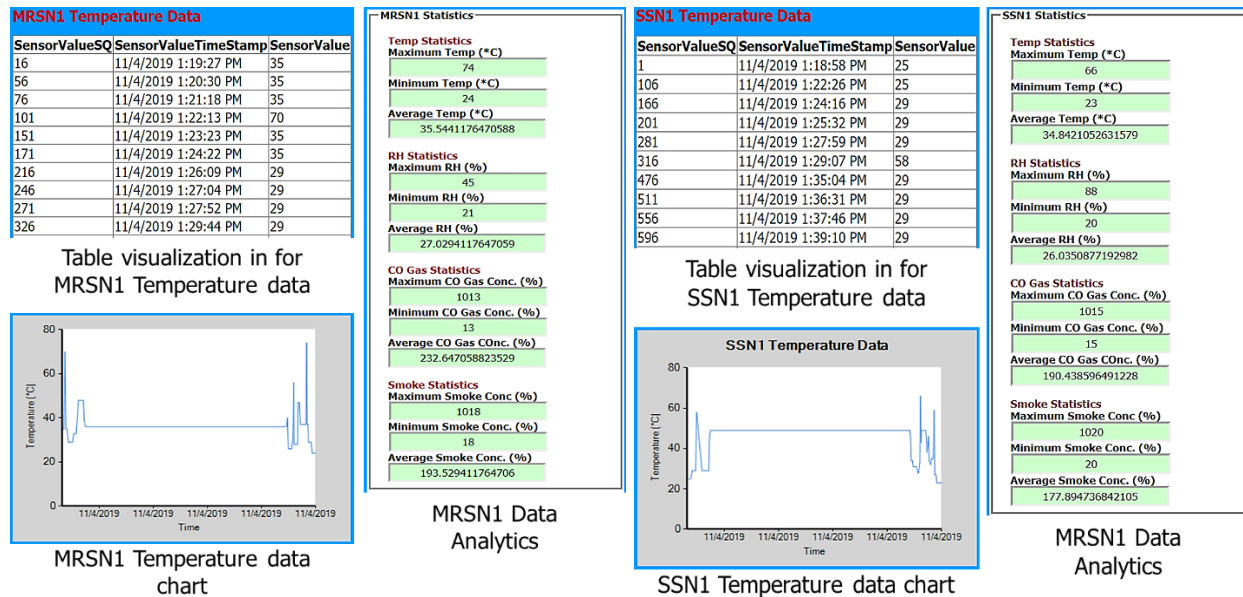
(b) MS SQL database analytics

(a) MS SQL database visualization

(b) MS SQL database analytics

Fig. 5.28. MS SSMS SQL database visualization and analytics

The SSMS is combined with web applications for remote visualization through connection with the internet. The web applications display the results data and analytics for each sensor node in tables, charts and textboxes as presented in section 4.4.3. Fig. 5.29(a) & (b) shows example tables and charts from MRSN1 and SSN1 results, respectively as snippets of the overall application page as shown in Fig. 4.20(a).



(a) MRSN1 web app visualization

(b) SSN1 web app visualization

Fig. 5.29. Web application visualization and analytics

The sensory nodes' information and result data visualization show some outliers that were removed when advanced analytics were performed on the data at the ThingSpeak cloud platform.

B. ThingSpeak Cloud Platform Data Visualization and Analytics

The results from the sensor nodes are uploaded to the ThingSpeak cloud platform through the BSG SM. The data is visualized on the ThingSpeak channel visualization interface for each sensor node as presented section 4.4.3. Fig. 5.30 (a) and (b) shows the simulation data visualization for MRSN1 and SSN1 channels displaying 40 and 60 aggregated data points respectively.

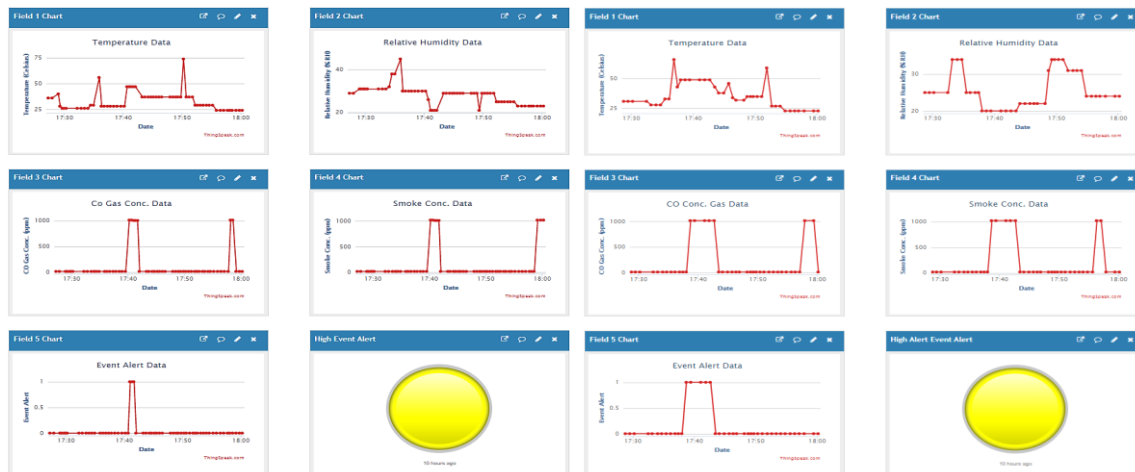


Fig. 5.30. ThingSpeak channel data visualization

ThingSpeak data analytics involved statistical and preservation metrics-based evaluation of the sensory nodes' information from the zones (zones 1 – 4). These analytics are hereby presented.

a. Statistical Analytics

The statistical analytics reports the maximum, minimum and mean values of the Temperature, RH, CO Gas and Smoke sensors associated with each zone. Also, it includes the Temperature/RH regression analysis of each zone. Table 5.3 shows the result of the basic statistical analytics for zones 1 – 4 after preprocessing the sensory data to remove outliers associated with the launching of the simulation environment.

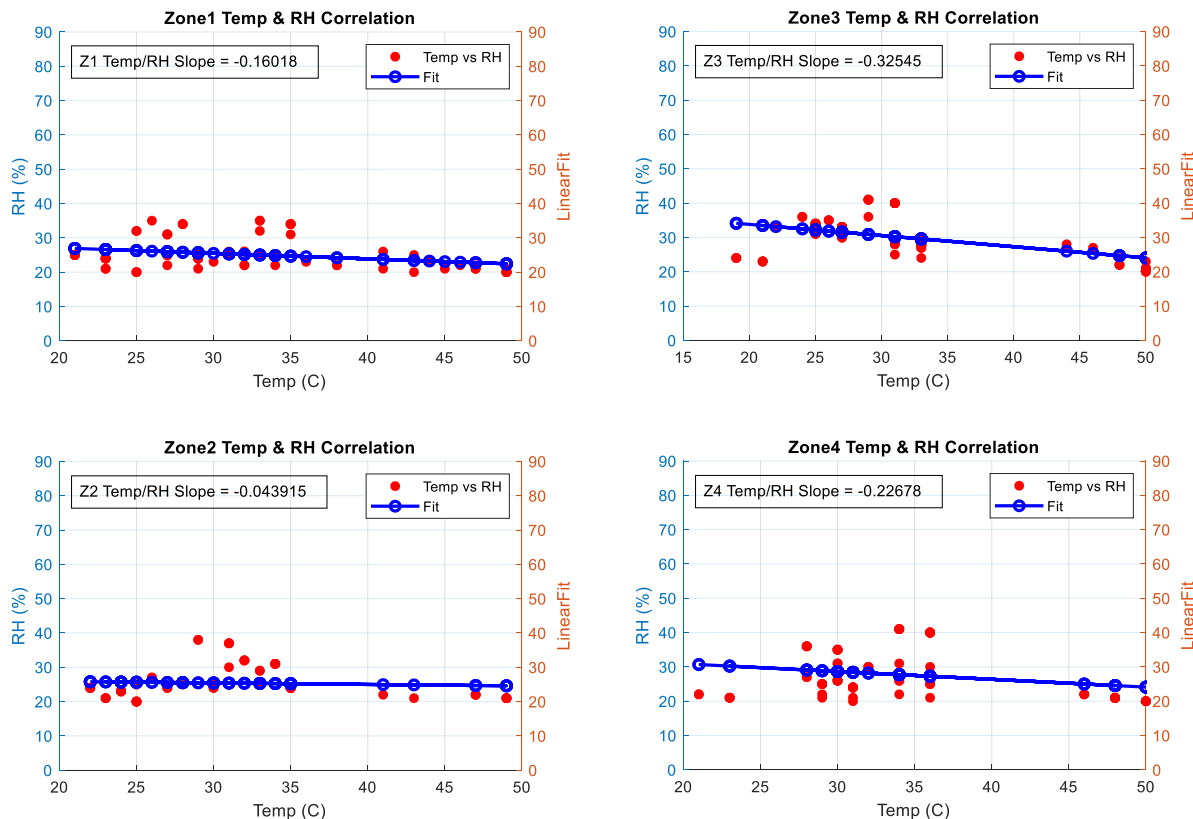
The analytics shows that zone 1 recorded an average temperature of 32.15⁰C slightly higher than zone 2 with 31.15⁰C, while zone 3 recorded 31.05⁰C and that of zone 4 was highest with temperature average equal to 34.44⁰C. The maximum temperature recorded across all the zones were 50⁰C, which is the highest temperature detected by the simulation model of the Temperature/RH sensor DHT-11. The range of RH average is approximately between 25% – 30% across the four zones. The CO gas and smoke concentrations averages are less than 250ppm across the four zones, implying that there were few instances of high combustion gas concentration.

Table 5.3. Statistical data analytics at each Zone

Zone 1 Statistical Data Analytics					
Sensor Variable	Maximum Value	Minimum Value	Mean Value	Maximum Value Timestamp	Minimum Value Timestamp
Temperature (⁰ C)	49	21	32.1522	04-Nov-2019 13:41:23	04-Nov-2019 17:49:33
RH (%)	35	20	25.1304	04-Nov-2019 17:28:49	04-Nov-2019 13:20:02
CO Gas (ppm)	1030	15	151.1413	04-Nov-2019 17:34:22	04-Nov-2019 13:20:02
Smoke (ppm)	1040	20	158.0435	04-Nov-2019 17:34:22	04-Nov-2019 13:20:02
Zone 2 Statistical Data Analytics					
Sensor Variable	Maximum Value	Minimum Value	Mean Value	Maximum Value Timestamp	Minimum Value Timestamp
Temperature (⁰ C)	49	22	31.1481	04-Nov-2019 17:39:16	04-Nov-2019 17:55:05
RH (%)	38	20	25.3827	04-Nov-2019 17:34:04	04-Nov-2019 13:22:07
CO Gas (ppm)	1030	15	168.8889	04-Nov-2019 17:38:32	04-Nov-2019 13:22:07
Smoke (ppm)	1040	20	163.4568	04-Nov-2019 17:38:32	04-Nov-2019 13:22:07
Zone 3 Statistical Data Analytics					
Sensor Variable	Maximum Value	Minimum Value	Mean Value	Maximum Value Timestamp	Minimum Value Timestamp
Temperature (⁰ C)	50	19	31.0494	04-Nov-2019 13:20:42	04-Nov-2019 19:01:21
RH (%)	41	20	30.2099	04-Nov-2019 18:48:18	04-Nov-2019 13:20:42
CO Gas (ppm)	1030	15	156.7284	04-Nov-2019 13:39:59	04-Nov-2019 13:25:06
Smoke (ppm)	1040	20	200.7407	04-Nov-2019 13:39:59	04-Nov-2019 13:25:06
Zone 4 Statistical Data Analytics					

Sensor Variable	Maximum Value	Minimum Value	Mean Value	Maximum Value Timestamp	Minimum Value Timestamp
Temperature ($^{\circ}\text{C}$)	50	21	34.4375	04-Nov-2019 13:20:09	04-Nov-2019 18:55:38
RH (%)	41	20	27.6375	04-Nov-2019 18:44:15	04-Nov-2019 13:20:09
CO Gas (ppm)	1030	15	246	04-Nov-2019 19:04:15	04-Nov-2019 13:26:36
Smoke (ppm)	1040	20	253	04-Nov-2019 19:04:15	04-Nov-2019 13:26:36

The regression analysis shows a decreasing pattern of the temperature and humidity correlation for all zones as shown in Fig. 5.31(a) and (b), respectively. The slopes for the best fit pattern lines between the temperature and RH are presented in Table 5.4; the temperature/RH fitting slope of zone 3 is the steepest with a value of -0.32545. The results show that the Temperature and RH were mostly decreasing with simulation time within the zones. The regression analysis can be used to maintain a specific temperature and humidity within the zones by controlling the conditions to elevate the changing trends and maintain a zero-slope pattern between the temperature and relative humidity.



(a) Zone 1&2 Temperature/RH correlation

(b) Zone 3&4 Temperature/RH correlation

Fig. 5.31. Zones temperature and RH regression analysis

Table 5.4. Zone temperature/RH regression fit pattern line slopes

Zones	Slope of Temperature/RH Best Fit Pattern Line
Zone 1	– 0.16018
Zone 2	– 0.043915
Zone 3	– 0.32545
Zone 4	– 0.22678

b. Preservation Metrics Analysis

Preservation metrics analytics predicts the risk of corrosion for metals (%EMC) and the risk of organic degradation (PI) for organic materials within the zones based on their temperature and relative humidity data. The %EMC and PI metrics are correlated with the dewpoint (moisture content) data for each zone to verify their patterns and relationships since moisture content is one of the leading causes of metal corrosion and organic degradation. Fig. 5.32(a) shows the relationship between the dewpoint with %EMC and PI separately for zone 1 and 2 and Fig. 5.32(b) shows that of zone 3 and 4, respectively, while Table 5.5 shows the slopes of the best fit pattern lines of the comparison between %EMC and PI and dewpoint.

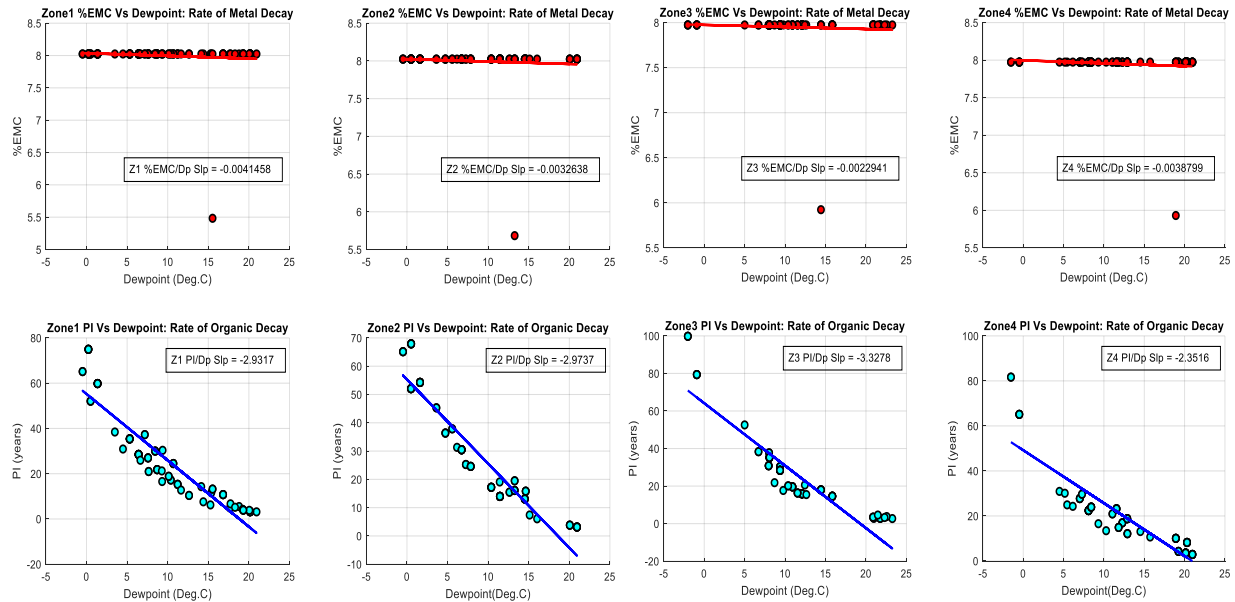


Fig. 5.32. Zones %EMC and PI vs Dewpoint Analytics

Table 5.5. %EMC and PI vs dewpoint best fit pattern line slopes per zone

Zones	Slope of Best Fit Pattern Line	
	%EMC	PI
Zone 1	-0.0041	-2.9317
Zone 2	-0.0033	-2.9737
Zone 3	-0.0023	-3.3278
Zone 4	-0.0039	-2.3516

The outcome of the correlation shows that %EMC is relatively constant as the dewpoint changes across the four zones while the PI is decreasing as the dewpoint is increasing in all four zones. But zone 1 %EMC falls slightly faster than the other three zones since the slope of its best fit pattern line is -0.0041 , while zone 3 PI decreases faster than the other three zones since the slope of its best fit pattern line is -3.3278 . The constant trend of %EMC with dewpoint implies that metals stored within the zones are at a lower risk of corrosion while the decreasing trend of PI with dewpoint means that organic materials stored within the zones are in higher risk of degradation.

The measure of the metallic corrosion (%EMC) and organic degradation (PI) risks are predicted by the use of a neural network (see section 4.4.3) trained on temperature and RH data collected from each zone. The neural network developed for each zone has two inputs (temperature and RH) and two targets (%EMC and PI) from the combination of the inputs and targets from each sensor node within the zone. The sensor nodes' inputs and targets are combined because they are located within the same zone space; also the %EMC and PI are recommended to be predicted based on the running average of the temperature and relative humidity within a given space [147], [151]. The neural network training inputs/targets data per zone is illustrated to show zone 1 as an example in Fig. 5.33(a), while Fig. 5.33(b) shows the %EMC and PI prediction based on the average temperature and relative humidity.

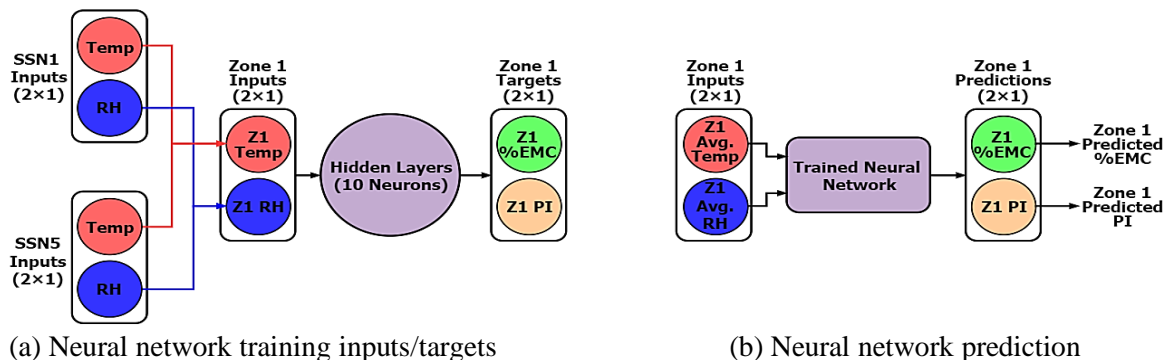


Fig. 5.33. Neural network data inputs/targets for zone 1

Table 5.6 presents the sizes of the data obtained through simulation as inputs to the neural network for each zone.

Table 5.6. Neural network data per zone

	Neural Network Data	
Zones	Inputs	Targets
Zone 1	2×92	2×92
Zone 2	2×81	2×81
Zone 3	2×81	2×81
Zone 4	2×80	2×80

80% of the data obtained through simulation was used for training, 10% for test, and 10% for validation.

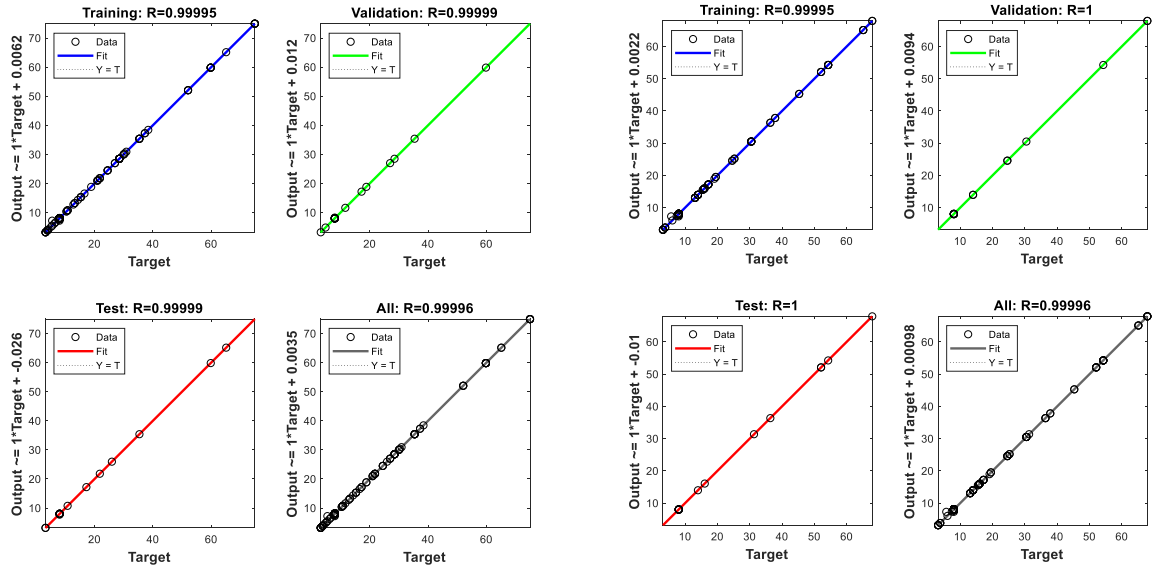
The performance of the neural network at each zone can be measured using regression analysis. The regression analysis measures the correlation between the predicted response (output) and the targets of the network. The results of the correlation are represented by a correlation coefficient called the R-value. When R-values representing the correlation between the predicted network output and targets approaches 1.0, it indicates a good correlation, and this implies that the network is well trained. The neural network for each zone is expressed by equation (5.1).

$$Y = (W \times T) + b \quad (5.1)$$

Where Y = the predicted outputs vector, W = weights vector, T = targets vector and b = biases vector. W and b are vectors are determined by the Levenberg-Marquardt (LM) training method expressed in equation (4.8) of section 4.4.3. The predicted outputs obtained during the training, testing and validating of the network, and that obtained after an overall test on the network are used to calculate the regression. Based on equation (5.1), the R-value is calculated from the predicted outputs and targets using the equation (5.2) for each of the mentioned cases.

