ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

NAVIGATING MOBILE ROBOTS IN WIRELESS SENSOR NETWORKS

M.Sc. Thesis by
Mustafa Çağlar BULAN

Department : Electrical Engineering

Programme : Control and Automation Engineering

Thesis Supervisor:Prof.Dr. Hakan TEMELTAŞ

JUNE 2009

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

NAVIGATING MOBILE ROBOTS IN WIRELESS SENSOR NETWORKS

M.Sc. Thesis by
Mustafa Çağlar BULAN
(504041128)

Date of submission : **05 May 2009**
Date of defence examination: **03 June 2009**

Supervisor (Chairman) : **Prof. Dr. Hakan TEMELTAŞ (ITU)**
Members of the Examining Committee : **Prof. Dr. Metin GÖKAŞAN (ITU)**
Prof. Dr. Sema OKTUĞ (ITU)

JUNE 2009

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GEZGİN ROBOTLARIN KABLOSUZ ALGILAYICI AĞLARDA DOLANIMI

YÜKSEK LİSANS TEZİ

Mustafa Çağlar BULAN
(504041128)

Tezin Enstitüye Verildiği Tarih : 05 Mayıs 2009
Tezin Savunulduğu Tarih : 03 Haziran 2009

Tez Danışmanı : Prof. Dr. Hakan TEMELTAŞ (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Metin GÖKAŞAN (İTÜ)
Prof. Dr. Sema OKTUĞ (İTÜ)

HAZİRAN 2009

FOREWORD

I would like to express my deep appreciation and thanks for Prof. Dr. Hakan TEMELTAŞ my advisor for all of his insight and guidance throughout the progression of my study and preparation of my thesis. Also a special thank to my dear wife and my family for all moral support.

June 2009

Mustafa Çağlar BULAN
Control and Automation Engineer

TABLE OF CONTENTS

Page

ABBREVATIONS.. xi
LIST OF TABLES..xiii
LIST OF FIGURES.. xv
SUMMARY.. xvii
ÖZET...xix
1. INTRODUCTION... 1
1.1 Introduction to Study.. 1
1.2 Background.. 4
1.3 Motivations and Research Objects... 5
1.4 Thesis Organization... 6
2. WIRELESS SENSOR NETWORKS...9
2.1 Usage of Sensor Networks.. 9
2.2 Network Model...10
2.3 Constituent Layers of A WSN..16
3. A WIRELESS COMMUNCATION SPEC: ZIGBEE......................................19
3.1 The Outstanding Advantages of ZigBee...19
3.2 A ZigBee Based WSN.. 20
3.3 Application Concepts..21
3.3.1 Endpoints.. 21
3.3.2 Application framework and SAPs.. 22
3.3.3 Application framework.. 22
3.3.4 Service access points... 22
3.4 ZigBee Device Objects (ZDO)..23
3.5 ZigBee Communication Protocol and Peripheral Units...................................24
3.6 The RSS Approach.. 25
3.6.1 Parameters that affect RSS...26
3.6.2 Transmitter variability.. 26
3.6.3 Receiver variability... 27
3.6.4 Antenna orientation.. 27
3.6.5 Multi-path fading and shadowing in the RF channel.....................................27
4. LOCALIZATION...29
4.1 What Is Localization?... 29
4.2 Common Localization Methods...31
4.2.1 Probabilistic localization... 31
4.2.2 Kalman filter... 31
4.2.3 Grid based Markov localization and ML-EKF.. 31
4.2.4 Monte Carlo localization.. 32
4.2.5 Markov localization... 34
4.3 Three Questions For Robot Navigation.. 35
4.4 Why Localization Is Hard?.. 36
4.5 Previous Works.. 37
4.5.1 Some known approcahes and techniques for localization.............................. 37