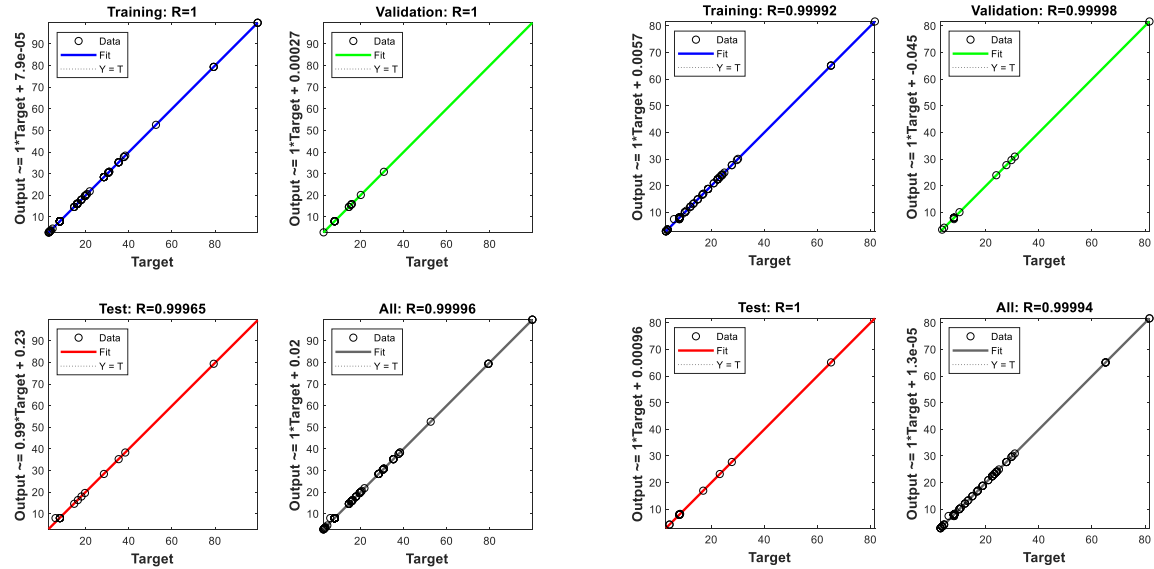
$$R - value = \frac{\sum (T_i - \bar{T})(Y_i - \bar{Y})}{\sqrt{\sum (T_i - \bar{T})^2 \sum (Y_i - \bar{Y})^2}} \quad (5.2)$$

The plots of all obtained R-values of the neural network at each zone is shown in Fig. 5.34(a) – (d). Each plot shows the R-value for training, test, validation, and that obtained using random data across the dataset. The obtained R – values and network training time for each zone are presented in Table 5.7.



(a) Zone 1 neural network regression analysis

(b) Zone 2 neural network regression analysis



(c) Zone 3 neural network regression analysis

(d) Zone 4 neural network regression analysis

Fig. 5.34. Zones' neural network regression analysis.

Table 5.7. Zone neural network regression analysis R – values

Zones	Correlation Coefficient (R – Values)			
	Training	Validation	Test	Using All Data
Zone 1	0.99995	0.99999	0.99999	0.99996
Zone 2	0.99995	1.0	1.0	0.99996
Zone 3	1.0	1.0	0.99965	0.99996
Zone 4	0.99992	0.99998	1	0.99994
Network Training Time (s)				
Zone 1 – 4		< 5		

From the R – values presented in Table 5.7 and the regression plots in Fig. 5.34, it can be deduced that there is a good fit between the outputs (predicted %EMC and PI) and targets (generated %EMC and PI) for each zone since all the R-values are close to 1.0. This indicates that the network is well trained on the given data.

To predict the measure of risk associated with a zone in terms of %EMC (for metallic corrosion) and PI (organic degradation), the actual running averages of the temperature and RH from each zone are inputted into the trained neural network. The values presented in Table 5.8 shows the prediction results of the neural network for each zone evaluated based on their respective risk levels as presented in Table 4.7 for %EMC and Table 4.8 for PI in section 4.4.3.

Table 5.8. %EMC and PI predictions per zone

% EMC Predictions per Zone				
Zones	%EMC	Status/Color Code	Metal Corrosion Risk	Average Dewpoint
Zone 1	8.0355	OK	Limited risk of corrosion	9.5106
Zone 2	8.1334	OK	Limited risk of corrosion	8.8515
Zone 3	7.9738	OK	Limited risk of corrosion	11.2272
Zone 4	7.9728	OK	Limited risk of corrosion	12.6425
PI Predictions per Zone				
Zones	PI (years)	Status/Color Code	Organic Degradation Risk	Average Dewpoint
Zone1	18.9409	RISK	Accelerated rate of organic decay	9.5106
Zone 2	21.1765	RISK	Accelerated rate of organic decay	8.8515
Zone 3	19.0098	RISK	Accelerated rate of organic decay	11.2272
Zone 4	13.5477	RISK	Accelerated rate of organic decay	12.6425

The prediction values for %EMC shows limited metal corrosion risk for all zones, while the prediction values for PI show high organic degradation risk for all zones. From the results, it can be deduced that the zones are favorable to store metallic materials more than organic materials if temperature and RH within the zones remain relatively constant. Also, an organic material will start degrading after approximately 19 years in zone 1, 21 years in zone 2, 19 years in zone and 14 years in zone. These analytics can help determine the storage time, and type of material to be stored within each zone.

5.5 Hardware in the Loop (HIL) based Simulation

The developed system architecture is featured by HIL-based simulation whereby all modules such as sensor nodes and robots can be incorporated into the system simulation environment individually supporting application development by operating, testing and validating the hardware

modules before full hardware implementation. The roles of HIL based simulation and advantages, setup and results are hereby presented.

5.5.1 Role and Advantages of HIL based Simulation

HIL based simulation has the following roles in system development [161]:

- a. Provides a means for system testing and validation.
- b. Enables module-based system verification, ensuring that the development algorithms meet the system requirements.
- c. Allows for verification of system resources, to ensure meeting system requirements.

Based on this role, HIL based simulation has the following advantages to the overall system development [162]:

- a. Evaluates system performance with actual hardware components
- b. Facilitates the test and development of system algorithms in a more realistic environment.
- c. It helps to identify system errors and bottlenecks that could reduce the system performance.
- d. Increases reliability and defines the success probability of the overall system.
- e. Reduces development cost and in some cases, health damages on the system users.
- f. Reduces system design to implementation time.

5.5.2 HIL based Simulation: The Setup

According to the developed system architecture described in section 4.1, the HIL based simulation setup supporting the development of the ANR system is shown in Fig. 5.35, illustrating how the hardware and simulation modules are integrated to support the system development, validation and testing. As shown in Fig. 5.35, the hardware and simulated modules communicate their results through their various networks, i.e. from SSNs to MRSN O-SSNs and MRSN O-SSNs and then to BSG. At the base station, the hardware BSG connects to the simulated BSG in the SE, forming the HIL BSG, whereby both the hardware and simulated results are then aggregated and visualized in the SND GUI and uploaded to the cloud platforms.

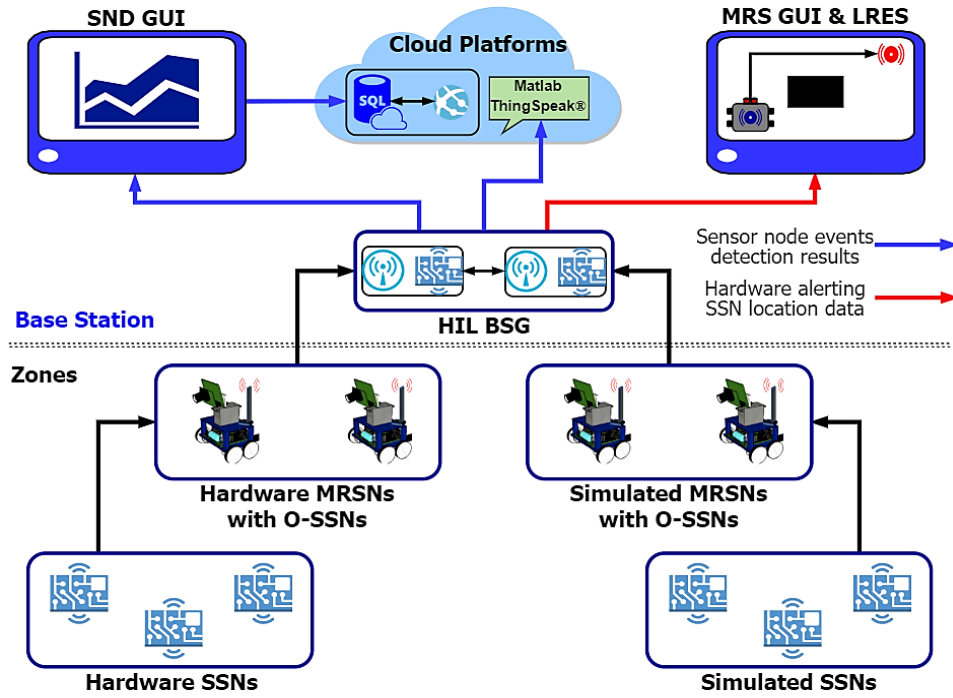


Fig. 5.35. HIL based Simulation Setup

The following steps illustrate the insertion of a zones' hardware (sensor nodes and the related mobile robot) and the BSG hardware into the SE.

How to insert a hardware sensor node into the simulation environment?

- The simulated models of the target sensor node are deactivated from the SE. This is done by going to the network simulator (emulator) interface and deactivating the com ports related to these nodes.
- The hardware sensor node is then switched to begin its sensing and communication operations as shown in SSN1 and SSN5 hardware examples in Fig. 5.36(a) and (b).



(a) SSN1 hardware module

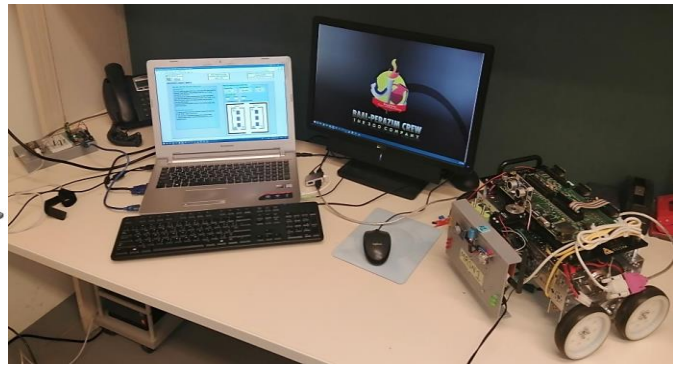
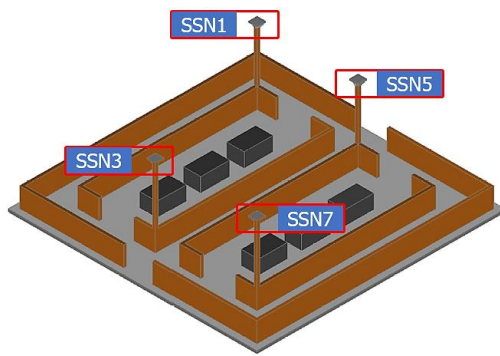


(b) SSN5 hardware module

Fig. 5.36. Hardware sensor nodes during HIL based simulation

How to insert the hardware mobile robot.

- a. The hardware robot is switched on to receive data from the hardware sensor nodes and communicate them to the hardware BSG.
- b. The simulated mobile robot is not deactivated since it will continue to represent the hardware robot within the SE.
- c. Since the simulated mobile robot is not deactivated, a simulated navigation environment reflective of the actual lab environment is designed and imported into the MRS/LRES SE. The navigation environment is shown in Fig. 5.37(a)
- d. The hardware robots are then connected to the MRS/LRES SE by as shown in Fig. 5.37(b)



(a) Lab Experimental NE CAD Model (b) MRSN1 connection to MRS GUI/LRES SE
Fig. 5.37. MRS hardware during HIL based Simulation

How to insert the hardware BSG

- a. The hardware BSG and simulated BSG work together. Hence, it is switched on and connected to the simulated BSG, forming the HIL BSG.
- b. The HIL BSG then connects to the SND GUI and ThingSpeak cloud platform through its other com ports.
- c. The other simulated zones remained connected within the PCD-SE.
- d. After all the connections and networks have been established, the simulation is started for the simulation models and the whole HIL based simulation starts running and aggregating both hardware and simulation results, as shown in Fig. 5.38.

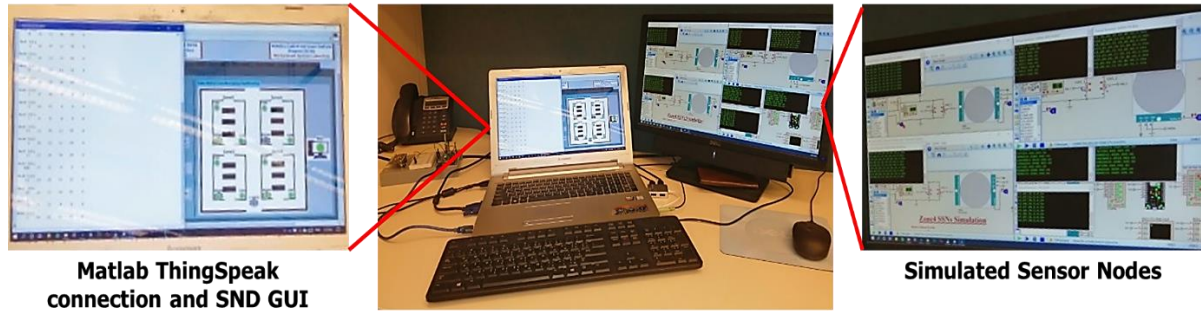
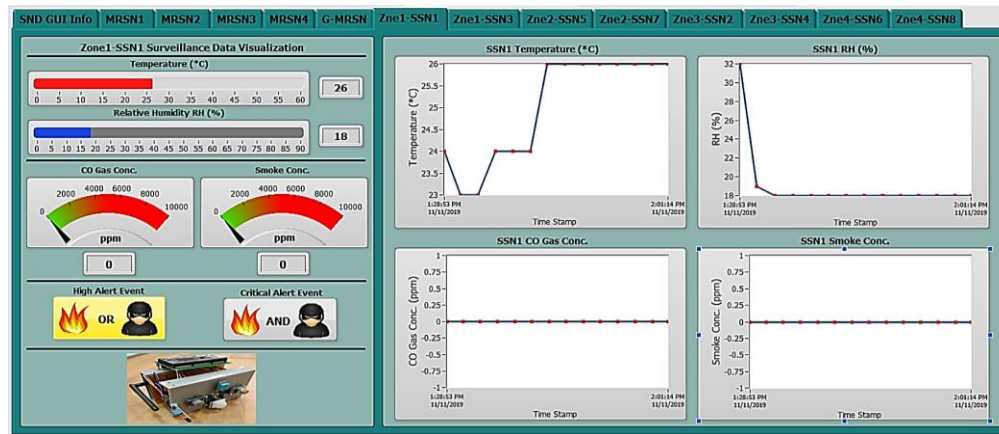


Fig. 5.38. BS during HIL based Simulation

5.5.3 HIL Based Simulation: The Results

The following results show example sensory information and results of SSN1 hardware module during HIL based simulation compared to the same sensor node when simulated. Fig. 5.39(a) – (e) shows hardware SSN1 results from the HIL based simulation, while Fig. 5.40(a) – (d) shows simulated SSN1 results from the simulation only test. The simulation only and HIL based simulation results for SSN1 are displayed for the purpose of comparison.



(a) SSN1 tab interface

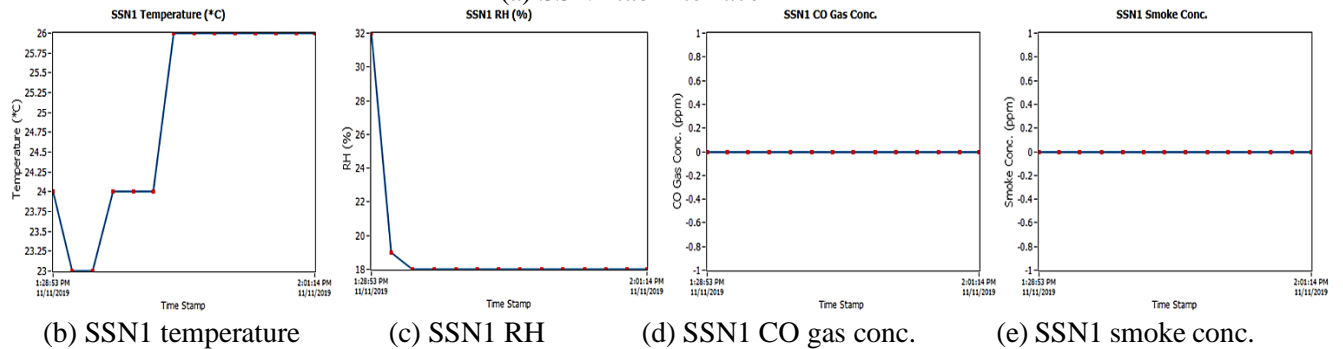


Fig. 5.39. SSN1 hardware module results



(a) SSN1 Tab Interface

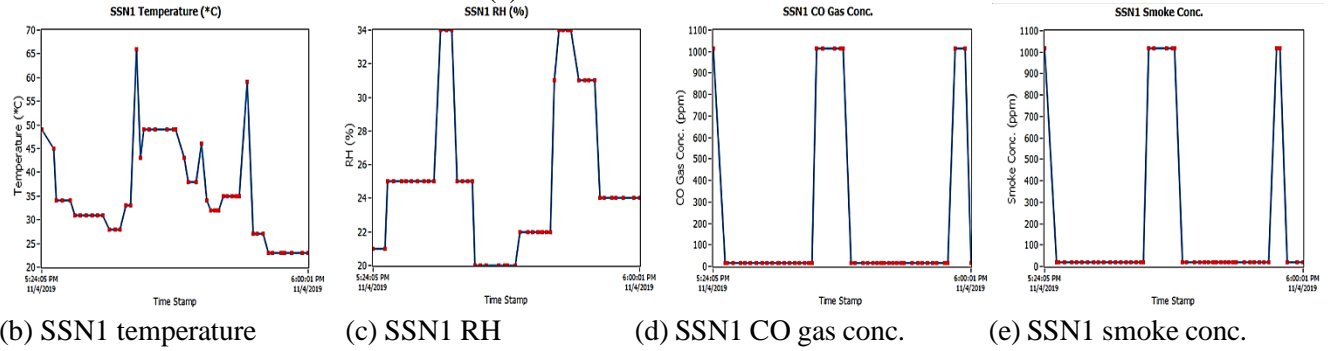


Fig. 5.40. SSN1 simulation model results

Comparing the results shown in Fig. 5.38 and Fig. 5.39, the SSN1-SM records more data than the SSN1 hardware module, while also recording some outlier data. The difference in data recorded between the simulated and hardware SSN1 is because the hardware gas sensors require calibration, and specifically the CO Gas sensor requiring a total of 2.5 minutes to acquire its data. Hence, that time became the interval at which the whole hardware sensor node data is communicated. The simulated model records outlier data when the models are being executed at a low host PC CPU processing speed. All the events recorded by the hardware sensor node are ‘High Alert’ events due lab temperature/relative humidity conditions and the movement of the user within the lab environment.

For the robot HIL based simulation, Fig. 5.41 shows MRSN1-SM example, whereby Fig. 5.41(a) shows the event notification on the MRS GUI, while Fig. 5.41(b) shows the navigation path generated to the alerting SSN, in this case, SSN3. The coordinates of SSN3 within the lab environment is (2.5, 1.5), while the initial position of MRSN1 is (1.5, 1.5).

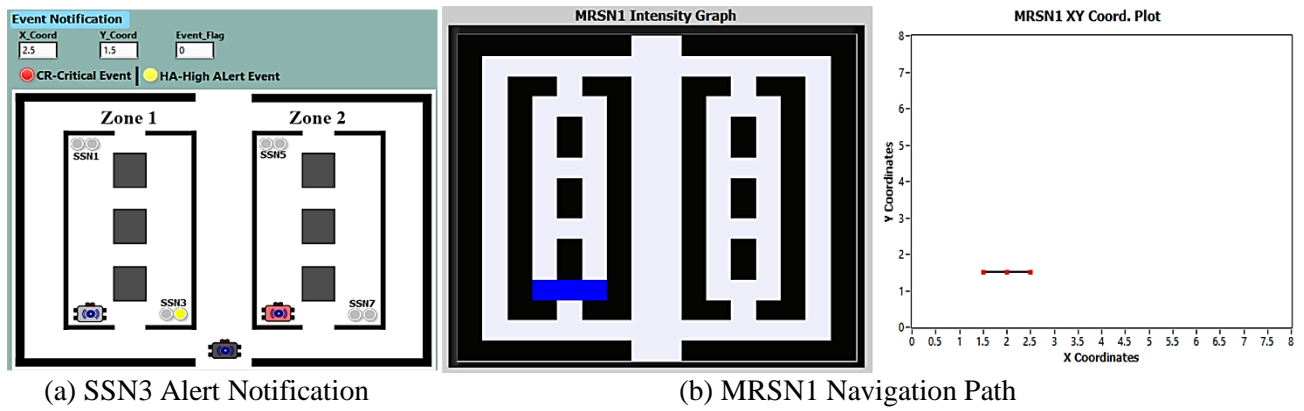


Fig. 5.41. MRSN1-SM Navigation

The navigation of MRSN1-SMs within the lab environment model NE as visualized in the LRES is shown in Fig. 5.42, where Fig. 5.42(a) shows MRSN1-SM at its initial position before navigation and Fig. 5.42(b) shows MRSN1-SM at SSN3 location after navigating the path. The coordination status of MRSN1 is shown in Fig. 5.43.

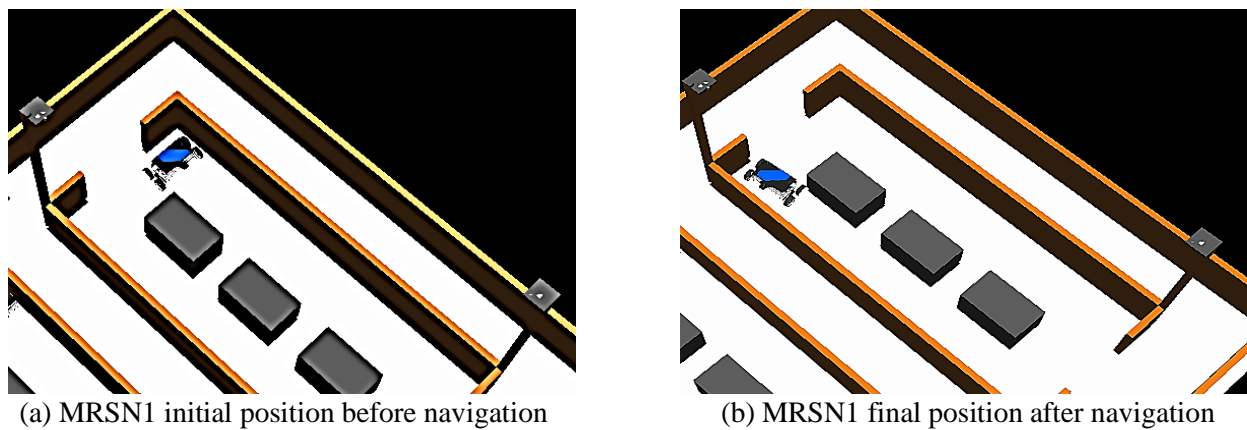


Fig. 5.42. LRES visualization of MRSN1 navigation during HIL based simulation

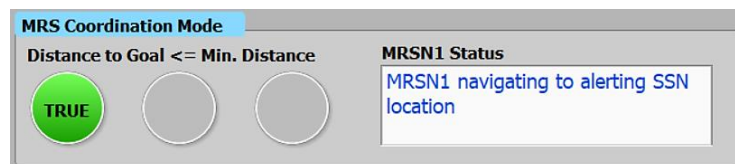


Fig. 5.43. MRSN1-SM coordination status during HIL based simulation

Chapter 6

IoT BASED HYBRID ANR SYSTEM IMPLEMENTATION

This chapter presents the hardware implementation of the developed IoT based hybrid ANR system. This involves the selection, integration and testing of all hardware components making the system and then the whole system is tested in a lab environment. The hardware includes the WSNs, MRS, BS with all its sensor nodes and navigation support.

6.1 WSN Hardware Components and Implementation

The design of each sensor node structure introduced in section 4.2.1 and 4.3.1 comprises the following hardware components to enable the developing of the zone SSNs and O-SSNs.

- a. Microcontroller board for data processing,
- b. Sensors for event detection,
- c. RF radio modules for data communication and
- d. Power source supporting the node operation.

6.1.1 The Microcontroller (MCU) Board

The hardware of an SSN data processor in relation to the selected application should have the following requirements:

- a. Data processing capabilities for analyzing sensory data with a minimum of 4MHz clock speed.
- b. 3 digital inputs for the digital sensors.
- c. 8 digital outputs: 2 for the green and red LEDs used as indicators and 6 to support information display on the LCD.
- d. A/D converters of at least 2 channels to connect the analog sensors.
- e. A minimum of 16kb memory for programming
- f. 1 serial com port to integrate the ZigBee RF module that facilitates communication to the robot nodes.

Multiple MCU boards with PIC, AVR, AMR, etc. processors are available in the market that fulfilled the data processor requirements for the SSNs. Therefore, the AVR based Arduino Uno MCU board [163] shown in Fig. 6.1 was selected, and it has the following features:

- a. ATmega328P processor with 16MHz clock speed.

- b. 16 programmable digital I/O ports (with 6 PWM embedded ports).
- c. ADC with 6 channels and 10-bit conversion resolution.
- d. 32kb memory size for programming and flash storage.
- e. 1 hardware serial com port and an internal software serial capability on any digital port.



Fig. 6.1. Arduino Uno MCU board [163].

The hardware of an O-SSNs data processor in relation to the selected application have similar requirements to the SSN data processor except that the hardware O-SSNs requires more communication ports. The following communication ports are required in the O-SSNs:

- a. 1 serial com port to integrate the ZigBee module for communication with the SSNs.
- b. 1 serial com port to integrate the wireless serial module for communication with the BSG.
- c. 1 serial com port for serial connection to robot's controller.

Based the O-SSNs extra processor requirements, the Arduino Mega MCU [164] board shown in Fig. 6.2 was selected, and it has the following features:

- a. Atmega 2560 processor with 16MHz clock speed.
- b. 54 programmable digital I/O ports (with 14 PWM embedded ports).
- c. ADC with 16 channels and 10-bit conversion resolution.
- d. 256kb memory size for programming and flash storage.
- e. 4 hardware serial com ports and an internal software serial capability on any digital port.



Fig. 6.2. Arduino Mega MCU board [164]

6.1.2 The Sensors

The sensors are selected based on the application and the type of events to be detected. Hence, for fire and intruder events, the following sensors are selected: Temperature/RH, CO Gas, Smoke, Flame and Motion sensors. The required sensor and the selected hardware features are presented in Table 6.1.

Table. 6.1. Required sensor features and selected sensors

S/N	Sensor	Features	Selected Hardware
1	Temperature/RH	<ul style="list-style-type: none"> ▪ RH range: 20 – 90% with an accuracy of $\pm 5\%$ ▪ Temperature range: 0 – 50°C with accuracy of $\pm 2^{\circ}\text{C}$. ▪ Digital Output 	Digital Humidity & Temperature Sensor (DHT – 11) [124]
2	CO Gas Sensor	<ul style="list-style-type: none"> ▪ Metal-Oxide semiconductor (MOS) based sensor. ▪ Sensitivity range: 0 – 10,000ppm. ▪ Analog/Digital output 	MQ-7 CO gas sensor [127]
3	Smoke Sensor	<ul style="list-style-type: none"> ▪ Metal-Oxide semiconductor (MOS) based sensor. ▪ Sensitivity range: 0 – 10,000ppm. ▪ Analog/Digital output 	MQ-2 Smoke sensor [127]
4	Flame Sensor	<ul style="list-style-type: none"> ▪ IR receiver wavelength range: 760 – 1100nm. ▪ Field of View (FOV): 60° ▪ Detection range: 1m. ▪ Analog/Digital output. 	IR Flame Sensor [165]
5	Motion sensor	<ul style="list-style-type: none"> ▪ Pyroelectric Infrared (PIR) based. ▪ Field of View (FOV): $110^{\circ} \times 70^{\circ}$ ▪ Detection range: 7 – 12m. ▪ Digital output 	PIR Motion Sensor [114]

The hardware of the selected sensors are shown in Fig. 6.3(a) – (e).

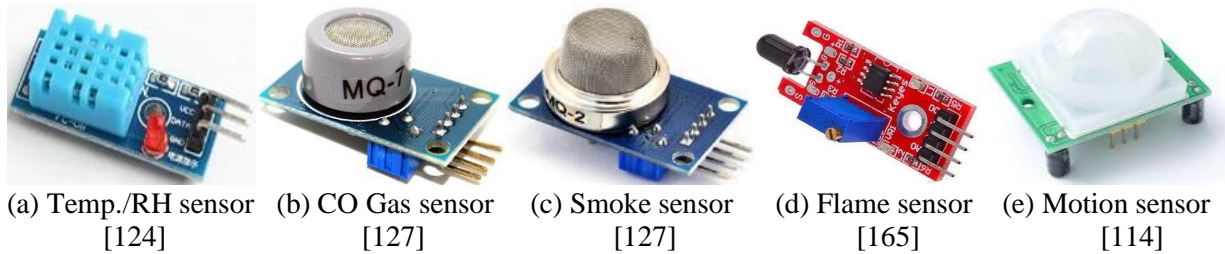


Fig. 6.3. WSN Sensors

6.1.3 The RF Communication Modules

The RF modules are used for data communication between the sensor nodes and the zone robot through its O-SSN and they are required to have the following features:

- Ability to form ad-hoc networks immediately they are connected to the sensor nodes.
- Easy to interface with the MCU boards for data transmission/reception.

- c. Point-to-point or broadcast communication modes
- d. Have a minimum of 10m communication range.

Based on the RF modules requirements, the selected communication modules include the ZigBee wireless UART (2.4GHz) for SSNs to MRSN O-SSNs communication and the Wireless Serial (UART - 433MHz) modules for MRSN O-SSNs to BSG communication. Their features are presented in Table 6.2. Note that the O-SSN can connect directly to the MRSN controller.

Table. 6.2. RF communication modules connections and features

RF Communication Module	Connection	Features
ZigBee Wireless UART Module (2.4 GHz) [166]	SSNs-to-MRSN O-SSNs	<ul style="list-style-type: none"> ▪ Based on TI CC2530 chip with IEEE 802.15.4 protocol. ▪ Plug and play. ▪ Baud rate range: 2400 – 115200bps, 16 channels. ▪ Two com. modes: point-to-point and broadcast. ▪ Maximum transmission range: 250m outdoor
Wireless Serial (UART – 433MHz) Module [86]	MRSN O-SSNs-to-BSG	<ul style="list-style-type: none"> ▪ Based on TI C1101 chip. ▪ Plug and play ▪ Baud rate range: 1200 – 115200bps, 20 channels. ▪ Broadcast com. mode. ▪ Transmission range 40m (indoor), 100m (outdoor).

The selected communication modules are shown in Fig. 6.4(a) and (b).



(a) ZigBee wireless UART (2.4GHz) module [166] (b) Wireless serial (UART – 433MHz) module [86]

Fig. 6.4. RF communication modules.

6.1.4 The Sensor Nodes

The assembled SSN and the MRSN O-SSN are shown in Fig. 6.5(a) – (b). Both sensor nodes are powered by 5V or 12V DC power sources.



(a) SSN



(b) MRSN2 O-SSN

Fig. 6.5. WSN hardware sensor nodes

6.2 MRSN Hardware Components

The MRSN hardware components comprises the following components

- a. The mobile robot platform
- b. Wi-Fi router
- c. O-SSN

6.2.1 The Mobile Robot Platform

The mobile robot hardware platform for MRSNs should have the following requirements:

- a. A controller to enable programming the mobile robot and further functional development.
- b. Sensors for obstacle detection, avoidance, and position tracking to support navigation.
- c. Physical characteristics such as for locomotion and body frame structure to support movement and attachment of other supporting hardware.
- d. Serial com port for connection to the MRSN O-SSNs.
- e. Ethernet connection ports for connection from the robot to the BS to report its location.

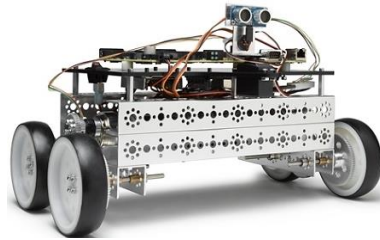
Based on the mobile robot platform requirements, the DaNI 1.0 mobile robot platform [160] shown in Fig. 6.6(a) was selected. The main features of DaNI 1.0 mobile robot platform includes [160]:

- a. A single board reconfigurable input/output (sbRIO) controller platform. The controller platform integrates a reconfigurable FPGA, real-time processor and digital/analog I/Os.
- b. Two 12V DC motors (left and right) controlling the four wheels. The motors are rated 152rpm for speed and 300 oz-in of torque.
- c. A Pitsco body frame supporting the attachment of other hardware
- d. Optical quadrature encoders rated at 400 pulses per revolution

- e. An ultrasonic sensor mounted on an 180° sweep servo motor, for distance measurements between 2cm and 3m.
- f. 1 RS232 port for serial connection to external hardware
- g. 1 Ethernet connect port for wired or wireless connection to host PC.

6.2.2 The Wi-Fi Router

The Wi-Fi router enables the wireless connection of the mobile robot platforms to the host PC. The selected Wi-Fi router is shown in Fig. 6.6(b).



(a) DaNI Robot 1.0 [160]



(b) Wi-Fi router [167]

Fig. 6.6. MRS hardware components

6.2.3 The MRSN

The assembled mobile robot with the O-SSN is shown in Fig. 6.7(a) featuring the O-SSN (front view) and Wi-Fi router (back view) for connection to the BS. The DaNI 1.0 robots communicate with the host PC of the BS via an Ethernet cable by default. To establish a wireless connection between the host PC and the robot, the Wi-Fi router is used. The robot connects to the router via the Ethernet port using its unique local IP address (for example 169.254.62.15) and a general subnet address. The router is then connected to the host PC on the same subnet address, but with another IP address set at the host PC. This enables the wireless connection between the robot and the host PC since they share the same subnet addresses. Hence, the robot paths and locations are communicated to the host PC of the BS through the established wireless connection. On the other hand, the O-SSNs communicates the sensory information from zones to the BS through its wireless serial modules. Fig. 6.7(b) illustrates the robot – BS connections and data communication.

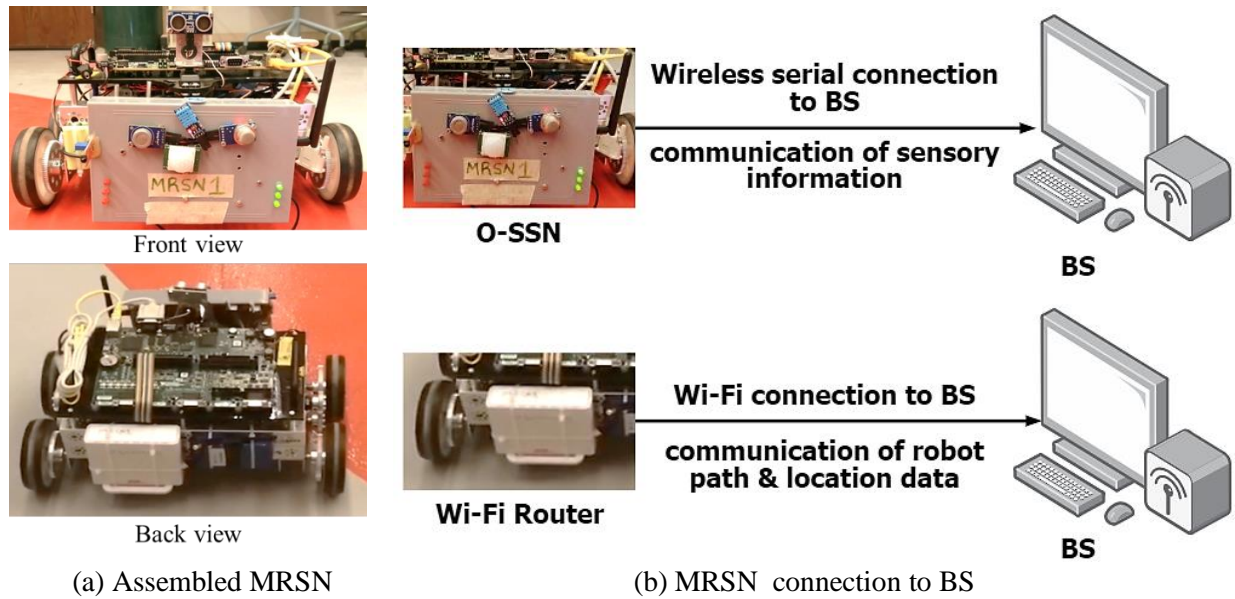


Fig. 6.7. Assembled MRSN and connection to BS

6.3 IoT BS Hardware Components

The BS hardware components comprises the following:

- MCU board for processing received data from O-SSNs and sending it to the host PC
- Wi-Fi programming board to connect to ThingSpeak cloud platform.
- Communication module for connection between O-SSNs, BS and the host PC.
- The Host PC for operating the BS GUIs and enabling connection to the cloud platforms.

The MCU board, Wi-Fi programming module and communication together constitute the BSG that receives sensory information from the O-SSNs. The Arduino Uno MCU board presented in section 6.1.1 was selected as the BSG MCU board, while the Wireless Serial (UART – 433MHz) module presented in section 6.1.3 was selected as the BSG communication module. The Wi-Fi programming modules and host PC are hereby presented.

6.3.1 The Wi-Fi Programming Module

The following are the requirements for the Wi-Fi modules that connect directly from the BS to ThingSpeak cloud:

- A Wi-Fi chipset for connection to ThingSpeak cloud through the Internet.
- Minimum of 16kb memory for programming.
- 1 Serial communication port to connect for interfacing with Host PC during board programming.

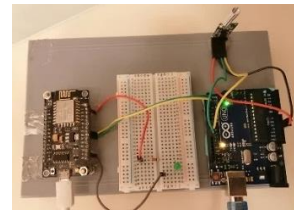
- d. 2 digital lines for software serial communication with the BSG MCU board.

Based on the listed requirements, the NodeMCU Wi-Fi programming module [168] shown in Fig. 6.8(b) was selected. Fig. 6.8(b) shows the assembled BSG. The selected NodeMCU Wi-Fi programming module has the following features:

- a. On-board ESP8266-12E module with PCB antenna for Wi-Fi connection.
- b. 4MB memory size for flash storage and programming.
- c. 1 serial com port (USB-TTL) for plug & play connections.
- d. 10 digital, analog and I/O ports.



(a) NodeMCU Wi-Fi programming board [168]



(b) BSG

Fig. 6.8. NodeMCU Wi-Fi programming board and assembled BSG

6.3.2 The Host PC

The host PC contains the SND GUI, MRS GUI and enables the visualization and analytics of the aggregated results on the cloud platforms. The host PC should have the following capabilities:

- a. Serial and Ethernet connection ports.
- b. Connection to the Internet to facilitate integration of the cloud platforms (MS and ThingSpeak).
- c. Minimum of 8G RAM, 2GHz processing speed and 500GB storage.

Based on the mentioned requirements a Lenovo® intel core i5 2.20GHz, 8GB RAM and 1TB PC was selected. The host PC with the connection the BSG is shown in Fig. 6.9.

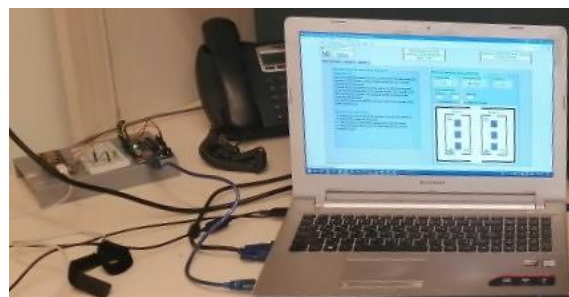


Fig. 6.9. Host PC connected to BSG

6.3.3 Comparison between the ANR IoT Gateway and the Research/Commercial Gateways

The use of gateways in IoT systems has the following key advantages [169], [170]:

- Aggregates and process data from the SSNs and O-SSNs and communicate the data to SND GUI for visualization and cloud platforms for storage and analysis.
- Eliminates the need for individual sensor nodes to connect directly to cloud platforms in case for IoT at the base station.
- Reduce cost and complexity in IoT based WSNs.
- Facilitates protocol interfacing and internetworking, thereby leading to effective device management. In addition, they provide a wide range of protocol accessibility.

The comparison of available research based IoT gateways and commercial gateways with the developed ANR IoT gateway is presented in Table 6.2 and 6.3, respectively.

Table 6.2. Comparison of IoT Gateway Features (Research based Gateways)

Ref	Name	Features	Function
[171]	Heterogenous IoT Gateway	<ul style="list-style-type: none"> Ethernet, CAN and RS485 interfaces ZigBee, Bluetooth and GSM/GPRS modules 	Dynamic priority scheduling for data conversion and communication.
[172]	Multi-Interface IoT Gateway	<ul style="list-style-type: none"> Wireless sensing, Bluetooth and Wi-Fi modules. IR light emitting diode 	Control for smart buildings
[173]	Large scale autonomic gateway	<ul style="list-style-type: none"> ZigBee, Bluetooth and Wi-Fi modules. Apps for data processing and communication. 	<ul style="list-style-type: none"> Dynamic discovery of IoT devices Automatic hardware updates Management of smart things
[174]	Reliable and self-configurable IoT gateway	<ul style="list-style-type: none"> Wi-Fi and LTE interfaces Ethernet interface IoTivity framework 	Smart home management and control
ANR BSG	Base station gateway	<ul style="list-style-type: none"> Wi-Fi development board Wireless serial module UART interface 	<ul style="list-style-type: none"> Direct data aggregation to cloud platforms Data processing and communication to base station visualization GUIs

Table 6.3. Comparison of IoT Gateway Features (Commercial Gateways)

Ref	Name	Features	Function
[175]	Volansys IoT Gateway	<ul style="list-style-type: none"> Ethernet, USB and NFC interfaces ZigBee, Bluetooth, Thread, Z-Wave connectivity MQTT and CoAP protocols for cloud integration. Multiple memory capabilities (SDRAM, uSD Card, eMMC, etc.) 	<ul style="list-style-type: none"> Smart building management Industrial automation Smart healthcare Smart security

		<ul style="list-style-type: none"> ▪ Edge computing and data pre-processing 	
[176]	Cascade IoT Gateway	<ul style="list-style-type: none"> ▪ Ethernet and USB interfaces ▪ ZigBee, Bluetooth, Wi-Fi, IPv6, Thread, Wirepas Mesh connectivity ▪ Cellular LTE Cat-1 (optional) ▪ MQTT or HTTPS protocol for cloud integration ▪ Up to 8GB memory for data pre-processing 	Smart systems
ANR BSG	Base station gateway	<ul style="list-style-type: none"> ▪ UART interface ▪ Wi-Fi and Wireless serial connectivity ▪ Reduced memory for data pre-processing ▪ HTTP and MQTT protocols for direct ThingSpeak cloud integration 	<ul style="list-style-type: none"> ▪ Direct data aggregation to cloud platforms ▪ Data processing and communication to base station visualization GUIs

The developed ANR IoT gateway shares similar features as that available with the research based and commercial IoT gateways, and these features are:

- External hardware connection interfaces i.e. UART or ethernet
- Wi-Fi connectivity between gateway and internet.
- HTTP and MQTT protocols for direct integration to cloud platforms.
- Data pre-processing capabilities based on large memory size.

While the major differences in features include:

- The use of wireless serial connectivity for communication with the sensor nodes in the ANR IoT gateway. However, in the research based and commercial IoT gateways, multiple connectivity protocols are used such as ZigBee, Thread, Z-Wave, Bluetooth, etc.
- Small memory size is used for data pre-processing.

The differences between the ANR IoT gateway and research based, and commercial IoT gateways is due to the fact that the ANR IoT gateway was developed and customized for the given ANR system application.

6.3.4 IoT BS and IoT Sensor Node

The developed ANR hybrid architecture supports IoT integration at the base station level. The current architecture is considered fully integrable with IoT technologies and consistent with other WSN IoT based architectures and applications as presented in [177]–[183].

In order to have the IoT at the node level, each sensor node should be connected directly to the ThingSpeak IoT/cloud platform. Hence, the sensor nodes communicate their data both to the cloud platforms directly, and to the base station (for visualization) through the MRS nodes. The data communicated directly to the cloud can also be accessed at the local base station.

In order to adopt IoT at the node level, the following are required:

- An additional wireless hardware module with Wi-Fi (IEEE 802.11) protocol on all the sensor nodes to enable direct internet connection.
- A processor at each sensor node that can accommodate the additional wireless communication module.

The proposed structure is illustrated in Fig. 6.10.

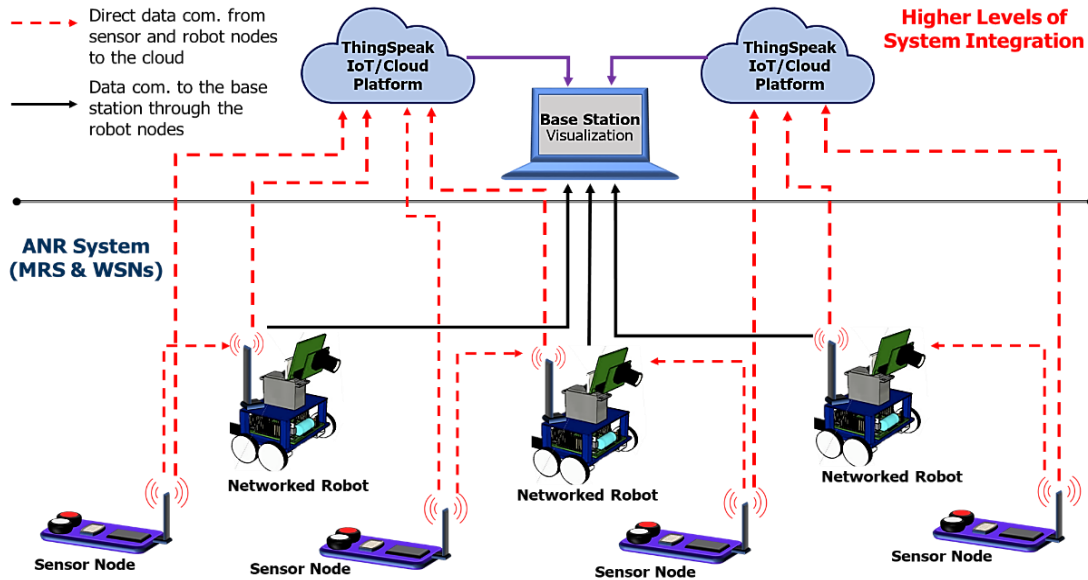


Fig. 6.10. ANR structure for node based IoT integration

The IoT at the node level has the following pros and cons:

Pros:

- It improves the system reliability and scalability.
- It facilitates cloud analytics-based action triggers on the environment through the various sensor nodes

Cons:

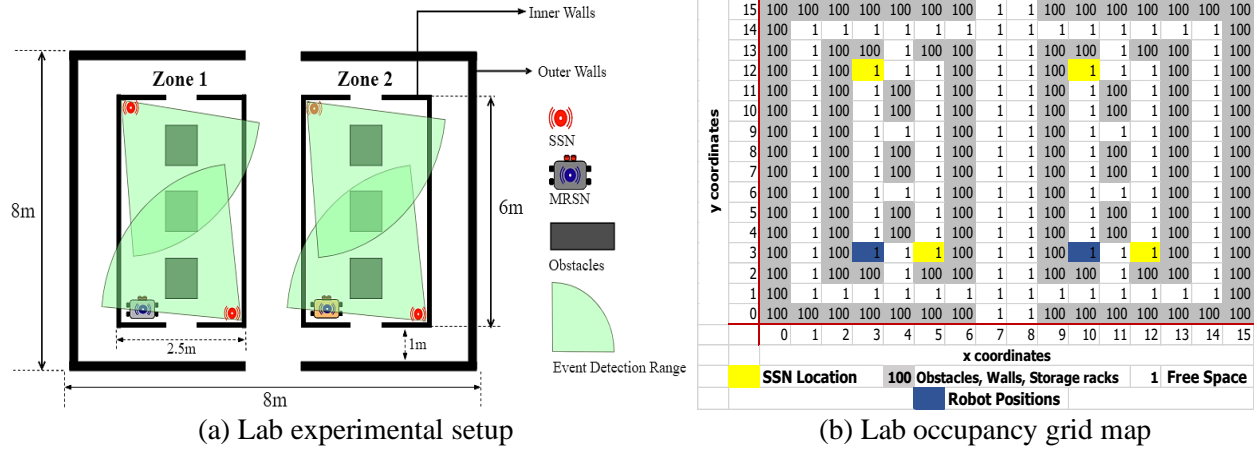
- Increases the structural complexity of the sensor nodes.
- Increases the cost of the sensor nodes since extra hardware is needed.