4.5.2 Some methods using at localization in WSN…..38
4.5.3 Ideal localization scheme... 38
5. LOCALIZATION AND NAVIGATION IN WIRELESS NETWORKS........39
5.1 Localization by Range Approaches in WSN.. 39
5.1.1 RSSI (received signal strength indicator) approach................................ 39
5.1.2 Position estimation calculation at the sink node.......................................43
5.1.3 ToF (Time of Flight)..45
5.1.4 Using multi methods together: Calamari..45
5.1.5 Calibration...46
5.1.6 ToA (Time of Arrival).. 48
5.2 Range-Free Methods..49
5.2.1 Can any node hear me?...49
5.2.2 APIT approach... 49
5.2.3 Multi-hop... 50
5.2.4 Multi dimensional scaling..51
5.2.5 N-Hop multiliteration.. 51
5.3 Measuring Distance... 51
5.4 Determining the Direction and Angle..52
5.5 Obstacle Avoidance... 54
5.6 Heading Towards Sensor Nodes...55
5.7 PQV (Path Quality Value)... 56
6. EXPERIMENTAL STUDY and SUMILATION SOFTWARE RESULTS....59
6.1 Methodology and Purpose... 59
6.1.1 Setting up the network by hop-count numbering......................................59
6.2 Using PQV to Choose the Best Path...63
6.3 Developed Application: SOLAN...63
6.4 Communication SW and HW of Mobile Robots and Nodes........................... 69
6.4.1 Node-node and node-MR communication.. 70
6.4.2 Communication geometry... 73
6.5 Determining the Direction of Mobile Robot...75
6.6 Application Results..76
7. CONCLUSIONS and FUTURE WORKS...81
7.1 Conclusions..81
7.2 Future Works.. 83
7.3 Spinning Antenna Approach..83
REFERENCES...85
CURRICULUM VITA.. 89

ABBREVATIONS

WSN	: Wireless Sensor Networks
GPS	: Global Positioning System
RSSI	: Received Signal Strength Indicator
SS	: Signal Strength
LQI	: Link Quality Indicator
ToA	: Time of Arrival
DoA	: Direction of Arrival
AoA	: Angle of Arrival
RF/RB	: Range Free / Range Based
HA (1)	: High Availability
HA (2)	: Home Automation
OS	: Operating System
ZigBee	: A protocol name and IEEE standard for wireless communication
IEEE	: Institute of Electrical and Electronics Engineers
CIP	: Central Information Processor
POS	: Personal Operation Systems
PQV	: Path Quality Value
MR	: Mobile Robot
MV	: Mobile Vehicle

LIST OF TABLES

Page

Table 3.1: Classification of factors the affect RSS.. 26

Table 5.1: RSSI values at different positions of robot..................................... 41

Table 6.1: Received signals received by mobile robot at initial three steps..... 71

LIST OF FIGURES

Page

Figure 2.1 : A typical sensor network.. 11
Figure 2.2 : Wireless Sensor Network concepts and using areas......................... 12
Figure 2.3 : A sample Wireless Sensor Network located in ITU Campus........... 13
Figure 2.4 : Several common topologies for WSN... 15
Figure 2.5 : Communication protocols and sections... 16
Figure 2.6 : Cascading networks and subnetworks... 17
Figure 3.1 : An example ZigBee network... 20
Figure 3.2 : Physical Schema of ZigBee... 20
Figure 3.3 : Application Levels of Endpoints ZigBee... 21
Figure 3.4 : Power consumption for sensors and sensor components.................. 25
Figure 4.1 : The flow of localization.. 29
Figure 4.2 : Classification of localization algorithms.. 29
Figure 4.3 : A sample multilateration and determination of location of robot 30
Figure 4.4 : The problem situation for MCL... 33
Figure 4.5 : Trajectory of a mobile vehicle navigated by sensor nodes of WSN..35
Figure 5.1 : RSS values are inversely proportional to distance........................... 40
Figure 5.2 : The robot in a position that can be heard 3 or more nodes.............. 40
Figure 5.3 : The mobile robot in nodes' effects area... 41
Figure 5.4 : The flowchart of localization in WSN algorithm............................ 42
Figure 5.5 : The relation between signal strength and radius............................. 43
Figure 5.6 : Uncalibrated TOF readings... 47
Figure 5.7 : The illustration of ToA.. 49
Figure 5.8 : Different signal strength values received from around.................... 52
Figure 5.9 : Robot hears signal from a node but doesn't know which direction 53
Figure 5.10 : The distance between mobile robot and 3 of nodes........................ 54
Figure 5.11 : Three conditions that there is an obstacle on MR's way................. 55
Figure 5.12 : Two paths that have same HC numbering order, but have different lengths.. 56
Figure 5.13 : Determining the PQV of a path.. 57
Figure 6.1 : Initial position of the network.. 59
Figure 6.2 : View of the WSN at t instance while process continues.................. 60
Figure 6.3 : After propagation process has finished.. 61
Figure 6.4 : Robot/Vehicle tending to the destination in mapped network......... 62
Figure 6.5 : Main window of SOLAN... 64
Figure 6.6 : The legend of the network... 65
Figure 6.7 : RSSI report of WSN and instant communication data of MR......... 65
Figure 6.8 : SOLAN at runtime.. 66
Figure 6.9 : Mobile vehicle is in the range of 4 nodes.. 67
Figure 6.10 : SOLAN screen after process was completed.................................. 68
Figure 6.11 : Sensor node proximity window by mobile robot............................ 69
Figure 6.12 : Node-node and node-MV communication and central station........ 70