6.4 Case Study Implementation with the Developed IoT based ANR

A layout comprising two zones, 4 SSNs (SSN1, SSN3 – zone 1 and SSN5, SSN7 – zone 2) and two MRSNs – MRSN1 and MRSN2 was designed and implemented to demonstrate the operation of the system in a real lab environment.

6.4.1 The Test Environment

The setup of lab environment supporting the selected case study to demonstrate the IoT based ANR system is shown in Fig. 6.11(a), while Fig. 6.11(b) shows its occupancy grid map.



6.4.2 Sensor Nodes Distribution and Operation

In the selected lab environment, the four SSNs, SSN1, SSN3 – zone 1 and SSN5, SSN7 – zone 2 were attached to poles of height 2m, while the MRSNs were placed on the lab floor. Boxes were used to represent fixed objects within the zones. The layout is shown Fig. 6.12.

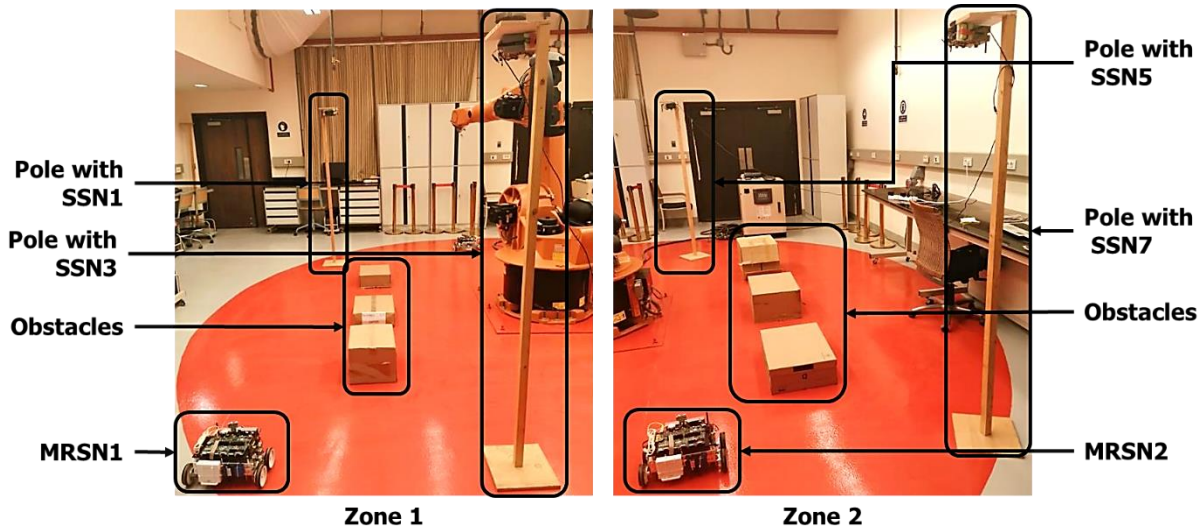


Fig. 6.12. The actual layout of the implemented lab environment.

The experimental set up was ran for 1hour 30 minutes, during which the lab door was opened several times to have external disturbance that influence a source changing the temperature and humidity within the lab. Fig. 6.13(a) – (f) shows SSNs and MRSNs in zones 1 and 2 at the start of experiment. As seen from the figures, the LCD indicator does not show any event data on its screen yet, as the sensors are being calibrated for the environment conditions. Hence, the green LEDs are ON, while the event alert is “Normal”.

After the calibration phase, the sensors begin to detect the environment conditions within the lab environment and communicate the data to the base station (BS). Fig. 6.14(a) – (f) shows the sensor nodes after several sensor readings have been recorded and processed by the MCU. For example, in Fig. 6.14(a), SSN1 recorded a temperature of 26⁰C, RH of 18%, flame – “YES” (erroneous reading due to infrared lamps in the environment), motion – “YES” (correct due to movements within environment) and odour/smoke – “NO” (correct). The resulting event alert was correctly reported as “HIGH”, since motion was detected within the environment, no actual flames were detected (eliminating the erroneous reading from the infrared lamps) and the temperature/RH readings fall within the high alert threshold. The results of all sensor nodes are communicated to the BS in the same format as presented in section 5.2.2.

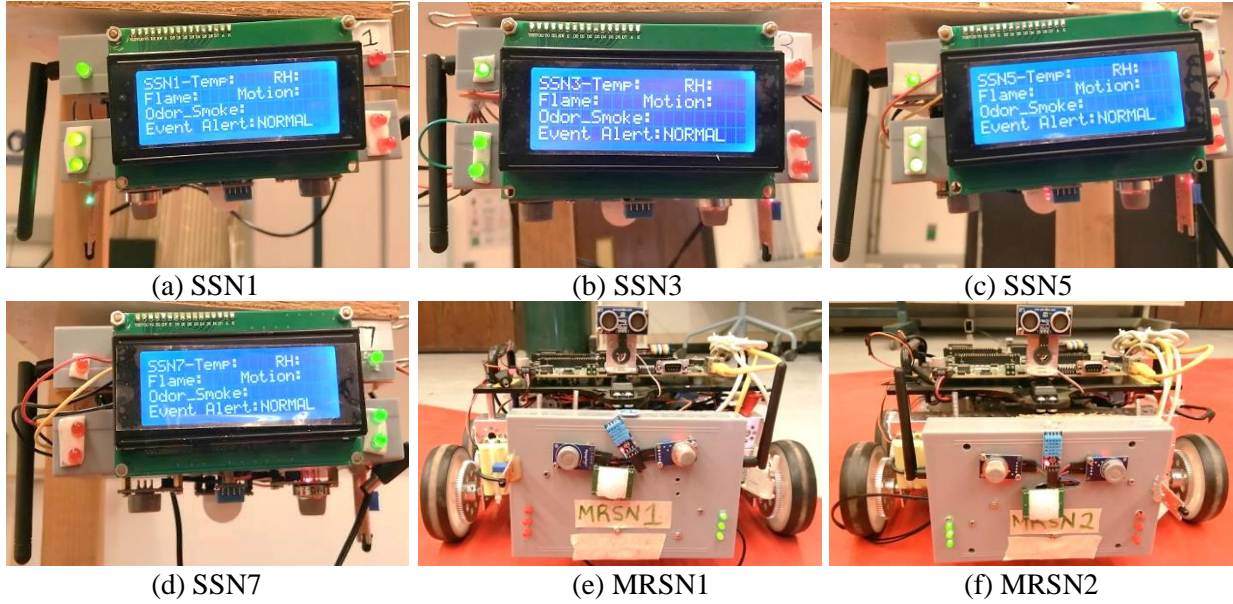


Fig. 6.13. Sensor nodes at the start of the experiment

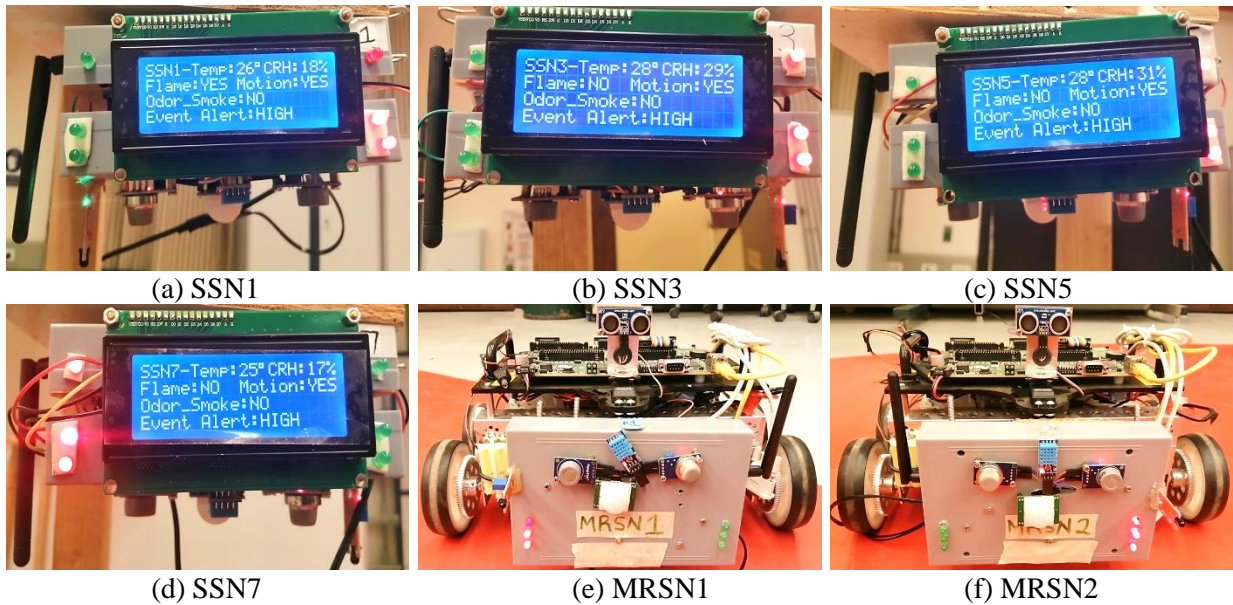


Fig. 6.14. Sensor nodes used during experiment

6.4.3 MRSN Operation

In section 5.3.3, three test cases were presented for the MRS navigation. For the hardware implementation, only the first two cases are experimented, they involve:

- The robots navigating assigned paths within/around the zones when no SSN alert is received.
- The robots (MRSNs) navigating to the alerting SSN location.

Case 1: MRSN navigating assigned path within/around zone

When no alert is issued by the SSNs, the MRSNs can navigate assigned paths within/around their zones. Fig. 6.15 shows MRSN1 example assigned path within its zone in the lab environment.

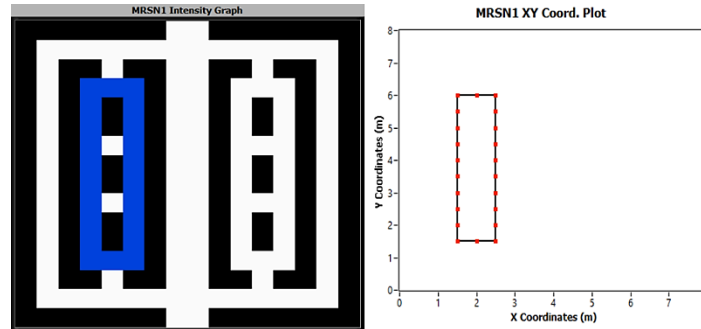
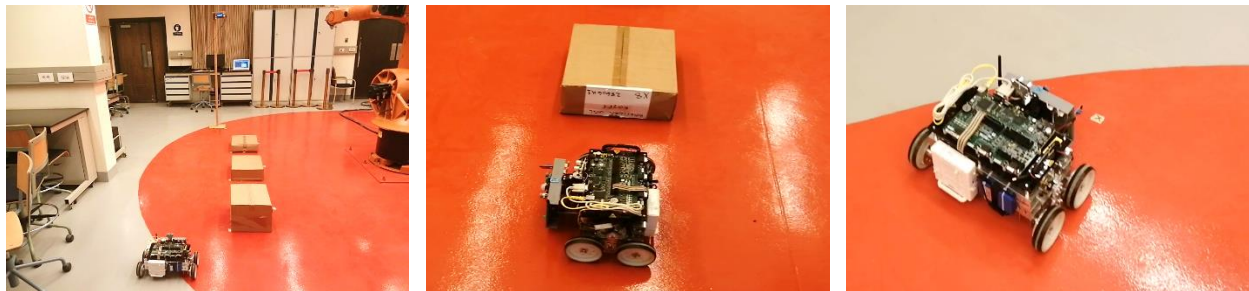


Fig. 6.15. MRSN1 navigation assigned path within zone

The navigation of MRSN1 within its zone in the lab environment is shown in Fig. 6.16(a) – (c). Fig. 6.16(a) shows it at the beginning of the path navigation, while Fig. 6.16(b) shows it half-way through the path navigation and Fig. 6.16(c) shows it at the end of the path navigation.



(a) Start of path navigation (b) Half-way through path navigation (c) End of path navigation

Fig. 6.16. MRSN1 navigation of assigned path within zone1

As the MRSN1 is navigating its assigned paths within its zone, the associated coordination status displays the text message “MRSN1 is navigating its assigned paths around its zone”, as shown in Fig. 6.17.



Fig. 6.17. MRSN1 Coordination Status for Case 1

Case 2: MRSNs navigating paths to alerting SSNs

When alerts are issued by the zone SSNs, the MRSNs navigates to the alerting SSN locations following paths generated by the A* algorithm from their start positions to the SSN locations. Table 6.4 shows the MRNs initial locations, the alerting SSN locations from within the zones, and the effective distance of each MRSN to its relevant alerting SSN.

Table 6.4. MRSNs and alerting SSN coordinates with distance between them: case 2

	Initial Position	Alerting SSNs Location		Effective Distance
MRSN1	(1.5, 1.5)	SSN3	(2.5, 1.5)	1.0 (< 7)
MRSN2	(5.0, 1.5)	SSN5	(5.0, 6.0)	4.5 (< 7)

The hardware MRS GUI control interface showing the alerting SSN for MRSN1 and that for MRSN2 in Fig. 6.18(a) and (b) respectively.

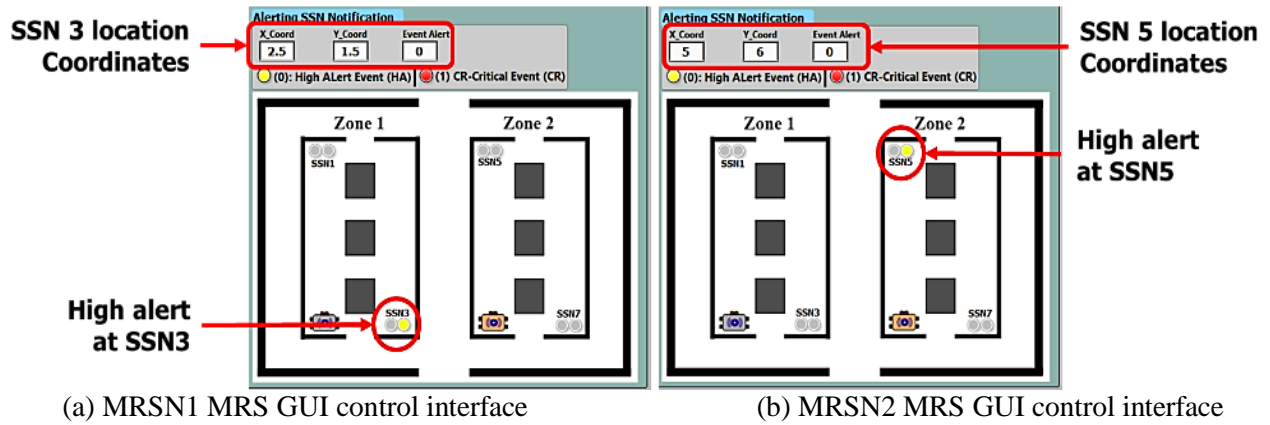
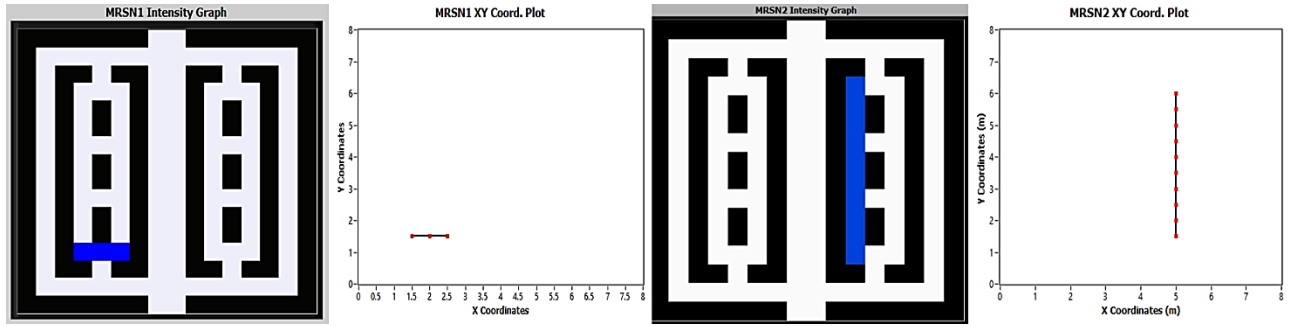


Fig. 6.18. MRS GUI control interface displaying SSN alert

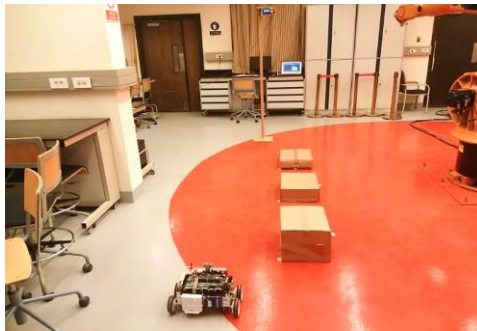
The generated paths to the relevant alerting SSN locations for each MRSN, using the A* algorithm are shown in Fig. 6.19(a) and (b). The navigation of the MRSNs within their zones in the lab environment is shown in Fig. 6.20(a) – (d). Fig. 6.20(a) shows MRSN1 at its initial position before path navigation, while Fig. 6.20(b) shows MRSN1 at its final position after navigating the generated path, and Fig. 6.20(c) shows MRSN2 at its initial position before path navigation, while Fig. 6.20(d) shows MRSN2 at its final position after navigating the generated path.



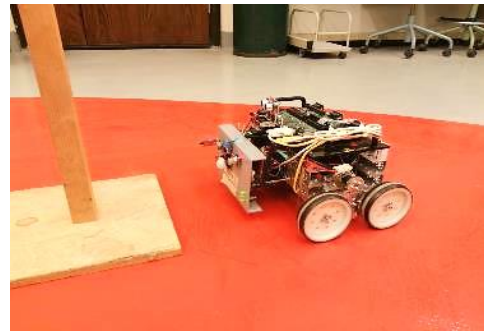
(a) MRSN1 navigation path to alerting SSN3

(b) MRSN2 navigation path to alerting SSN5

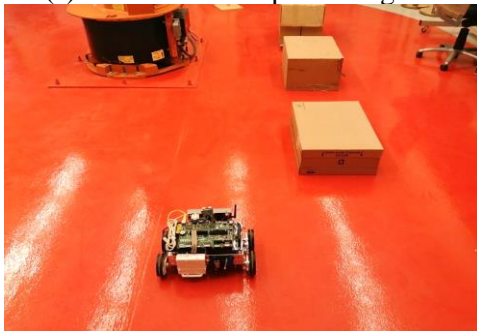
Fig. 6.19. MRSNs navigation paths for case 2.



(a) MRSN1 before path navigation



(b) MRSN1 after path navigation



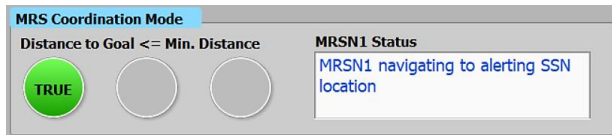
(c) MRSN2 before path navigation



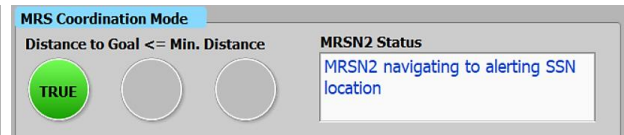
(d) MRSN2 after path navigation

Fig. 6.20. MRSs navigation of paths to alerting SSNs within zones of lab environment

The resulting coordination mode for MRSN1 and MRSN2 is shown in Fig. 6.21(a) and (b) respectively.



(a) MRSN1 coordination status



(b) MRSN2 coordination status

Fig. 6.21. MRSNs coordination status

6.4.5 IoT BS Hardware Results Aggregation, Visualization and Cloud Integration

Similar to the BS simulation, the BS implementation involves receiving the sensor node information, aggregating and visualizing it on the SND GUIs, then storing them on the cloud platforms for analysis.

A. SND GUI Results Aggregation and Visualization

The hardware implementation involved just two zones (1 and 2), hence, the SND GUI information tab displayed the sensor nodes of these zones as the only nodes sending sensory data during the experiment. The SND GUI information tab is shown in Fig. 6.22. Fig. 6.23(a) – (e) and Fig. 6.24(a) – (e) shows visualizations examples of the aggregated sensor results data plots for MRSN1 and SSN1 respectively. The results show that temperature around MRSN1 O-SSN was relatively constant at 25°C, while the relative humidity fluctuated between 25% and 32%, and no CO gas or smoke was detected. For SSN1, the temperature ranged from 23 – 33°C, the relative humidity was relatively constant at 18% while no CO gas was detected, and maximum smoke concentration detected was 15ppm. The difference in temperature and relative humidity detected by SSN1 and MRSN1 O-SSN can be attributed to the position (due to elevation) of the sensor nodes and circulation of air within the lab environment when the lab door was half opened or closed. Similar results were obtained for the other sensor nodes. From the SND GUI, the aggregated results data is uploaded to the MS cloud platform, while the BSG directly uploads the results data to the ThingSpeak cloud platform.

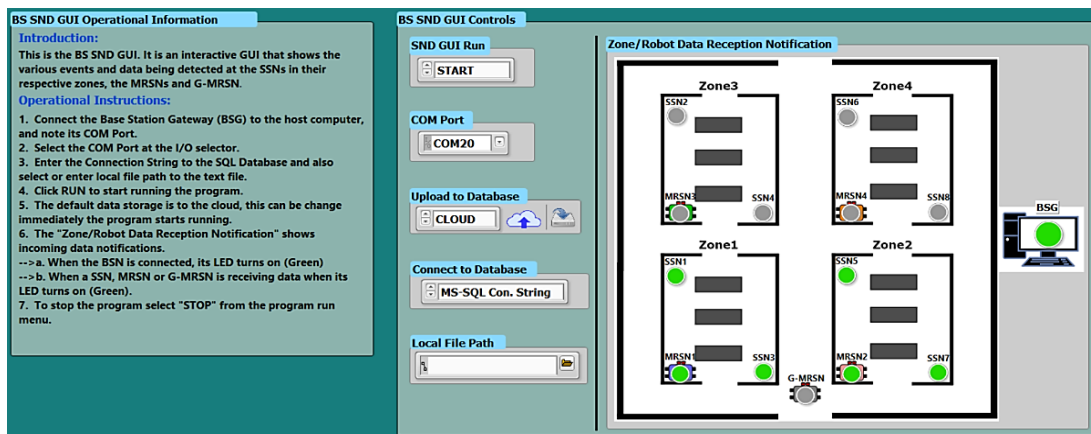
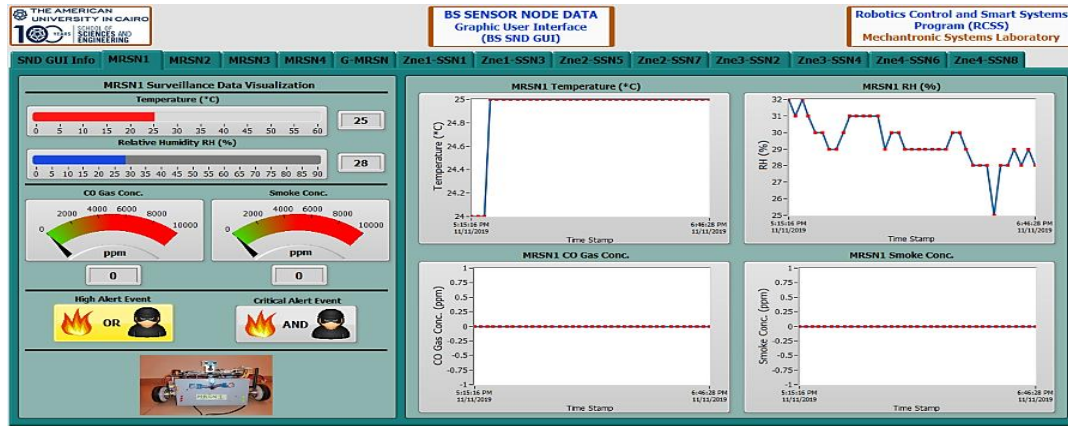
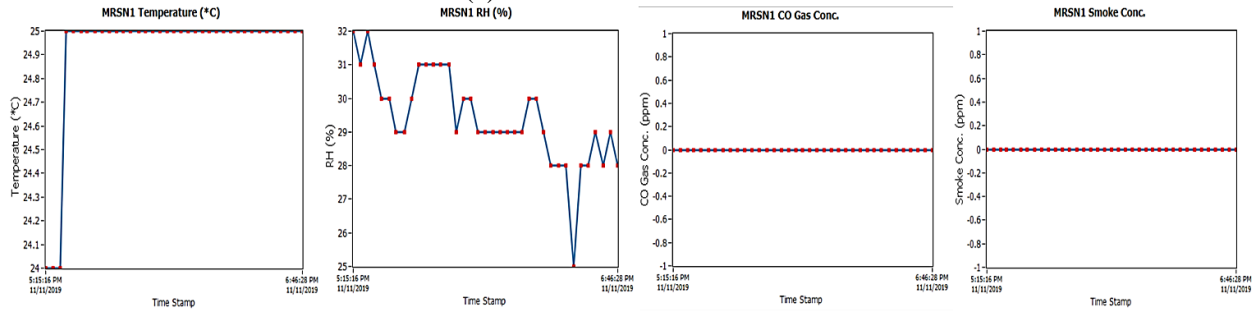


Fig. 6.22. BS SND GUI information tab during hardware implementation



(a) MRSN1 tab visualization



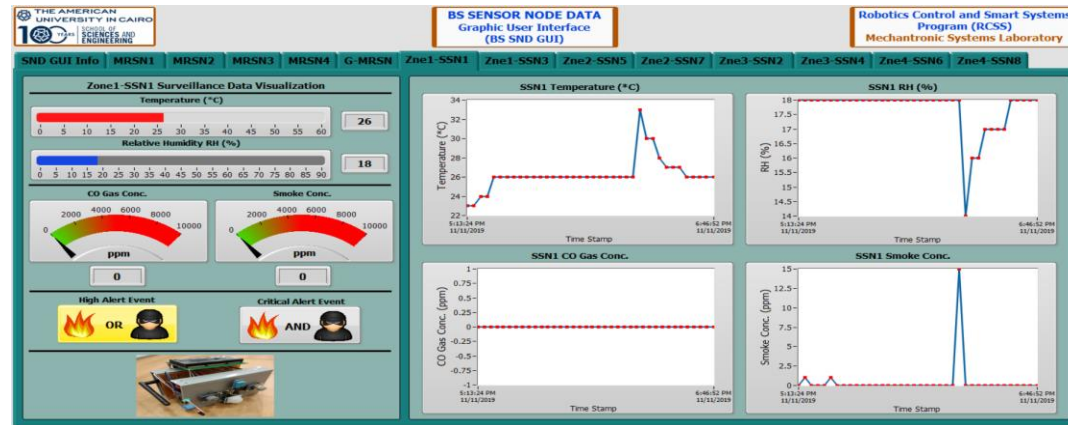
(b) MRSN1 temperature

(c) MRSSN1 RH

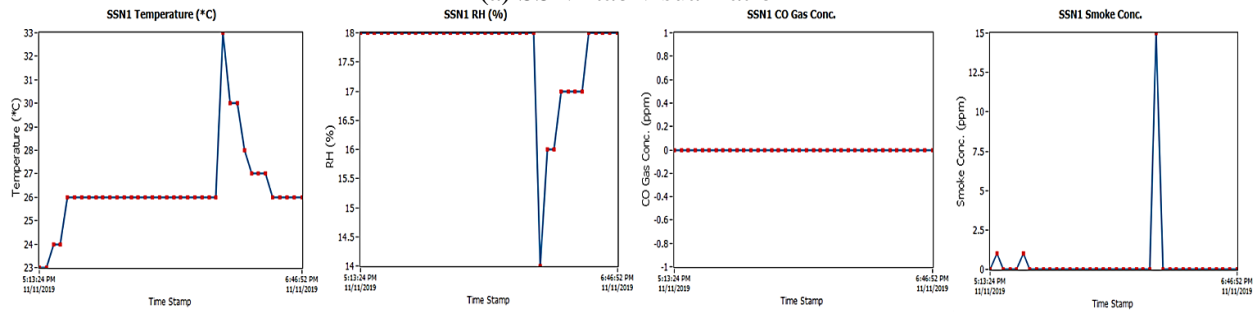
(d) MRSN1 CO gas conc.

(e) MRSN1 smoke conc.

Fig. 6.23. MRSN O-SSN1 hardware results



(a) SSN1 tab visualization



(b) SSN1 Temperature

(c) SSN1 RH

(d) SSN1 CO Gas Conc.

(e) SSN1 Smoke Conc.

Fig. 6.24. SSN1 hardware results

B. Cloud Platforms Result Aggregation, Visualization and Analytics

The hardware sensor node results were aggregated at the MS cloud platform through the SND GUI, and at the ThingSpeak cloud platform through the BSG.

a. MS Cloud Platform Data Visualization and Analytics

At the MS cloud platform, the complete resulting data (both simulation and hardware implementation) is aggregated. The MS SSMS can display the SQL database table entry for the hardware results section only, but the basic statistical analysis table shows the complete results of the analytics on the whole SQL database, likewise the web application visualization. Fig. 6.25(a) and (b) shows the hardware database table entry and analytics on the whole database respectively.

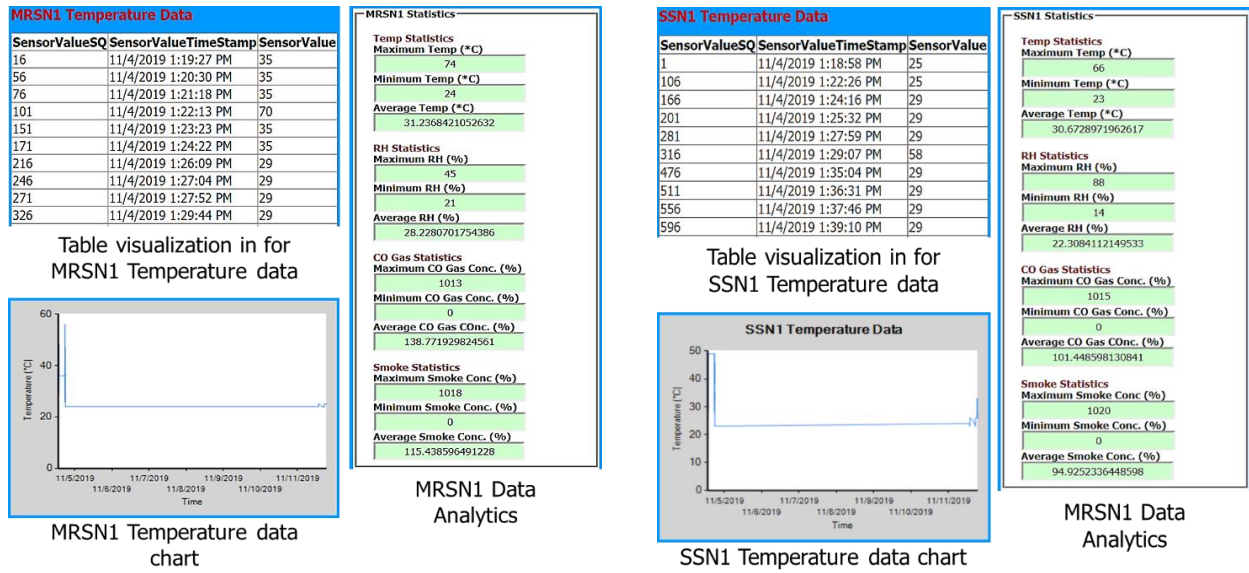
The web application snippets showing charts and table visualization for MRSN1 and SSN1 sensor data and their basic analytics are shown in Fig. 6.26(a) and (b) respectively.

Results						Messages									
	SensorID	SensorName	SensorValueSQ	SensorValueTimeStamp	SensorValue	EventAlert		SensorID	SensorName	SensorTypeID	SensorStats_DataID	SensorID	AverageData	MinData	MaxData
1	26	SSN1_Temp	5242	2019-11-11 15:47:54.887	25	NULL	1	1	G-MRSN_Temp	1	1	1	33.8604651162791	21	49
2	27	SSN1_RH	5243	2019-11-11 15:47:55.057	18	NULL	2	2	G-MRSN_RH	2	2	2	28.5581395348837	21	41
3	28	SSN1_Gas	5244	2019-11-11 15:47:55.127	0	NULL	3	3	G-MRSN_Gas	3	3	3	239.558139534884	7	1007
4	29	SSN1_Smk	5245	2019-11-11 15:47:55.250	0	NULL	4	4	G-MRSN_Smk	4	4	4	199.046511627907	13	1013
5	30	SSN1_EA	5246	2019-11-11 15:47:55.450	0	High Alert	5	5	G-MRSN_EA	5	5	5	0.13953488372093	0	1
6	36	SSN3_Temp	5247	2019-11-11 15:48:12.500	26	NULL	6	6	MRSN1_Temp	6	6	6	31.2368421052632	24	74
7	37	SSN3_RH	5248	2019-11-11 15:48:12.630	29	NULL	7	7	MRSN1_RH	7	7	7	28.2280701754386	21	45
8	38	SSN3_Gas	5249	2019-11-11 15:48:12.700	0	NULL	8	8	MRSN1_Gas	8	8	8	138.771929824561	0	1013
9	39	SSN3_Smk	5250	2019-11-11 15:48:12.763	0	NULL	9	9	MRSN1_Smk	9	9	9	115.438596491228	0	1018
10	40	SSN3_EA	5251	2019-11-11 15:48:12.837	0	High Alert	10	10	MRSN1_EA	10	10	10	0.0701754385964912	0	1
11	6	MRSN1_Temp	5252	2019-11-11 15:48:20.373	24	NULL	11	11	MRSN2_Temp	11	11	11	32.6306306306306	22	46
12	7	MRSN1_RH	5253	2019-11-11 15:48:20.440	32	NULL	12	12	MRSN2_RH	12	12	12	35.8018018018018	22	68
13	8	MRSN1_Gas	5254	2019-11-11 15:48:20.527	0	NULL	13	13	MRSN2_Gas	13	13	13	106.288288288288	0	1014
14	9	MRSN1_Smk	5255	2019-11-11 15:48:20.620	0	NULL	14	14	MRSN2_Smk	14	14	14	100.405405405405	0	1020
15	10	MRSN1_EA	5256	2019-11-11 15:48:20.723	0	High Alert	15	15	MRSN2_EA	15	15	15	0.0630630630630631	0	1
16	46	SSN5_Temp	5257	2019-11-11 15:48:35.573	27	NULL	16	16	MRSN3_Temp	16	16	16	33.125	22	96
17	47	SSN5_RH	5258	2019-11-11 15:48:36.110	30	NULL	17	17	MRSN3_RH	17	17	17	27.4196428571429	20	40
18	48	SSN5_Gas	5259	2019-11-11 15:48:36.480	0	NULL	18	18	MRSN3_Gas	18	18	18	199.857142857143	13	1013
19	49	SSN5_Smk	5260	2019-11-11 15:48:36.857	0	NULL	19	19	MRSN3_Smk	19	19	19	213.785714285714	18	1018
20	50	SSN5_EA	5261	2019-11-11 15:48:37.167	0	High Alert	20	20	MRSN3_EA	20	20	20	0.0803571428571429	0	1
21	56	SSN7_Temp	5262	2019-11-11 15:48:55.820	24	NULL	21	21	MRSN4_Temp	21	21	21	29.5340909090909	20	48
22	57	SSN7_RH	5263	2019-11-11 15:48:56.307	18	NULL	22	22	MRSN4_RH	22	22	22	27.3522727272727	21	41
23	58	SSN7_Gas	5264	2019-11-11 15:48:56.780	0	NULL	23	23	MRSN4_Gas	23	23	23	184.090909090909	14	1014
24	59	SSN7_Smk	5265	2019-11-11 15:48:57.100	0	NULL	24	24	MRSN4_Smk	24	24	24	201.363636363636	20	1020
25	60	SSN7_EA	5266	2019-11-11 15:48:57.377	0	High Alert	25	25	MRSN4_EA	25	25	25	0.0909090909090909	0	1
26	11	MRSN2_Temp	5267	2019-11-11 15:49:02.650	35	NULL	26	26	SSN1_Temp	26	26	26	30.6728971962617	23	66
27	12	MRSN2_RH	5268	2019-11-11 15:49:03.470	50	NULL	27	27	SSN1_RH	27	27	27	22.3084112149533	14	88
28	13	MRSN2_Gas	5269	2019-11-11 15:49:03.757	0	NULL	28	28	SSN1_Gas	28	28	28	101.448598130841	0	1015
29	14	MRSN2_Smk	5270	2019-11-11 15:49:04.000	0	NULL	29	29	SSN1_Smk	29	29	29	94.9252336448598	0	1020
30	15	MRSN2_EA	5271	2019-11-11 15:49:04.400	0	High Alert									

(a) MS SQL database hardware results visualization

(b) MS SQL full database Analytics

Fig. 6.25. MS SSMS SQL database visualization and analytics



(a) MRSN1 web app visualization

(b) SSN1 web app Visualization

Fig. 6.26. Web application visualization and analytics

b. ThingSpeak Cloud Platform Data Visualization and Analytics

The sensory data of each sensor node can be visualized on the sensor node's channel on the ThingSpeak cloud platform. Fig. 6.27(a) and (b) shows the hardware data visualization for MRSN1 and SSN1 channels respectively.

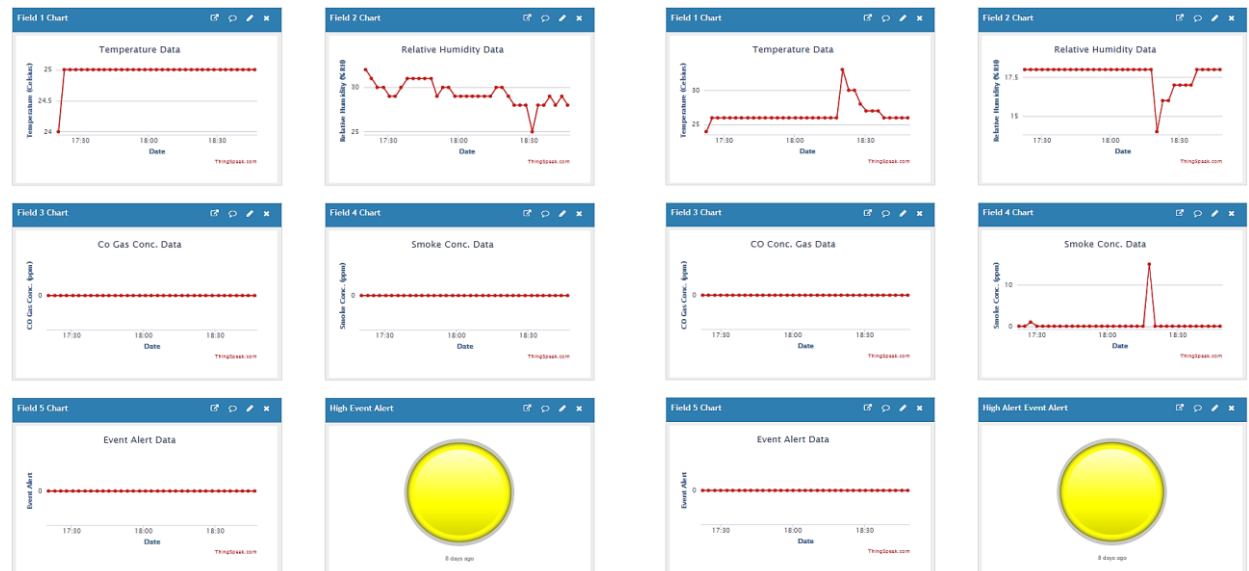


Fig. 6.27. ThingSpeak channel hardware results visualization

ThingSpeak data analytics involved statistical and preservation metrics analytics of the sensor node results from the zones (1 and 2). These analytics are hereby presented.

i. Statistical Data Analytics

Similar to the simulate results data analytics, the hardware results statistical results data analytics are presented in Table 6.5.

Table 6.5. Hardware results statistical data analytics

Zone 1 Statistical Data Analytics					
Sensor Variable	Maximum Value	Minimum Value	Mean Value	Maximum Value Timestamp	Minimum Value Timestamp
Temperature ($^{\circ}\text{C}$)	33	23	26.8936	11-Nov-2019 18:20:21	11-Nov-2019 17:17:29
RH (%)	32	14	22.8617	11-Nov-2019 13:29:36	11-Nov-2019 18:20:21
CO Gas (ppm)	719	0	7.6489	11-Nov-2019 17:20:18	11-Nov-2019 13:29:36
Smoke (ppm)	574	0	6.2979	11-Nov-2019 18:23:10	11-Nov-2019 13:29:36
Zone 2 Statistical Data Analytics					
Sensor Variable	Maximum Value	Minimum Value	Mean Value	Maximum Value Timestamp	Minimum Value Timestamp
Temperature ($^{\circ}\text{C}$)	29	23	26.3511	11-Nov-2019 18:26:02	11-Nov-2019 15:46:47
RH (%)	31	17	23.4149	11-Nov-2019 17:15:35	11-Nov-2019 13:41:41
CO Gas (ppm)	0	0	0	11-Nov-2019 13:39:36	11-Nov-2019 13:39:36
Smoke (ppm)	1514	0	27.0638	11-Nov-2019 17:20:53	11-Nov-2019 13:39:36

The analytics shows that similar temperature and RH conditions were obtainable in both zones, whereby the temperature range is 23 – 33 $^{\circ}\text{C}$ and 23 – 29 $^{\circ}\text{C}$ for zone 1 and 2 respectively, while the RH range is 14 – 32% and 17 – 31% for zone 1 and 2 respectively. This can be attributed to the fact that they were located within the same lab environment. The CO gas and smoke concentrations had mean values less 28ppm across both zones, implying that there were few instances of high concentration of combustion gases within the lab environment.

The regression analysis shows an increasing pattern for the temperature and humidity correlation as shown in Fig. 6.28(a) and (b) for zone 1 and 2 respectively. The slopes for the best fit pattern lines between the temperature and RH are presented in Table 6.6; the temperature/RH fitting slope of zone 2 is the steepest with a value of 3.5666. The results show that the Temperature and RH were mostly increasing within the zones. The regression analysis can be used to maintain a specific temperature and humidity within the zones by controlling the conditions to lower the changing trend and maintain a zero-slope pattern between the temperature and relative humidity.

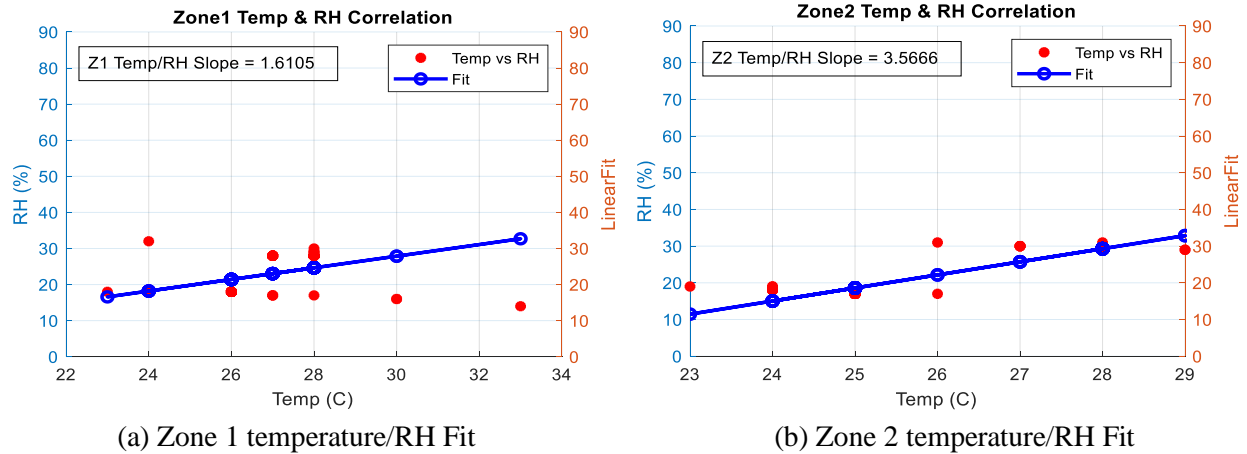


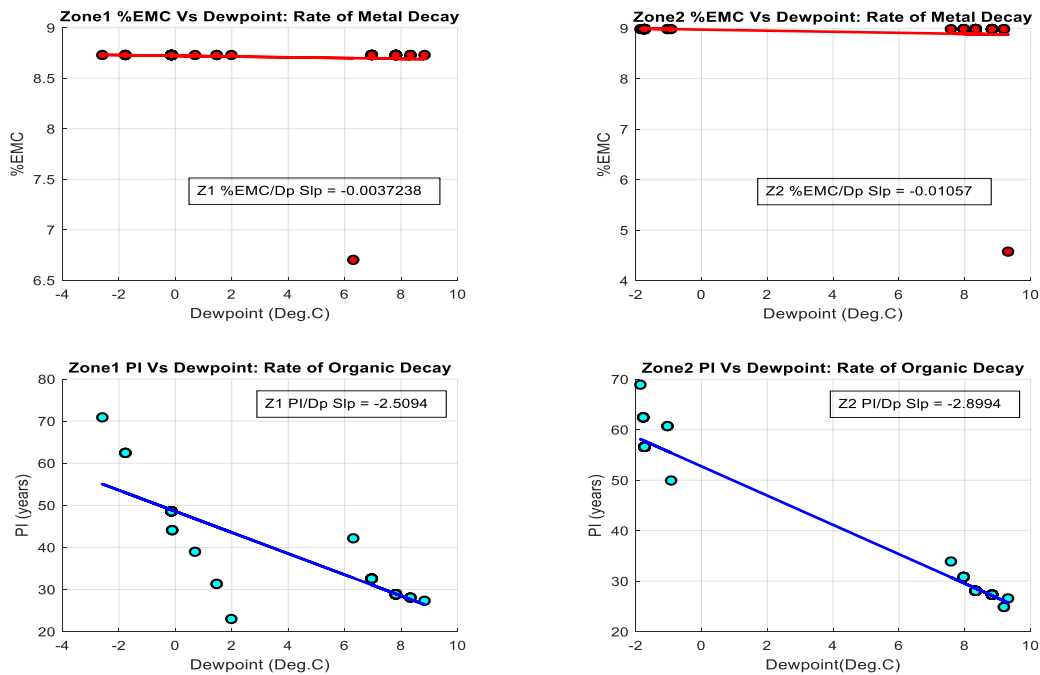
Fig. 6.28. Zones temperature and RH regression analysis

Table 6.6. Temperature/RH regression fit pattern line slopes per zone

Zones	Slope of Temperature/RH Best Fit Pattern Line
Zone 1	1.6501
Zone 2	3.5666

ii. Preservation Metrics Analytics

Similar to the simulation results, the relationship between the %EMC and PI preservation metrics and dewpoint is shown in Fig. 6.29(a) and (b) for zone 1 and 2 respectively, while Table 6.7 shows the slopes of the best fit pattern lines.