Figure 6.13 : ZigBee Node with antenna module.. 70
Figure 6.14 : A mobile vehicle (AGV) that can be used in an indoor WSN......... 72
Figure 6.15 : Two surfaces of system.. 73
Figure 6.16 : Communication lines in 3-dimensional volume............................. 74
Figure 6.17 : Geometrical view of MR in range of 3 nodes................................. 74
Figure 6.18 : Geometrical view of mobile vehicle with 2 antennas..................... 75
Figure 6.19 : Signal strength measurements as a function of the distance............ 76
Figure 6.20 : Standard deviation of ranges as a function of the signal strength.... 76
Figure 6.21 : The distance and RSSI values during mobile robot's travel........... 77
Figure 6.22 : The estimated and exact positions of nodes that determined by mobile robot.. 77
Figure 6.23 : The actual and robot's paths to the target position.......................... 78
Figure 6.24 : Percentage of nodes can be localized for different node density..... 78
Figure 6.25 : Average localization error by number of hops................................ 79
Figure 6.26 : Sensor Count in range and obtained accuracy of localization by time... 79

NAVIGATING MOBILE ROBOTS IN WIRELESS SENSOR NETWORKS

SUMMARY

The new age in technology provides important challenges such as humanless autonomous systems, navigation models in recent years, using in several fields.Maybe we can say the most important one of these systems is wireless sensor networks, navigation and auto-action applications, done by mobile robots or vehicles.

In a new age factory, that's possible to think all works are being done by robots which are being controlled by sensors around. Or it's came into real that your automobile is being moved driverless throught a sensor network mesh over a big city and navigated by sensors from the shortest and avaliable way. Moreover, there are hundreds of new models are making our life easier and greater applications and new ones are being developed. In military area, health industry and hospitals, nature concervancy, etc.

In this thesis, we'll discuss about localization in wireless sensor networks and navigating mobile robots in wireless sensor network. The approaches, recent works, therotical studies. Also we'll talk about the application and the simulation software (SOLAN) that was developed by us.

MOBİL ROBOTLARIN KABLOSUZ SENSÖR AĞLARDA DOLANIMI

ÖZET

Günümüzde çeşitli alanlarda insansız yapılanmalar ve insansız otomatik kontrollü sistemlerin hizmet verdiği alanlar artmaktadır. Bu yapılanmalardan belkide en göze çarpanı geniş ağlarda yürütülen araç yönetimi ve araçların yaptıkları işlerin kapsamıdır.

Yeni nesil bir fabrikada artık yük taşıyan insanlar veya araçlar, giriş çıkışları ve sayımları kontrol eden kişiler yerine tüm bu işlemleri otomatik gerçekleştiren robotlardan bahsetmemek neredeyse imkansız. Kalabalık bir metropolde trafik yoğunluğuna göre insansız yönlendirilen araçlar ulaşmak istedikleri yere trafik yoğunluğu ve açık yollar göz önünde tutularak ve en kısa yoldan otomatik olarak gidebilmesine olanak veren modeller artık çok sık konuşulur hale gelmiştir. Bunların dışında saymakla bitiremeyeceğimiz yüzlerce konuda bu modeller hayatımızı kolaylaştırmak ve daha sağlıklı sonuçlar vermek üzere tasarlanmaktadır. Harp sistemleri, sağlık kurum ve kuruluşlarındaki iş yürütmeler, ormanlarımızı ve doğayı koruma ve daha bir çoğu.

Bu tezde günümüzde giderek yaygınlaşmaya başlayan kablosuz ağlar ve robot/araç dolanımı üzerine konuşacak, çeşitli yöntemleri ve yaklaşımları değerlendirecek ve kendi hazırlamış olduğumuz yazılım (SOLAN) ve uygulama sayesinde bu sistemin nasıl işlediğini görmeye çalışacağız.

1. INTRODUCTION

1.1 Introduction to Study

Robots begin to subsitute for humans day by day at some of military issues, manufacturing and heavy industry area and even daily life in recent years. Some missions that are too heavy or more dangerous for humans are done by robots. Mobile robots are the most important study at this subject. Indoor and outdoor mobile robot systems can be deployed at various issues at factories, manufacturing buildings, great centers, military and disaster areas. The design of any successful robot involves the integration of many different disciplines, among them kinematics, signal analysis, information theory, artificial intelligence, and probability theory. Mobile robot hardware has two main parts; moving parts and the equipments for specific missions such as sensors, manipulators, holders etc.

Mobile robots can be classified as following:

- The environment in which they travel:
  - Land or home robots. They are most commonly wheeled, but also include legged robots with two or more legs (humanoid, or resembling animals or insects). (AVGs)
  - Aerial robots are usually referred to as unmanned aerial vehicles (UAVs)
  - Underwater robots are usually called autonomous underwater vehicles (AUVs)
- The device they use to move, mainly:
  - Legged robot: human-like legs (i.e. an android) or animal-like legs.
  - Wheeled robot.
  - Tracks

Followings are some examples of mobile robot using projects:

Military robots are autonomous or remote-controlled devices designed for military applications. With advances in microchip design, nanotech sciences, software architecture, and mini-power cells, robot systems can be more than just another pair of eyes. They are already being tested and used in a variety of applications. They can traverse different, even dangerous environments and perform complex tasks on their own. For example, mil-spec iRobot Packbots [41] have been used in Afghanistan to detect and map the locations and contents of caves. Mobile robots can also move to dangerous military areas that soldiers can not go or carry out hundreds of payloads to the large distance or be deployed to observe large areas. For this applications, the area that mobile robots move is dangerous Mobile robots can be used to move into the dangerous areas Such systems are currently being researched by a number of militaries. Already remarkable success has been achieved with unmanned aerial vehicles [40] like the Predator drone [39], which are capable of taking surveillance photographs, and even accurately launching missiles at ground targets, without a pilot. A subclass of these are Unmanned Combat Air Vehicles [38], which are designed to carry out strike missions in combat. Also mobile robots can help in disaster environments. Mobility and information mapping of/by information collection robots aiming at surveillance of disaster field that is inaccessible by human and at enhancement of information collection ability. Mobile robots may have intelligent sensors and portable digital assistances for collection and synthesis of disaster damage information. In addition to this missions, mobile robots can be used in many different areas.

Regardless of a robot's design or tasks, there are still three main issues with its mobility:

- Localization: How does a robot know where it is in its environment?
- Mapping: How does the robot know the details of its environment?
- Navigation: How does a robot traverse its environment?

All these above requires a well designed wireless sensor network that provides location information to the mobile vehicle and navigation as well. The wireless sensor network consists of required sensor count and capabilities due to information that is used in processes. For example if a robot should establish sensitive jobs in milimeters, it must be navigated at very high resultion and positioning process should update robot location very frequently. The design of mobile robots and the deployment area should be identified by the specifications of the desired project. Also temperature, humidity, image sensors or cameras can be included at the robots to collect the data if needed.

Two technologies in particular seem to be moving toward an interesting convergence; mobile robotics and wireless sensor networks. The two main questions here are:

- Can a mobile robot act as a gateway into a wireless sensor network?
- Can sensor networks take advantage of a robot's mobility and intelligence?

One major issue with a mobile robot acting as a gateway is the communication between the robot and the sensor network. Sensor networks typically communicate using 900 MHz radio waves. Mobile robots may use processing units or laptops that communicate via 802.11, in the 2.4- to 2.483-GHz range.

Additionally, in the area of robotics research, there have been studies on such topics as environmental information structuring and intelligent environments that examine the creation of intelligence not just in robots, but also in ambient environments (e.g. Sato *et al.*,1996). WSN technology is now the object of attention among researchers attempting to create such intelligent environments. Sensor networks has designed it's own wireless sensor protocol to meet the demanding system level requirements of our applications. The protocol and underlyling implementation have proven to be robust and very effective. Although we are surrounded by robots that we think of as automated tools, there are some sophisticated robots already in use. A remote telepresence is one of the most common applications that today's mobile, autonomous robots provide. Intelligence for these robots is handled via an embedded microcontroller that manages internal systems, and may be by a laptop that is attached to the robot. Humans control the robot through wireless communications. In this way, humans can tell the robot to change directions, shift a camera angle, take measurements, and grasp objects, and so on. For example, mobile robots can let security personnel stay in a central office and still check out unsupervised areas in a warehouse or other remote site.