(a) Zone 1&2 %EMC and PI vs Dewpoint

(b) Zone 3&4 %EMC and PI vs Dewpoint

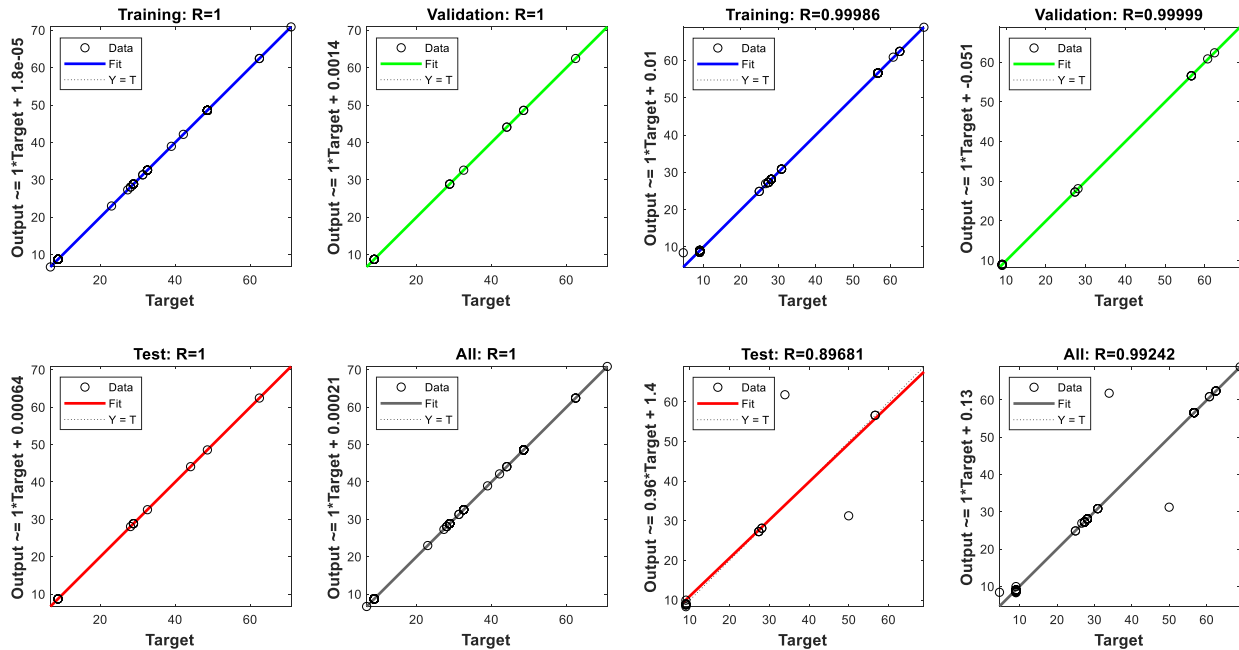
Fig. 6.29. Zones %EMC and PI vs dewpoint analytics

Table 6.7. %EMC and PI vs Dewpoint best fit pattern line slopes

Zones	Slope of Best Fit Pattern Line	
	%EMC	PI
Zone 1	– 0.0037328	– 2.5094
Zone 2	– 0.01057	– 2.8994

The outcome of the correlation shows that %EMC is relatively constant as the dewpoint changes across the two zones while the PI is decreasing as the dewpoint is increasing across both zones. But zone 2 %EMC falls slightly faster than zone 1 %EMC since the slope of its best fit pattern line is -0.01057 compared to -0.0037328 of zone 1, while zone 2 PI decreases slightly faster than zone 1 since the slope of its best fit pattern line is -2.8994 compared to -2.5094 of zone 1. The constant trend of %EMC with dewpoint implies that metals stored within the zones are at a lower risk of corrosion while the decreasing trend of PI with dewpoint implies that organic materials stored within the zones are in greater risk of degradation.

To predict the %EMC and PI per zone using a trained neural network, the hardware data inputs/targets were structured in the same pattern as the simulation data presented in section 5.4.2. Hence, the regression analysis of the neural network for the hardware results for zone 1 and 2 is presented in Fig. 6.30(a) and (b) respectively. The correlation coefficients (R – values) between the predicted outputs and targets for each zone is presented in Table 6.8, together with the training time.



(a) Zone 1 neural network regression analysis

(b) Zone 2 neural network regression analysis

Fig. 6.30. Neural network regression analysis per zone.

Table 6.8. Neural network regression analysis (R – Values) per zone

Zones	Correlation Coefficient (R – Values)			
	Training	Validation	Test	All
Zone 1	1.0	1.0	1.0	1.0
Zone 2	0.99986	0.99999	0.89681	0.99242
		Network Training Time (s)		
Zone 1 – 2		< 5		

From the R – values presented in Table 6.8 and the regression plots in Fig. 6.30, it can be deduced that there is a good fit between the predicted outputs and targets for each zone, indicating that the network was well trained on the given data.

The measure of risk associated with a zone in terms of %EMC (for metallic corrosion) and PI (organic degradation), are predicted based on the running averages of the temperature and RH from each zone inputted into the trained neural network. The following values presented in Table 6.9 shows the prediction results of the neural network for each zone.

Table 6.9. %EMC and PI predictions per zone

% EMC Predictions per Zone				
Zones	%EMC	Status/Color Code	Metal Corrosion Risk	Average Dewpoint
Zone 1	7.8168	OK	Limited risk of corrosion	3.6162
Zone 2	8.3270	OK	Limited risk of corrosion	3.4151
PI Predictions per Zone				
Zones	PI (years)	Status/Color Code	Organic Degradation Risk	Average Dewpoint
Zone1	28.7899	RISK	Accelerated rate of organic decay	3.6162
Zone 2	26.6957	RISK	Accelerated rate of organic decay	3.4151

The prediction values for %EMC shows limited metal corrosion risk for zone 1 and minimal corrosion risk for zone 2. While the prediction values for PI shows high organic degradation risk for both zones. From the results, it can be deduced that the zones are favorable to store metallic materials more than organic materials, if temperature and RH within the zones remain fairly constant. Also, an organic material will start degrading after approximately 28 years in zone 1 and 26 years in zone 2. These analytics can help determine the storage time, and type of material to be stored within each zone. Generally, the conditions within the zones are favorable for a warehouse storage facility with constant movement of stored goods in and out of the facility.

6.4.6 Triggering actions on the environment through Cloud Analytics

The developed ANR architecture involves an upward data communication from the zone sensor nodes to the robots nodes, and then to the base station, where the data is aggregated and visualized, and communicated to the cloud platforms for storage and analytics.

Action triggering on the operational environment can be operated:

- Directly by the SSNs when they trigger an action within their zones based on the classified event.
- From the cloud analytics, whereby specific actions are triggered based the data analyzed in the the cloud platforms.

A downward triggering action/command can be initiated and communicated through the cloud analytics to:

- The base station and unto the sensor nodes, robots or other actuators using the current ANR architecture.
- The sensor nodes and robots directly using the IoT at the node level architecture.

Both action triggering scenario are illustrated in Fig. 6.31.

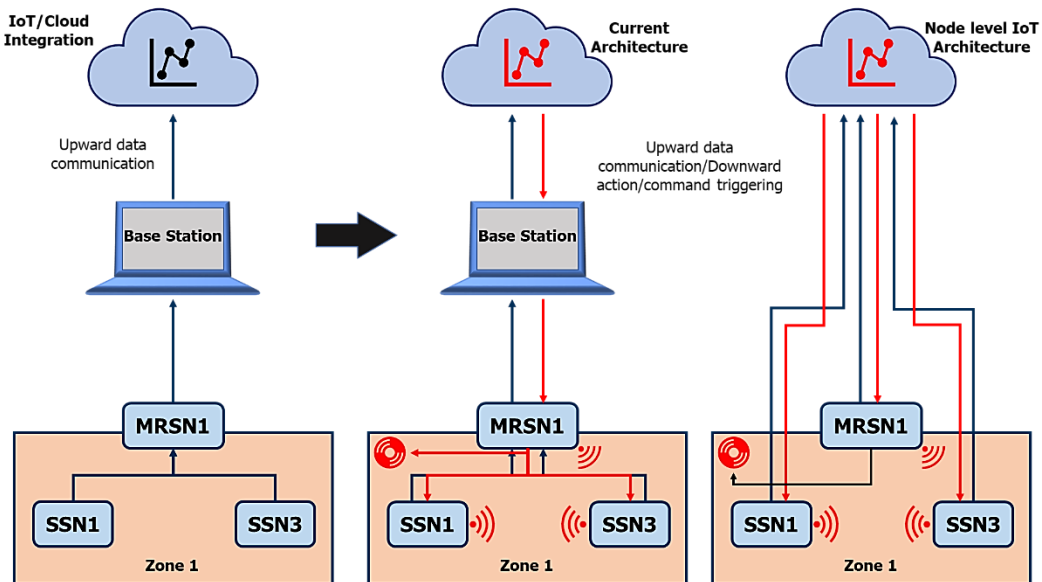


Fig. 6.31. Downward action/command triggering communication

6.5 ANR System Results Discussion

The simulation and hardware implementation results of the developed IoT based hybrid ANR system have thus far been presented. The results confirm that events were detected, correctly classified and communicated by the zone SSNs and MRSN O-SSNs to the BS. The zone MRSNs could navigate assigned paths within/around their environments when no alerts were issued by SSNs, as shown in the MRSN1 example. When alerts were issued by the SSNs, each zone MRSN could navigate to the alerting SSN location, as shown by both MRSN1 and MRSN2 examples. Furthermore, the MRSN O-SSNs could communicate all the SSNs and their own sensory information and results to the BS, which was received by the BSG, processed and sent to the SND GUI and cloud platforms. Also, the MRSNs communicated their navigation paths and location within the zones to the BS to be visualized at the MRS GUI. The cloud platforms stored and performed analytics on the sensory information from the zones.

Comparing the simulation and hardware results, it should be noted that the simulation sensory data was based on pre-collected data contained in the sensor libraries of the PCD-SE while the hardware sensory data were real data reflecting the actual conditions within the lab experimental environment. Also, the models of the sensor nodes used in the recorded outlier data that occurred when the PCD-SE was not running at the required processing speed from the host PC CPU. The hardware sensor nodes did not record such outliers as it interacts with the environment directly. Furthermore, the same navigation algorithm was used for both MRS simulation and hardware implementation, hence, the results were similar except for small errors associated with the actual navigation of the robot while negotiating with the ground.

Finally, at the BS, the SND GUI and cloud platforms aggregated and visualized both the simulation and hardware results. The MS cloud platform aggregated all the results (simulation and hardware), analyzed and displayed them together, while the ThingSpeak cloud platform could analyze and display the simulation and hardware results separately. Also, on the ThingSpeak cloud platform, advance analytics were performed on the results due to the availability of Matlab toolboxes. The ThingSpeak advance analytics can be used to control temperature/RH conditions within the zones and also predict the duration time and type of materials to be stored within the zones. The ThingSpeak cloud platform, therefore, is preferable for IoT/cloud integration than the MS cloud platform. Generally, the cloud platforms provided remote access to the sensory information and results for various types of uses such as processing and analytics.

Chapter 7

CONCLUSIONS AND FUTURE WORK

The concluding remarks and recommendations for future work to further develop the presented ANR system are hereby introduced. It discusses the developed, implemented and tested system including major reflections on the thesis objectives. The chapter also introduces a set of recommendations that can be considered and implemented for further system improvement.

7.1 Conclusions

The developed, implemented and tested hybrid ANR system is featured by the following:

- a. WSNs comprising the good and reliable SSNs and O-SSNs structures, that detect, characterize events and communicate the detection results through a two-tier (ZigBee and Wireless serial) network system to the BS.
- b. MRS comprising the MRSNs and G-MRSN, that navigate assigned paths within/around their assigned spaces. They coordinate their actions toward an alerting SSN using the A* and CNP algorithms.
- c. BS comprising the BSG, SND, MRS GUIs and the integration of cloud platforms:
 - i. The BSG receives detection results from the MRSNs and G-MRSN and communicate them to the SND GUI and ThingSpeak cloud platform.
 - ii. The SND GUI aggregates, visualizes and uploads the detection results to the MS cloud platform.
 - iii. The MRS GUI visualizes the navigation paths of the MRSNs and G-MRSN with the environment.
 - iv. Integrate different cloud platforms (MS cloud and ThingSpeak) to receive, store and analyze system information for the prediction of preservation metrics.
- d. Simulation development environment (SDE) that facilitates the development, integration, testing and validation of the different simulated and hardware components of the system (HIL).

The hybrid system architecture fully supports the integration and concurrent operation of all system components whereby the system information detected results were constantly being updated at the SND GUI and MRS GUI in real-time.

Event at each zone are characterized using fuzzy logic decision making system based on the sensed information within the zone

Both MS and ThingSpeak cloud platforms are used to store sensory information and for basic analytics, while the ThingSpeak cloud platform is used for advance analytics. The advance analytics involves regression analysis between temperature and relative humidity that can be used to control the zone environmental conditions. Furthermore, it includes the prediction of preservation metrics using neural networks. All the sensory information data and analytics can be accessed by the system and by remote users.

Finally, the HIL based simulation concept has been developed and integrated into the system to enable testing, evaluating and validating of the developed algorithm partially on hardware or fully in real-time.

7.2 Future Work

Despite the successful simulation and implementation of the ANR system, the following points can be considered for future improvement and enhancement of the ANR system:

- a. The zone environments were considered to be static i.e. no moving objects within them. This is not usually the case in actual real-life warehousing environments. Hence, a dynamic environment should be considered for future work. The A* algorithm was sufficient to generate paths within the static environment, but for a dynamic environment, a more dynamic path planner like AD* can be combined with localization algorithms to guide the robots as they navigate to alerting SSN locations or continuously navigate their zones when no alerts are issued by the SSNs.
- b. The HIL based simulation involved the integration of whole zone hardware modules (SSNs and MRSNs) into the simulation environment. A further development on the HIL based simulation can involve hardware at the component level.
- c. The MRSNs and G-MRSNs can be equipped with a camera so that they can capture or video various events within the environment and relay them to the BS.
- d. Enhancing the scaling capability of the system to accommodate the growth of the operational environment in terms of space and capabilities or potential enlargement.
- e. The concluded analytics in the cloud can be communicated to the BS whereby the BS can trigger actions such as control of the zone environment.

REFERENCES

- [1] A. Knoll and R. Prasad, "Wireless Robotics: A Highly Promising Case for Standardization," *Wireless Personal Communications*, vol. 64, no. 3, pp. 611–617, Jun. 2012, doi: 10.1007/s11277-012-0604-8. [Online]. Available: <http://link.springer.com/10.1007/s11277-012-0604-8>. [Accessed: 14-Aug-2017]
- [2] L. E. Parker, D. Rus, and G. S. Sukhatme, "Multiple Mobile Robot Systems," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, Cham, 2016, pp. 1109–1134 [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-32552-1_44. [Accessed: 13-Dec-2017]
- [3] M. Chen, Y. Ma, S. Ullah, W. Cai, and E. Song, "ROCHAS: Robotics and Cloud-assisted Healthcare System for Empty Nester," in *Proceedings of the 8th International Conference on Body Area Networks*, ICST, Brussels, Belgium, Belgium, 2013, pp. 217–220, doi: 10.4108/icst.bodynets.2013.253922 [Online]. Available: <http://dx.doi.org/10.4108/icst.bodynets.2013.253922>. [Accessed: 04-Jul-2019]
- [4] M. D. Weiss, J. Peak, and T. Schwengler, "A Statistical Radio Range Model for a Robot MANET in a Subterranean Mine," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2658–2666, Sep. 2008, doi: 10.1109/TVT.2007.912606.
- [5] M. A. Hsieh *et al.*, "Adaptive Teams of Autonomous Aerial and Ground Robots for Situational Awareness," *Journal of Field Robotics*, vol. 24, no. 11–12, pp. 991–1014, Nov. 2007, doi: 10.1002/rob.20222. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20222>. [Accessed: 27-Apr-2018]
- [6] "Networked Robots - IEEE Robotics and Automation Society." [Online]. Available: <http://www.ieee-ras.org/technical-committees/index.php>. [Accessed: 09-Sep-2017]
- [7] A. Manoj, "Wireless Sensor Network- A Theoretical Review," *International Journal of Wired and Wireless Communications*, vol. 1, no. 2, pp. 11–19, Apr. 2013.
- [8] L. E. Parker, "Current State of the Art in Distributed Autonomous Mobile Robotics," in *Distributed Autonomous Robotic Systems 4*, vol. 4, 2003, pp. 1–5 [Online]. Available: <http://www.springerlink.com/index/19821467301079W5.pdf>. [Accessed: 13-Sep-2017]
- [9] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Cham: Springer International Publishing, 2016 [Online]. Available: <http://link.springer.com/10.1007/978-3-319-32552-1>. [Accessed: 02-Sep-2017]
- [10] A. Gautam and S. Mohan, "A Review of Research in Multi-robot Systems," in *Industrial and Information Systems (ICIIS), 2012 7th IEEE International Conference on*, 2012, pp. 1–5 [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/6304778/>. [Accessed: 13-Sep-2017]
- [11] Z. Yan, N. Jouandeau, and A. A. Cherif, "A Survey and Analysis of Multi-Robot Coordination," *International Journal of Advanced Robotic Systems*, vol. 10, no. 12, p. 399, Dec. 2013, doi: 10.5772/57313. [Online]. Available: <http://journals.sagepub.com/doi/10.5772/57313>. [Accessed: 22-Jul-2017]
- [12] P. Lynne E., "Current Research in Multirobot Systems," *Artificial Life Robotics*, vol. 7, pp. 1–5, 2003, doi: 10.1007/s10015-003-0229-9.
- [13] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot systems: A Classification Focused on Coordination," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 2015–2028, 2004 [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/1335496/>. [Accessed: 22-Jul-2017]

- [14] V. Adolfsson, "The State of the Art in Distributed Mobile Robotics," 2001 [Online]. Available: <http://www.diva-portal.org/smash/record.jsf?pid=diva2:833625>. [Accessed: 13-Sep-2017]
- [15] C. Buratti, M. Martalò, G. Ferrari, and R. Verdone, *Sensor Networks with IEEE 802.15.4 Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011 [Online]. Available: <http://link.springer.com/10.1007/978-3-642-17490-2>. [Accessed: 17-Jul-2017]
- [16] A. D. Patel, R. H. Jhaveri, and K. J. Dangarwala, "Wireless Sensor Network – Theoretical Findings and Applications," *IJCA*, vol. 63, no. 10, pp. 25–29, 2013, doi: 10.5120/10503-5270.
- [17] Q. Wang and I. Balasingham, "Wireless Sensor Networks - An Introduction," in *Wireless Sensor Networks: Application-Centric Design*, Y. K. Tan, Ed. InTech, 2010 [Online]. Available: <http://www.intechopen.com/books/wireless-sensor-networks-application-centric-design/wireless-sensor-networks-an-introduction>. [Accessed: 04-Aug-2019]
- [18] V. Jindal and D. A. V. College, "History and Architecture of Wireless Sensor Networks for Ubiquitous Computing," vol. 7, no. 2, p. 4, 2018.
- [19] S. H. Baeg, J. H. Park, J. Koh, K. W. Park, and M. H. Baeg, "RoboMaidHome: A Sensor Network-based Smart Home Environment for Service Robots," in *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, 2007, pp. 182–187, doi: 10.1109/ROMAN.2007.4415077.
- [20] M. Waibel *et al.*, "RoboEarth," *IEEE Robotics Automation Magazine*, vol. 18, no. 2, pp. 69–82, Jun. 2011, doi: 10.1109/MRA.2011.941632.
- [21] K. Kamei, M. Sato, S. Nishio, and N. Hagita, "Cloud networked robotics," *IEEE Network Magazine*, 2012.
- [22] J. H. Ryu, M. Irfan, and A. Reyaz, "A Review on Sensor Network Issues and Robotics," *Journal of Sensors*, vol. 2015, p. 14, 2015, doi: 10.1155/2015/140217. [Online]. Available: <https://www.hindawi.com/journals/js/2015/140217/>. [Accessed: 04-Aug-2019]
- [23] Motorola Solutions, "From Cost Center to Growth Center: Warehousing 2018," Motorola, USA, White Paper, 2013 [Online]. Available: https://www.automation.com/pdf_articles/motorola/Motorola_WarehouseVision_WP_FINAL_130814.pdf. [Accessed: 06-Aug-2019]
- [24] R. B. M. de Koster, "Automated and Robotic Warehouses: Developments and Research Opportunities," *Logistics and Transport*, no. 38, 2018, doi: 10.26411/83-1734-2015-2-38-4-18. [Online]. Available: <http://doi.org/10.26411/83-1734-2015-2-38-4-18>. [Accessed: 01-Aug-2019]
- [25] J. Li, "Design Optimization of Amazon Robotics," *ACIS*, vol. 4, no. 2, p. 48, 2016, doi: 10.11648/j.acis.20160402.17. [Online]. Available: <http://www.sciencepublishinggroup.com/journal/paperinfo?journalid=134&doi=10.11648/j.acis.20160402.17>. [Accessed: 01-Aug-2019]
- [26] J. Capitán Fernández, J. R. Martínez-de-Dios, I. Maza, F. F. Ramon, and A. Ollero, "Ten Years of Cooperation Between Mobile Robots and Sensor Networks," *International Journal of Advanced Robotic Systems*, vol. 12, no. 6, p. 70, Jun. 2015, doi: 10.5772/60689. [Online]. Available: <http://journals.sagepub.com/doi/10.5772/60689>. [Accessed: 04-Jul-2019]
- [27] R. R. Selmic, V. V. Phoha, and A. Serwadda, *Wireless Sensor Networks*. Cham: Springer International Publishing, 2016 [Online]. Available: <http://link.springer.com/10.1007/978-3-319-46769-6>. [Accessed: 29-Jun-2017]

- [28] J. H. Jung, S. Park, and S. L. Kim, "Multi-robot path finding with wireless multihop communications," *IEEE Communications Magazine*, vol. 48, no. 7, pp. 126–132, Jul. 2010, doi: 10.1109/MCOM.2010.5496889.
- [29] I. Korobiichuk, Y. Danik, P. Pozdniakov, and D. Jackiewicz, "Coordination and Cooperation Mechanisms of the Distributed Robotic Systems," in *Recent Advances in Systems, Control and Information Technology*, 2016, pp. 263–272, doi: 10.1007/978-3-319-48923-0_32 [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-48923-0_32. [Accessed: 10-Oct-2017]
- [30] H. Asama, M. K. Habib, I. Endo, K. Ozaki, A. Matsumoto, and Y. Ishida, "Functional Distribution Among Multiple Mobile Robots in an Autonomous and Decentralized Robot System," in *1991 IEEE International Conference on Robotics and Automation Proceedings*, 1991, pp. 1921–1926 vol.3, doi: 10.1109/ROBOT.1991.131907.
- [31] L. E. Parker, "ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220–240, Apr. 1998, doi: 10.1109/70.681242.
- [32] S. Li, "Research on Multi-Robot Architecture and Decision-Making Model," in *Multi-Robot Systems Trends and Developments*, T. Yasuda, Ed. Online: InTech, 2011, pp. 409–436 [Online]. Available: <http://www.intechopen.com/books/multi-robot-systems-trends-and-development/research-on-multi-robot-architecture-and-decision-making-model>. [Accessed: 24-Sep-2017]
- [33] E. Gonzalez, F. D. la Rosa, A. S. Miranda, J. Angel, and J. S. Figueredo, "A Control Agent Architecture for Cooperative Robotic Tasks," in *Multi-Robot Systems, Trends and Development*, T. Yasuda, Ed. Online: InTech, 2011, pp. 491–514 [Online]. Available: <http://www.intechopen.com/books/multi-robot-systems-trends-and-development/a-control-agent-architecture-for-cooperative-robotic-tasks>. [Accessed: 24-Sep-2017]
- [34] M. A. Neumann and C. A. Kitts, "A Hybrid Multirobot Control Architecture for Object Transport," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 6, pp. 2983–2988, Dec. 2016, doi: 10.1109/TMECH.2016.2580539.
- [35] J. T. Shepard and C. A. Kitts, "A Multirobot Control Architecture for Collaborative Missions Comprised of Tightly Coupled, Interconnected Tasks," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2016, doi: 10.1109/JSYST.2016.2590430.
- [36] S. García, C. Menghi, P. Pelliccione, T. Berger, and R. Wohlrab, "An Architecture for Decentralized, Collaborative, and Autonomous Robots," in *2018 IEEE International Conference on Software Architecture (ICSA)*, 2018, pp. 75–7509, doi: 10.1109/ICSA.2018.00017.
- [37] K. Skarzynski, M. Stepniak, W. Bartyna, and S. Ambroszkiewicz, "SO-MRS: A Multi-robot System Architecture Based on the SOA Paradigm and Ontology," in *Towards Autonomous Robotic Systems*, Cham, 2018, pp. 330–342, doi: 10.1007/978-3-319-96728-8_28.
- [38] M. O. F. Sarker and T. Dahl, "Bio-Inspired Communication for Self-Regulated Multi-Robot Systems," in *Multi-Robot Systems, Trends and Development*, T. Yasuda, Ed. Online: InTech, 2011, pp. 367–392 [Online]. Available: <http://www.intechopen.com/books/multi-robot-systems-trends-and-development/bio-inspired-communication-for-self-regulated-multi-robot-sytems>. [Accessed: 24-Sep-2017]
- [39] R. Fujisawa, H. Imamura, and F. Matsuno, "Cooperative Transportation by Swarm Robots using Pheromone Communication," in *Distributed Autonomous Robotic Systems (The 10th International Symposium)*, A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt,

- M. A. Hsieh, L. E. Parker, and K. Stoy, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 559–570.
- [40] D. Jeong and K. Lee, “Distributed Communication and Localization Algorithms for Homogeneous Robotic Swarm,” in *Distributed Autonomous Robotic Systems (The 12th International Symposium)*, vol. 112, N.-Y. Chong and Y.-J. Cho, Eds. Tokyo: Springer Japan, 2016, pp. 405–418.
 - [41] S. Muhammad, M. Al-Mouhamed, and N. Mohammad, “Reliable Communication Protocol for Applications in Multi-Robot Systems,” *Arabian Journal for Science and Engineering*, vol. 41, no. 8, pp. 2771–2785, Aug. 2016, doi: 10.1007/s13369-015-2012-3. [Online]. Available: <http://link.springer.com/10.1007/s13369-015-2012-3>. [Accessed: 14-Aug-2017]
 - [42] J. Stephan, J. Fink, V. Kumar, and A. Ribeiro, “Concurrent Control of Mobility and Communication in Multirobot Systems,” *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–7, 2017, doi: 10.1109/TRO.2017.2705119.
 - [43] A. Fernandes, M. S. Couceiro, D. Portugal, J. Machado Santos, and R. P. Rocha, “Ad hoc Communication in Teams of Mobile Robots using Zigbee Technology,” *Computer Applications in Engineering Education*, vol. 23, no. 5, pp. 733–745, Sep. 2015, doi: 10.1002/cae.21646. [Online]. Available: <http://doi.wiley.com/10.1002/cae.21646>. [Accessed: 14-Aug-2017]
 - [44] S. Behera, B. Panigrahi, H. K. Rath, and A. Pal, “Wireless Characteristics Study for Indoor Multi-Robot Communication System,” in *Proceedings of the 1st Workshop on Complex Networked Systems for Smart Infrastructure*, New York, NY, USA, 2018, pp. 1–6, doi: 10.1145/3265997.3265999 [Online]. Available: <http://doi.acm.org/10.1145/3265997.3265999>. [Accessed: 01-Dec-2019]
 - [45] F. Amigoni, J. Banfi, N. Basilico, I. Rekleitis, and A. Quattrini Li, “Online Update of Communication Maps for Exploring Multirobot Systems Under Connectivity Constraints,” in *Distributed Autonomous Robotic Systems*, Cham, 2019, pp. 513–526, doi: 10.1007/978-3-030-05816-6_36.
 - [46] G. Best, M. Forrai, R. R. Mettu, and R. Fitch, “Planning-Aware Communication for Decentralised Multi-Robot Coordination,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, pp. 1050–1057, doi: 10.1109/ICRA.2018.8460617 [Online]. Available: <https://ieeexplore.ieee.org/document/8460617/>. [Accessed: 03-Dec-2019]
 - [47] E. Gerlein and E. Gonzalez, “Multirobot Cooperative Model Applied to Coverage of Unknown Regions,” in *Multi-Robot Systems, Trends and Development*, T. Yasuda, Ed. Online: INTECH Open Access Publisher, 2011, pp. 109–130 [Online]. Available: <http://www.intechopen.com/books/multi-robot-systems-trends-and-development/multirobot-cooperative-model-applied-to-coverage-of-unknown-regions>. [Accessed: 24-Sep-2017]
 - [48] E. Masehian, M. Jannati, and T. Hekmatfar, “Cooperative Mapping of Unknown Environments by Multiple Heterogeneous Mobile Robots with Limited Sensing,” *Robotics and Autonomous Systems*, vol. 87, no. Supplement C, pp. 188–218, Jan. 2017, doi: 10.1016/j.robot.2016.08.006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889015301342>. [Accessed: 22-Oct-2017]
 - [49] X. Shen, S. Pendleton, and M. H. Ang Jr., “Scalable Cooperative Localization with Minimal Sensor Configuration,” in *Distributed Autonomous Robotic Systems (The 12th International*

- Symposium*), vol. 112, N.-Y. Chong and Y.-J. Cho, Eds. Tokyo: Springer Japan, 2016, pp. 89–103.
- [50] A. G. Pires, D. G. Macharet, and C. Luiz, “Towards Cooperative Localization in Robotic Swarms,” in *Distributed Autonomous Robotic Systems (The 12th International Symposium)*, vol. 112, Tokyo: Springer Japan, 2016, pp. 105–118.
 - [51] J. Capitan, M. T. J. Spaan, L. Merino, and A. Ollero, “Decentralized multi-robot cooperation with auctioned POMDPs,” *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 650–671, May 2013, doi: 10.1177/0278364913483345. [Online]. Available: <https://doi.org/10.1177/0278364913483345>
 - [52] Z. Talebpour and A. Martinoli, “Risk-Based Human-Aware Multi-Robot Coordination in Dynamic Environments Shared with Humans,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, 2018, pp. 3365–3372, doi: 10.1109/IROS.2018.8593586 [Online]. Available: <https://ieeexplore.ieee.org/document/8593586/>. [Accessed: 03-Dec-2019]
 - [53] B. Lacerda and P. U. Lima, “Petri net based multi-robot task coordination from temporal logic specifications,” *Robotics and Autonomous Systems*, vol. 122, p. 103289, Dec. 2019, doi: 10.1016/j.robot.2019.103289. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889019302441>. [Accessed: 03-Dec-2019]
 - [54] L. Jin, S. Li, H. M. La, X. Zhang, and B. Hu, “Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach,” *Automatica*, vol. 100, pp. 75–81, Feb. 2019, doi: 10.1016/j.automatica.2018.11.001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109818305338>. [Accessed: 03-Dec-2019]
 - [55] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, “Heuristic Approaches in Robot Path Planning: A Survey,” *Robotics and Autonomous Systems*, vol. 86, no. Supplement C, pp. 13–28, Dec. 2016, doi: 10.1016/j.robot.2016.08.001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889015300671>
 - [56] S. Joyeux, R. Alami, S. Lacroix, and R. Philippsen, “A Plan Manager for Multi-robot Systems,” *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 220–240, Feb. 2009, doi: 10.1177/0278364908098370. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0278364908098370>. [Accessed: 13-May-2018]
 - [57] P. Surynek, “Multi-Robot Path Planning,” in *Multi-Robot Systems, Trends and Development*, T. Yasuda, Ed. Online: InTech, 2011, pp. 267–290 [Online]. Available: <http://www.intechopen.com/books/multi-robot-systems-trends-and-development/multi-robot-path-planning>. [Accessed: 24-Sep-2017]
 - [58] E. Masehian and D. Sedighizadeh, “An Improved Particle Swarm Optimization Method for Motion Planning of Multiple Robots,” in *Springer Tracts in Advanced Robotics*, S. B. Mikkelsen, R. Jespersen, T. Dung Ngo, A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M. A. Hsieh, L. E. Parker, and K. Stoy, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 175–188.
 - [59] R. Al-Jarrah, A. Shahzad, and H. Roth, “Path Planning and Motion Coordination for Multi-Robots System Using Probabilistic Neuro-Fuzzy,” *IFAC-PapersOnLine*, vol. 48, no. 10, pp. 46–51, Jan. 2015, doi: 10.1016/j.ifacol.2015.08.106. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896315009738>

- [60] P. K. Das, H. S. Behera, S. Das, H. K. Tripathy, B. K. Panigrahi, and S. K. Pradhan, "A Hybrid Improved PSO-DV Algorithm for Multi-robot Path Planning in a Clutter Environment," *Neurocomputing*, vol. 207, pp. 735–753, Sep. 2016, doi: 10.1016/j.neucom.2016.05.057. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0925231216304945>. [Accessed: 02-Oct-2017]
- [61] A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour, and B. Bouzouia, "Optimal Path Planning and Execution for Mobile Robots using Genetic Algorithm and Adaptive Fuzzy-logic Control," *Robotics and Autonomous Systems*, vol. 89, no. Supplement C, pp. 95–109, Mar. 2017, doi: 10.1016/j.robot.2016.12.008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889016302512>
- [62] M. Štolba and A. Komenda, "The MADLA planner: Multi-agent planning by combination of distributed and local heuristic search," *Artificial Intelligence*, vol. 252, no. Supplement C, pp. 175–210, Nov. 2017, doi: 10.1016/j.artint.2017.08.007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0004370217301042>. [Accessed: 22-Oct-2017]
- [63] C. Gao, Y. Kou, Z. Li, A. Xu, Y. Li, and Y. Chang, "Optimal Multirobot Coverage Path Planning: Ideal-Shaped Spanning Tree," *Mathematical Problems in Engineering*, 2018, doi: 10.1155/2018/3436429. [Online]. Available: <https://www.hindawi.com/journals/mpe/2018/3436429/>. [Accessed: 03-Dec-2019]
- [64] A. Mokhtar, N. Murphy, and J. Bruton, "Blockchain-based Multi-Robot Path Planning," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, 2019, pp. 584–589, doi: 10.1109/WF-IoT.2019.8767340 [Online]. Available: <https://ieeexplore.ieee.org/document/8767340/>. [Accessed: 03-Dec-2019]
- [65] E. T. S. Alotaibi and H. Al-Rawi, "A complete multi-robot path-planning algorithm," *Auton Agent Multi-Agent Syst*, vol. 32, no. 5, pp. 693–740, Sep. 2018, doi: 10.1007/s10458-018-9391-2. [Online]. Available: <http://link.springer.com/10.1007/s10458-018-9391-2>. [Accessed: 03-Dec-2019]
- [66] K. Anis, T. Sahar, and C. Imen, "Indoor Surveillance Application using Wireless Robots and Sensor Networks: Coordination and Path Planning," in *Mobile Ad Hoc Robots and Wireless Robotic Systems: Design and Implementation*, R. A. Santos, O. Lengerke, and A. Edwards-Block, Eds. IGI Global, 2013 [Online]. Available: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-4666-2658-4>. [Accessed: 24-Aug-2017]
- [67] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot Task Allocation: A Review of the State-of-the-Art," in *Cooperative Robots and Sensor Networks 2015*, vol. 604, A. Koubâa and J. R. Martínez-de Dios, Eds. Cham: Springer International Publishing, 2015, pp. 31–51 [Online]. Available: http://link.springer.com/10.1007/978-3-319-18299-5_2. [Accessed: 12-Aug-2017]
- [68] D. H. Lee, S. A. Zaheer, and J. H. Kim, "A Resource-Oriented, Decentralized Auction Algorithm for Multirobot Task Allocation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1469–1481, Oct. 2015, doi: 10.1109/TASE.2014.2361334.
- [69] E. Schneider, E. I. Sklar, S. Parsons, and A. T. Özgelen, "Auction-Based Task Allocation for Multi-robot Teams in Dynamic Environments," in *Towards Autonomous Robotic Systems*, 2015, pp. 246–257, doi: 10.1007/978-3-319-22416-9_29 [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-22416-9_29. [Accessed: 02-Oct-2017]

- [70] G. P. Das, T. M. McGinnity, S. A. Coleman, and L. Behera, "A Distributed Task Allocation Algorithm for a Multi-Robot System in Healthcare Facilities," *Journal of Intelligent & Robotic Systems*, vol. 80, no. 1, pp. 33–58, Oct. 2015, doi: 10.1007/s10846-014-0154-2. [Online]. Available: <http://link.springer.com/10.1007/s10846-014-0154-2>. [Accessed: 06-Sep-2017]
- [71] C. Wei, K. V. Hindriks, and C. M. Jonker, "Dynamic Task Allocation for Multi-robot Search and Retrieval Tasks," *Applied Intelligence*, vol. 45, no. 2, pp. 383–401, Sep. 2016, doi: 10.1007/s10489-016-0771-5. [Online]. Available: <http://link.springer.com/10.1007/s10489-016-0771-5>. [Accessed: 10-Aug-2017]
- [72] T. Morisawa, K. Hayashi, and I. Mizuuchi, "Allocating Multiple Types of Tasks to Heterogeneous Agents Based on the Theory of Comparative Advantage," *Journal of Robotics*, 2018, doi: 10.1155/2018/1408796. [Online]. Available: <https://www.hindawi.com/journals/jr/2018/1408796/>. [Accessed: 03-Dec-2019]
- [73] N. Nedjah, R. M. de Mendonça, and L. de M. Mourelle, "PSO-based Distributed Algorithm for Dynamic Task Allocation in a Robotic Swarm," *Procedia Computer Science*, vol. 51, pp. 326–335, 2015, doi: 10.1016/j.procs.2015.05.250. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S1877050915010583>. [Accessed: 10-Aug-2017]
- [74] S. Momen and A. Sharkey, "An Ant-like Task Allocation Model for a Swarm of Heterogeneous Robots," in *Proceedings of the 2nd swarm intelligence algorithms and applications symposium (SIAAS)*, 2009, pp. 31–38.
- [75] J. Li, T. Dong, and Y. Li, "Research on Task Allocation in Multiple Logistics Robots based on an Improved Ant Colony Algorithm," in *2016 International Conference on Robotics and Automation Engineering (ICRAE)*, 2016, pp. 17–20, doi: 10.1109/ICRAE.2016.7738780.
- [76] C. Liu and A. Kroll, "Memetic Algorithms for Optimal Task Allocation in Multi-robot Systems for Inspection Problems with Cooperative Tasks," *Soft Computing*, vol. 19, no. 3, pp. 567–584, Mar. 2015, doi: 10.1007/s00500-014-1274-0. [Online]. Available: <http://link.springer.com/10.1007/s00500-014-1274-0>. [Accessed: 17-Oct-2017]
- [77] X. Dai, J. Wang, and J. Zhao, "Research on Multi-Robot Task Allocation Based on BP Neural Network Optimized by Genetic Algorithm," in *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, 2018, pp. 478–481, doi: 10.1109/ICISCE.2018.00106.
- [78] C.-Y. Shih *et al.*, "On the Cooperation between Mobile Robots and Wireless Sensor Networks," in *Cooperative Robots and Sensor Networks 2014*, vol. 554, A. Koubaa and A. Khelil, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 67–86.
- [79] F. Karray, M. W. Jmal, M. Abid, M. S. BenSaleh, and A. M. Obeid, "A review on wireless sensor node architectures," in *Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), 2014 9th International Symposium on*, 2014, pp. 1–8 [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/6861346/>. [Accessed: 06-Sep-2017]
- [80] M. Rahimi *et al.*, "Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks," *Center for Embedded Network Sensing*, May 2005 [Online]. Available: <http://escholarship.org/uc/item/5pq690k8>. [Accessed: 09-Sep-2017]
- [81] "PowWow," *Wikipedia*. 02-Jun-2017 [Online]. Available: <https://en.wikipedia.org/w/index.php?title=PowWow&oldid=783424959>
- [82] M. S. Mahmoud and A. A. H. Mohamad, "A Study of Efficient Power Consumption Wireless Communication Techniques/ Modules for Internet of Things (IoT) Applications," *Advances in Internet of Things*, vol. 06, no. 02, pp. 19–29, 2016, doi: 10.4236/ait.2016.62002. [Online].

- Available: <http://www.scirp.org/journal/doi.aspx?DOI=10.4236/ait.2016.62002>. [Accessed: 14-Mar-2018]
- [83] M. Elkhodr, S. Shahrestani, and H. Cheung, "Emerging Wireless Technologies in the Internet of Things : A Comparative Study," *International Journal of Wireless & Mobile Networks*, vol. 8, no. 5, pp. 67–82, Oct. 2016, doi: 10.5121/ijwmn.2016.8505. [Online]. Available: <http://aircconline.com/ijwmn/V8N5/8516ijwmn05.pdf>. [Accessed: 14-Mar-2018]
 - [84] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of Things (IoT) Communication Protocols: Review," in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 685–690, doi: 10.1109/ICITECH.2017.8079928.
 - [85] O. O. Kazeem, O. O. Akintade, and L. O. Kehinde, "Comparative Study of Communication Interfaces for Sensors and Actuators in the Cloud of Internet of Things," *International Journal of Internet of Things*, vol. 6, no. 1, pp. 9–13, 2017.
 - [86] Elecrow, "HC-11 Wireless Serial Port Module." Elecrow Inc., 2012 [Online]. Available: <https://www.elecrow.com/download/HC-11.pdf>. [Accessed: 12-Feb-2019]
 - [87] S. Tennina *et al.*, *IEEE 802.15.4 and ZigBee as Enabling Technologies for Low-Power Wireless Systems with Quality-of-Service Constraints*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013 [Online]. Available: <http://link.springer.com/10.1007/978-3-642-37368-8>. [Accessed: 17-Jul-2017]
 - [88] S. Farahani, *ZigBee wireless networks and transceivers*. Amsterdam ; Boston: Newnes/Elsevier, 2008.
 - [89] S. Aust, R. V. Prasad, and I. G. M. M. Niemegeers, "IEEE 802.11ah: Advantages in standards and further challenges for sub 1 GHz Wi-Fi," in *2012 IEEE International Conference on Communications (ICC)*, Ottawa, ON, Canada, 2012, pp. 6885–6889, doi: 10.1109/ICC.2012.6364903 [Online]. Available: <http://ieeexplore.ieee.org/document/6364903/>. [Accessed: 17-Jun-2019]
 - [90] S. Aust, R. V. Prasad, and I. G. Niemegeers, "Analysis of the Performance Boundaries of Sub-1 GHz WLANs in the 920MHz ISM-Band," in *ISWCS*, 2013.
 - [91] Texas Instruments, "Low-Power Sub-1 GHz RF Transceiver." Texas Instruments Inc., 2019 [Online]. Available: <http://www.ti.com/lit/ds/symlink/cc1101.pdf>. [Accessed: 13-Jun-2019]
 - [92] X. Wu, X. Lu, and H. Leung, "A Video Based Fire Smoke Detection Using Robust AdaBoost," *Sensors (Basel)*, vol. 18, no. 11, Nov. 2018, doi: 10.3390/s18113780. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6263437/>. [Accessed: 09-Feb-2019]
 - [93] Y. Bian, M. Yang, X. Fan, and Y. Liu, "A Fire Detection Algorithm Based on Tchebichef Moment Invariants and PSO-SVM," *Algorithms*, vol. 11, no. 6, p. 79, May 2018, doi: 10.3390/a11060079. [Online]. Available: <http://www.mdpi.com/1999-4893/11/6/79>. [Accessed: 09-Feb-2019]
 - [94] H. Kumar, S. Bhattacharya, S. S. Thomas, S. Gupta, and K. S. Venkatesh, "Design of smart video surveillance system for indoor and outdoor scenes," in *2017 22nd International Conference on Digital Signal Processing (DSP)*, 2017, pp. 1–5, doi: 10.1109/ICDSP.2017.8096120.
 - [95] Virendra, V. Shete, and N. Ukunde, "Intelligent embedded video monitoring system for home surveillance," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, 2016, vol. 1, pp. 1–4, doi: 10.1109/INVENTIVE.2016.7823191.
 - [96] Q. Ding *et al.*, "Multi-Sensor Building Fire Alarm System with Information Fusion Technology Based on D-S Evidence Theory," *Algorithms*, vol. 7, no. 4, pp. 523–537, Oct.

- 2014, doi: 10.3390/a7040523. [Online]. Available: <https://www.mdpi.com/1999-4893/7/4/523>. [Accessed: 12-Oct-2018]
- [97] G. Marques and R. Pitarma, “An Indoor Monitoring System for Ambient Assisted Living Based on Internet of Things Architecture,” *International Journal of Environmental Research and Public Health*, vol. 13, no. 11, p. 1152, Nov. 2016, doi: 10.3390/ijerph13111152. [Online]. Available: <http://www.mdpi.com/1660-4601/13/11/1152>. [Accessed: 25-Dec-2018]
- [98] Y.-Y. Ting, C.-W. Hsiao, and H.-S. Wang, “A Data Fusion-Based Fire Detection System,” *IEICE Transactions on Information and Systems*, vol. E101.D, no. 4, pp. 977–984, 2018, doi: 10.1587/transinf.2016IIP0005. [Online]. Available: https://www.jstage.jst.go.jp/article/transinf/E101.D/4/E101.D_2016IIP0005/_article. [Accessed: 12-Oct-2018]
- [99] T. Adiono, M. Y. Fathany, S. Fuada, I. G. Purwanda, and S. F. Anindya, “A portable node of humidity and temperature sensor for indoor environment monitoring,” in *2018 3rd International Conference on Intelligent Green Building and Smart Grid (IGBSG)*, Yi-Lan, 2018, pp. 1–5, doi: 10.1109/IGBSG.2018.8393575 [Online]. Available: <https://ieeexplore.ieee.org/document/8393575/>. [Accessed: 13-Jan-2019]
- [100] R. S. Souza *et al.*, “Control of Mobile Robots Through Wireless Sensor Networks,” in *XXIX Brazilian symposium on computer networks and distributed systems*, 2011 [Online]. Available: http://www.cricte2004.eletrica.ufpr.br/anais/sbrc/2011/files/main/ST17_1.pdf. [Accessed: 14-Aug-2017]
- [101] A. Zanella and E. Menegatti, “Simultaneous Localization of Robots and Mapping of Wireless Sensor Nodes,” in *Cooperative Robots and Sensor Networks 2014*, vol. 554, A. Koubâa and J. R. Martinez-de Dios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 3–23.
- [102] A. Torres-Gonzalez, J. Ramiro Martinez-de Dios, and A. Ollero, “Robot-WSN Cooperation for Scalable Simultaneous Localization and Mapping,” in *Cooperative Robots and Sensor Networks 2014*, vol. 554, A. Koubâa and J. R. Martinez-de Dios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 25–41.
- [103] G. Tuna, V. Ç. Güngör, and S. M. Potirakis, “Wireless sensor network-based communication for cooperative simultaneous localization and mapping,” *Computers & Electrical Engineering*, vol. 41, pp. 407–425, Jan. 2015, doi: 10.1016/j.compeleceng.2014.03.003. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0045790614000573>. [Accessed: 02-Oct-2017]
- [104] H. Liang, T. Mei, and M. Q.-H. Meng, “A WSNs-based Approach and System for Mobile Robot Navigation,” in *Robot Localization and Map Building*, InTech, 2010 [Online]. Available: <https://www.intechopen.com/download/pdf/10569>. [Accessed: 24-Sep-2017]
- [105] S. Irtiza and B. Mertsching, “Surveillance System using a Mobile Robot Embedded in a Wireless Sensor Network,” in *Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, 2009, pp. 293–298, doi: 10.5220/0002215102930298 [Online]. Available: <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0002215102930298>. [Accessed: 29-Mar-2018]
- [106] J. Zhang, G. Song, G. Qiao, T. Meng, and H. Sun, “An indoor security system with a jumping robot as the surveillance terminal,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1774–1781, Nov. 2011, doi: 10.1109/TCE.2011.6131153.

- [107] G. Tuna, C. Tasdemir, K. Gulez, T. V. Mumcu, and V. C. Gungor, "Autonomous intruder detection system using wireless networked mobile robots," in *2012 IEEE Symposium on Computers and Communications (ISCC)*, 2012, pp. 000001–000005, doi: 10.1109/ISCC.2012.6249259.
- [108] O. Pascual, A. Brunete, and M. Abderrahim, "Integration of low-cost supervisory mobile robots in domestic Wireless Sensor Networks," in *2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE)*, 2014, pp. 259–264, doi: 10.1109/iCREATE.2014.6828376.
- [109] S. H. Chia, J. H. Guo, B. Y. Li, and K. L. Su, "Team Mobile Robots Based Intelligent Security System," *Applied Mathematics & Information Sciences*, vol. 7, no. 2L, pp. 435–440, Jun. 2013, doi: 10.12785/amis/072L08. [Online]. Available: <http://www.naturalspublishing.com/Article.asp?ArtcID=3232>. [Accessed: 12-Oct-2018]
- [110] A. Pennisi, F. Previtali, F. Ficarola, D. D. Bloisi, L. Iocchi, and A. Vitaletti, "Distributed Sensor Network for Multi-robot Surveillance," *Procedia Computer Science*, vol. 32, pp. 1095–1100, Jan. 2014, doi: 10.1016/j.procs.2014.05.538. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050914007388>. [Accessed: 04-Jul-2019]
- [111] A. Wichmann, B. D. Okkalioglu, and T. Korkmaz, "The Integration of Mobile (Tele) Robotics and Wireless Sensor Networks: A Survey," *Computer Communications*, vol. 51, no. Supplement C, pp. 21–35, Sep. 2014, doi: 10.1016/j.comcom.2014.06.005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0140366414002266>. [Accessed: 16-Nov-2017]
- [112] H. Elbehriy, "Developed Intelligent Fire Alarm System," *JAmSci*, vol. 8, no. 8, pp. 1016–1025, 2012.
- [113] R. B.M. de Koster, "Ware House Math," *In memoriam Jo van Nunen*, pp. 179–186, 2010 [Online]. Available: https://www.erim.eur.nl/fileadmin/default/content/erim/research/centres/material_handling_forum/research/papers/warehousemathacad%20v2.pdf. [Accessed: 09-Jun-2017]
- [114] A. Lady, "PIR Motion Sensor." Adafruit Learning System, 2018 [Online]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>. [Accessed: 09-Jun-2019]
- [115] W. Mardini, Y. Khamayseh, R. Jaradatand, and R. Hijjawi, "Interference Problem between ZigBee and WiFi," in *Proc. IACSIT*, Hong Kong, 2012, p. 7.
- [116] A. Hithnawi, H. Shafagh, and S. Duquennoy, "Understanding the impact of cross technology interference on IEEE 802.15.4," in *Proceedings of the 9th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization - WiNTECH '14*, Maui, Hawaii, USA, 2014, pp. 49–56, doi: 10.1145/2643230.2643235 [Online]. Available: <http://dl.acm.org/citation.cfm?doid=2643230.2643235>. [Accessed: 22-Aug-2019]
- [117] G. F. Roberto, K. C. Branco, J. M. Machado, and A. R. Pinto, "Local data fusion algorithm for fire detection through mobile robot," in *2013 14th Latin American Test Workshop - LATW*, Cordoba, Argentina, 2013, pp. 1–6, doi: 10.1109/LATW.2013.6562667 [Online]. Available: <http://ieeexplore.ieee.org/document/6562667/>. [Accessed: 12-Oct-2018]
- [118] R. C. Luo and K. L. Su, "Autonomous Fire-Detection System Using Adaptive Sensory Fusion for Intelligent Security Robot," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 3, pp. 274–281, Jun. 2007, doi: 10.1109/TMECH.2007.897260.

- [119] Q. Chen, A. Whitbrook, U. Aickelin, and C. Roadknight, "Data classification using the Dempster–Shafer method," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 26, no. 4, pp. 493–517, Oct. 2014, doi: 10.1080/0952813X.2014.886301. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/0952813X.2014.886301>. [Accessed: 18-Sep-2019]
- [120] F. Ye, J. Chen, and Y. Li, "Improvement of DS Evidence Theory for Multi-Sensor Conflicting Information," *Symmetry*, vol. 9, no. 5, p. 69, May 2017, doi: 10.3390/sym9050069. [Online]. Available: <https://www.mdpi.com/2073-8994/9/5/69>. [Accessed: 18-Sep-2019]
- [121] L. A. Zadeh, "Fuzzy Set Theory and Probability Theory: What is the Relationship?," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 563–566 [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_614
- [122] J. An, M. Hu, L. Fu, and J. Zhan, "A Novel Fuzzy Approach for Combining Uncertain Conflict Evidences in the Dempster-Shafer Theory," *IEEE Access*, vol. 7, pp. 7481–7501, 2019, doi: 10.1109/ACCESS.2018.2890419. [Online]. Available: <https://ieeexplore.ieee.org/document/8598705/>. [Accessed: 18-Sep-2019]
- [123] D. Andone, "Fuzzy rule base complexity reduction: a survey," in *CSCS-15, 15th INTERNATIONAL CONFERENCE ON CONTROL SYSTEMS AND COMPUTER SCIENCE*, Bucharest, Romania, 2005, vol. 1, p. 5, doi: 10.13140/RG.2.1.5021.8326 [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01246577>. [Accessed: 18-Sep-2019]
- [124] Aosong, "Temperature and Humidity Module DHT11 Product Manual." Aosong(Guangzhou) Electronics Co.,Ltd., 2018 [Online]. Available: www.aosong.com. [Accessed: 17-Jun-2018]
- [125] ADLI Logistics, "Temperature / Humidity Controlled Warehousing," 2018. [Online]. Available: <http://adlilogistics.com/temperature-humidity-controlled-warehousing/>. [Accessed: 16-Dec-2018]
- [126] Y. Ou, X. Wang, and J. Liu, "Warehouse multipoint temperature and humidity monitoring system design based on Kingview," presented at the 2017 5th International Conference on Computer-aided Design, Manufacturing, Modeling and Simulation (CDMMS 2017), Busan, South Korea, 2017, p. 040009, doi: 10.1063/1.4981605 [Online]. Available: <http://aip.scitation.org/doi/abs/10.1063/1.4981605>. [Accessed: 16-Dec-2018]
- [127] Co. Hanwei Electronics Ltd., "Technical Data MQ-2 Gas Sensor." 2016 [Online]. Available: <http://www.hwsensor.com>. [Accessed: 29-May-2018]
- [128] S. Chen, C. Hovde, K. A. Peterson, and A. Marshall, "Fire detection using smoke and gas sensors," *Fire Safety Journal*, 2007.
- [129] Yeung Yam, P. Baranyi, and Chi-Tin Yang, "Reduction of fuzzy rule base via singular value decomposition," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 2, pp. 120–132, Apr. 1999, doi: 10.1109/91.755394.
- [130] J. L. Nguyen and D. W. Dockery, "Daily indoor-to-outdoor temperature and humidity relationships: a sample across seasons and diverse climatic regions," *Int J Biometeorol*, vol. 60, no. 2, pp. 221–229, Feb. 2016, doi: 10.1007/s00484-015-1019-5. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4674394/>. [Accessed: 09-Apr-2019]
- [131] Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, Dec. 1980, doi: 10.1109/TC.1980.1675516.

- [132] R. G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," *IEEE TRANSACTIONS ON COMPUTERS*, no. 12, p. 10, 1980.
- [133] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968, doi: 10.1109/TSSC.1968.300136.
- [134] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, "Chapter 4 - Path Planning," in *Wheeled Mobile Robotics*, G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, Eds. Butterworth-Heinemann, 2017, pp. 161–206 [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780128042045000044>. [Accessed: 26-Aug-2019]
- [135] D. Ferguson, M. Likhachev, and A. Stentz, "A Guide to Heuristic-based Path Planning," School of Computer Science, CMU., 2005 [Online]. Available: https://www.cs.cmu.edu/~maxim/files/hsplanguide_icaps05ws.pdf. [Accessed: 03-Oct-2019]
- [136] A. Patel, "A* Heuristics," 2019. [Online]. Available: <http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>. [Accessed: 03-Oct-2019]
- [137] A. Apostu, F. C. Puican, G. Ularu, G. Suci, and G. Todoran, "Study on advantages and disadvantages of Cloud Computing – the advantages of Telemetry Applications in the Cloud," *RAACSDS*, 2013.
- [138] C. T. S. Xue and F. T. W. Xin, "Benefits and Challenges of the Adoption of Cloud Computing in Business," *IJCCSA*, vol. 6, no. 6, pp. 01–15, Dec. 2016, doi: 10.5121/ijccsa.2016.6601. [Online]. Available: <http://airconline.com/ijccsa/V6N6/6616ijccsa01.pdf>. [Accessed: 29-Jul-2019]
- [139] S. L. Obrutsky, "Cloud Storage: Advantages, Disadvantages and Enterprise Solutions for Business," in *Proceedings of the Eastern Institute of Technology Conference*, New Zealand, 2016, p. 10.
- [140] M. D. Ryan, "Cloud computing security: The scientific challenge, and a survey of solutions," *Journal of Systems and Software*, vol. 86, no. 9, pp. 2263–2268, Sep. 2013, doi: 10.1016/j.jss.2012.12.025. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0164121212003378>. [Accessed: 29-Jul-2019]
- [141] G. Gupta, L. P.R, and S. Sharma, "A Survey on Cloud Security Issues and Techniques," *IJCSA*, vol. 4, no. 1, pp. 125–132, Feb. 2014, doi: 10.5121/ijcsa.2014.4112. [Online]. Available: <http://www.airccse.org/journal/ijcsa/papers/4114ijcsa12.pdf>. [Accessed: 29-Jul-2019]
- [142] R. M. Jabir, S. I. R. Khanji, L. A. Ahmad, O. Alfandi, and H. Said, "Analysis of cloud computing attacks and countermeasures," in *2016 18th International Conference on Advanced Communication Technology (ICACT)*, Pyeongchang Kwangwoon Do, South Korea, 2016, pp. 1–1, doi: 10.1109/ICACT.2016.7423295 [Online]. Available: <http://ieeexplore.ieee.org/document/7423295/>. [Accessed: 29-Jul-2019]
- [143] G. Čandrlić, "Cloud Computing - Types of Cloud," 2013. [Online]. Available: <https://www.globaldots.com/blog/cloud-computing-types-of-cloud>. [Accessed: 10-Oct-2019]
- [144] D. U. Salunkhe, "A Study on the Scope of Cloud Computing in Management Education," *AIMA Journal of Management & Research*, vol. 10, no. 2/4, p. 9, 2016.

- [145] H. Yang and M. Tate, "A Descriptive Literature Review and Classification of Cloud Computing Research," *CAIS*, vol. 31, p. 2, 2012, doi: 10.17705/1cais.03102.
- [146] A. Nayyar, "The Top Open Source IoT Platforms for Developers," *Open Source For You*, 23-Oct-2018. [Online]. Available: <https://opensourceforu.com/2018/10/the-top-open-source-iot-platforms-for-developers/>. [Accessed: 07-Oct-2019]
- [147] D. Nishimura, "Understanding Preservation Metrics." Image Permanence Institute (IPI), 2018.
- [148] D. L. P. Sonntag, "Important new values of the physical constants of 1986, vapour pressure formulations based on the ITS-90, and psychrometer formulae," *Z. Meteorol*, vol. 5, no. 70, 1990.
- [149] P. H. Mitchell, "Calculating the Equilibrium Moisture Content for Wood Based on Humidity Measurements," *BioResources*, vol. 13, no. 1, pp. 171–175, Nov. 2018, doi: 10.15376/biores.13.1.171-175. [Online]. Available: <http://ojs.cnr.ncsu.edu/index.php/BioRes/article/view/12481>. [Accessed: 23-Jul-2019]
- [150] W. T. Simpson, "Equilibrium moisture content of wood in outdoor locations in the United States and worldwide," U.S. Department of Agriculture, Forest Service, Forest Products Laboratory, Madison, WI, FPL-RN-268, 1998 [Online]. Available: <https://www.fs.usda.gov/treearch/pubs/5913>. [Accessed: 23-Jul-2019]
- [151] Image Permanence Institute, "Preservation Metrics." IPI-RIT, 2018 [Online]. Available: https://www.imagepermanenceinstitute.org/webfm_send/764. [Accessed: 23-Jul-2019]
- [152] T. Padfield, "The Preservation Index and the Time Weighted Preservation Index," *Conservation Physics*, 2004. [Online]. Available: conservationphysics.org/twpi/twpi_01.html. [Accessed: 23-Jul-2019]
- [153] H. Liu, "On the Levenberg-Marquardt training method for feed-forward neural networks," in *2010 Sixth International Conference on Natural Computation*, 2010, vol. 1, pp. 456–460, doi: 10.1109/ICNC.2010.5583151.
- [154] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994, doi: 10.1109/72.329697. [Online]. Available: <http://ieeexplore.ieee.org/document/329697/>. [Accessed: 10-Oct-2019]
- [155] T. Pradeep, "Comparison of variable learning rate and Levenberg-Marquardt backpropagation training algorithms for detecting attacks in Intrusion Detection Systems," *IJCSE*, vol. 3, no. 11, p. 10, 2011.
- [156] C. Lv *et al.*, "Levenberg–Marquardt Backpropagation Training of Multilayer Neural Networks for State Estimation of a Safety-Critical Cyber-Physical System," *IEEE Trans. Ind. Inf.*, vol. 14, no. 8, pp. 3436–3446, Nov. 2017, doi: 10.1109/TII.2017.2777460. [Online]. Available: <https://ieeexplore.ieee.org/document/8119882/>. [Accessed: 25-Nov-2019]
- [157] Labcenter Electronics, "PROTEUS DESIGN SUITE: Getting Started Guide." Labcenter Electronics Ltd., 2019 [Online]. Available: <https://labcenter.s3.amazonaws.com/downloads/Tutorials.pdf>. [Accessed: 20-Oct-2019]
- [158] R. Smith, "Open Dynamics Engine," 2019. [Online]. Available: <http://www.ode.org/>. [Accessed: 21-Oct-2019]
- [159] A. F. Browne and J. M. Conrad, "A Versatile Approach for Teaching Autonomous Robot Control to Multi-Disciplinary Undergraduate and Graduate Students," *IEEE Access*, vol. 6, pp. 25060–25065, 2018, doi: 10.1109/ACCESS.2017.2689686.

- [160] NI LabVIEW, “NI LabVIEW Robotics Starter Kit: Robotics Platform for Teaching, Research, and Prototyping.” National Instruments, 2014 [Online]. Available: <http://www.ni.com/datasheet/pdf/en/ds-217>. [Accessed: 05-Aug-2019]
- [161] T. Kelemenová *et al.*, “Model Based Design and HIL Simulations,” *American Journal of Mechanical Engineering*, p. 7.
- [162] P. Sarhadi and S. Yousefpour, “State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software,” *Int. J. Dynam. Control*, vol. 3, no. 4, pp. 470–479, Dec. 2015, doi: 10.1007/s40435-014-0108-3. [Online]. Available: <http://link.springer.com/10.1007/s40435-014-0108-3>. [Accessed: 12-Nov-2019]
- [163] Arduino, “Arduino Uno Rev3,” 12-May-2019. [Online]. Available: <https://store.arduino.cc/usa/arduino-uno-rev3>. [Accessed: 12-May-2019]
- [164] Arduino, “Arduino Mega 2560 Rev3,” 12-May-2019. [Online]. Available: <https://store.arduino.cc/usa/mega-2560-r3>. [Accessed: 12-May-2019]
- [165] RAM Electronics, “Flame Sensor Module ‘Complete Module,’” *RAM Electronics*. [Online]. Available: <https://ram-e-shop.com/product/kit-flame-sensor/>. [Accessed: 18-Nov-2019]
- [166] Alibaba.com, “CC2530 2.4G Zigbee Wireless Serial Communication Module,” *Alibaba.com*, 2018. [Online]. Available: alibaba.com/product-detail/CC2530-2-4G-zigbee-wireless-serial_60612861785.html. [Accessed: 01-May-2018]
- [167] MHz Wireless Network Solutions, “Modem/Routeur VDSL2/ADSL2+ ZTE ZXHN H168N.” [Online]. Available: <https://www.mhzshop.com/shop/Reseaux/ADSL-VDSL/Modem-Routeur-VDSL2-ADSL2-ZTE-ZXHN-H168N.html>. [Accessed: 21-Nov-2019]
- [168] NodeMcu, “NodeMcu -- An open-source firmware based on ESP8266 wifi-soc.,” 2019. [Online]. Available: https://www.nodemcu.com/index_en.html#fr_54747361d775ef1a3600000f. [Accessed: 13-Aug-2019]
- [169] Digi-Key North America, “Gateways Bridge the LAN to IoT Gap | DigiKey,” *Digi-Key Electronics*, 2018 [Online]. Available: <https://www.digikey.be/en/articles/techzone/2018/feb/use-gateways-to-overcome-low-power-wireless-lan-to-iot-bridging-challenges>. [Accessed: 29-Dec-2019]
- [170] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, “IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things,” in *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, Hong Kong, China, 2010, pp. 347–352, doi: 10.1109/EUC.2010.58 [Online]. Available: <http://ieeexplore.ieee.org/document/5703542/>. [Accessed: 29-Dec-2019]
- [171] D. Min, Z. Xiao, B. Sheng, H. Quanyong, and P. Xuwei, “Design and implementation of heterogeneous IOT gateway based on dynamic priority scheduling algorithm,” *Transactions of the Institute of Measurement and Control*, vol. 36, no. 7, pp. 924–931, Oct. 2014, doi: 10.1177/0142331214527600. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0142331214527600>. [Accessed: 29-Dec-2019]
- [172] C.-T. Chang, C.-Y. Chang, K.-P. Shih, R. D. B. Martinez, P.-T. Chen, and Y.-D. Chen, “An IoT Multi-Interface Gateway for Building a Smart Space,” *JSS*, vol. 03, no. 07, pp. 56–60, 2015, doi: 10.4236/jss.2015.37010. [Online]. Available:

- <http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jss.2015.37010>. [Accessed: 29-Dec-2019]
- [173] B. Kang, D. Kim, and H. Choo, "Internet of Everything: A Large-Scale Autonomic IoT Gateway," *IEEE Trans. Multi-Scale Comp. Syst.*, vol. 3, no. 3, pp. 206–214, Jul. 2017, doi: 10.1109/TMSCS.2017.2705683. [Online]. Available: <http://ieeexplore.ieee.org/document/7931688/>. [Accessed: 29-Dec-2019]
- [174] B. Kang and H. Choo, "An experimental study of a reliable IoT gateway," *ICT Express*, vol. 4, no. 3, pp. 130–133, Sep. 2018, doi: 10.1016/j.icte.2017.04.002. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2405959516301485>. [Accessed: 29-Dec-2019]
- [175] "Enterprise IoT gateway for OEM zigbee & bluetooth gateway solution," *VOLANSYS*. [Online]. Available: <https://volansys.com/modular-iot-gateway/>. [Accessed: 02-Jan-2020]
- [176] "Cascade IoT Edge Gateway for Enterprise Applications," *Rigado*. [Online]. Available: <https://www.rigado.com/cascade-iot-gateway/>. [Accessed: 02-Jan-2020]
- [177] K. Laubhan, K. Talaat, S. Riehl, T. Morelli, A. Abdelgawad, and K. Yelamarthi, "A four-layer wireless sensor network framework for IoT applications," in *2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Abu Dhabi, United Arab Emirates, 2016, pp. 1–4, doi: 10.1109/MWSCAS.2016.7870142 [Online]. Available: <http://ieeexplore.ieee.org/document/7870142/>. [Accessed: 29-Dec-2019]
- [178] G. M. Dias, T. Adame, B. Bellalta, and S. Oechsner, "A Self-Managed Architecture for Sensor Networks Based on Real Time Data Analysis," *arXiv:1605.09011 [cs]*, Jul. 2016 [Online]. Available: <http://arxiv.org/abs/1605.09011>. [Accessed: 29-Dec-2019]
- [179] K. Jawad, K. Mansoor, A. F. Baig, A. Ghani, and A. Naseem, "An Improved three-factor anonymous Authentication Protocol for WSN s based IoT System Using Symmetric cryptography," in *2019 International Conference on Communication Technologies (ComTech)*, Rawalpindi, Pakistan, 2019, pp. 53–59, doi: 10.1109/COMTECH.2019.8737799 [Online]. Available: <https://ieeexplore.ieee.org/document/8737799/>. [Accessed: 29-Dec-2019]
- [180] N. Ahmed, D. De, and I. Hussain, "Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4890–4899, Dec. 2018, doi: 10.1109/JIOT.2018.2879579. [Online]. Available: <https://ieeexplore.ieee.org/document/8521668/>. [Accessed: 29-Dec-2019]
- [181] A. Flammini and E. Sisinni, "Wireless Sensor Networking in the Internet of Things and Cloud Computing Era," *Procedia Engineering*, vol. 87, pp. 672–679, 2014, doi: 10.1016/j.proeng.2014.11.577. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1877705814026927>. [Accessed: 29-Dec-2019]
- [182] C. Alcaraz, P. Najera, J. Lopez, and R. Roman, "Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?," in *Proc. SecIoT10*, 2010, p. 8 [Online]. Available: <https://www.nics.uma.es/pub/papers/calcaraz10.pdf>. [Accessed: 29-Dec-2019]
- [183] N. Khalil, M. R. Abid, D. Benhaddou, and M. Gerndt, "Wireless sensors networks for Internet of Things," in *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Singapore, 2014, pp. 1–6, doi: 10.1109/ISSNIP.2014.6827681 [Online]. Available: <http://ieeexplore.ieee.org/document/6827681/>. [Accessed: 29-Dec-2019]