At the other hand *navigation* is a significant problem in mobile robotics and ubiquitous computing. So many solutions about mobile robot navigation have been proposed. But most rely on navigating using a pre-specified map in [5], [6] then an applicable environment is limited. Some algorithms combining mobile robots with static sensor networks are proposed and maps are not needed. Error range is relatively large because distance measurement depends on RF RSS (Received Signal Strength) holding poor distance resolution. And in [4], a cost of the system is high because additional devices such as GPS module and digital compass are used.

1.2 Background

In the last 2 decay, the many potential "solutions" are roughly categorized into two groups: *relative* and *absolute* position measurements for localization and localization of mobile robots. The first includes odometry and inertial navigation; the second comprises active beacons, artificial and natural landmark recognition, and model matching. Authors compare and analyze these different methods based on technical publications and on commercial product and patent information. Comparison is centered on the following criteria: accuracy of position and orientation measurements, equipment needed, cost, sampling rate, effective range, computational power required, processing needs, and other special features.

Odometry: This method [7] uses encoders to measure wheel rotation and/or steering orientation. Odometry has the advantage that it is totally self-contained, and it is always capable of providing the vehicle with an estimate of its position. The disadvantage of odometry is that the position error grows without bound unless an independent reference is used periodically to reduce the error [28].

Inertial Navigation [30]: This method uses gyroscopes and sometimes accelerometers to measure rate of rotation and acceleration. Measurements are integrated once (or twice) to yield position. Inertial navigation systems also have the advantage that they are self-contained. For example, highly accurate gyros, used in airplanes, are inhibitively expensive. Very recently fiber-optic gyros (also called laser gyros), which are said to be very accurate, have fallen dramatically in price and have become a very attractive solution for mobile robot navigation.

Absolute Position Measurements ;

Active Beacons [31] : This method computes the absolute position of the robot from measuring the direction of incidence of three or more actively transmitted beacons.

The transmitters, usually using light or radio frequencies, must be located at known sites in the environment.

Artificial Landmark Recognition [32]: In this method distinctive artificial landmarks are placed at known locations in the environment. The advantage of artificial landmarks is that they can be designed for optimal detectability even under adverse environmental conditions.

Natural Landmark Recognition [33]: Here the landmarks are distinctive features in the environment. There is no need for preparation of the environment, but the environment must be known in advance. The reliability of this method is not as high as with artificial landmarks.

Model Matching [34]: In this method information acquired from the robot's onboard sensors is compared to a map or world model of the environment. If features from the sensor-based map and the world model map match, then the vehicle's absolute location can be estimated. Map-based positioning often includes improving global maps based on the new sensory observations in a dynamic environment and integrating local maps into the global map to cover previously unexplored areas. The maps used in navigation include two major types: geometric maps and topological maps. Geometric maps represent the world in a global coordinate system, while topological maps represent the world as a network of nodes and arcs.

Mobile robot navigation is a very diverse area, and a useful comparison of different approaches is difficult because of the lack of commonly accepted test standards and procedures. The research platforms used differ greatly and so do the key assumptions used in different approaches. Further difficulty arises from the fact that different systems are at different stages in their development. For example, one system may be commercially available, while another system, perhaps with better performance, has been tested only under a limited set of laboratory conditions. For these reasons we generally refrain from comparing or even judging the performance of different systems or techniques.

1.3 Motivations and Research Objectives

In this thesis, we aimed to research and apply a system on “navigating mobile robots in a ZigBee based wireless sensor network” including a combination of the methods and approaches told above and a new feature that was not mentioned at those projects above. Although the prototype application of our project was applied and examined in an indoor area, it is also includes features and specifications of an outdoor project as a network that has randomly distributed sensor nodes and mobile robot features.

We used hop-count method in an ad-hoc [29] network as a relation between sensor nodes constituting the network. It is the important point to let a mobile vehicle to find a target (or go to a destination position) in the wireless sensor network by the signals that collected from sensor nodes.

In addition, PQV notion was tried as a new method to create path qualities to let mobile robot knows which direction to follow. Through PQV factor, mobile robot would decide the shortest way to the target point by looking at the PQV value created by each sensor on/constituting a path. RSSI value is measured by both sensor nodes and mobile robot frequently and that helps calculating the distance between nodes. The speed of mobile robots is determined adaptively by the distance.

ZigBee spec is chosen because of the advantages that will be told in next pages as general information and advantages/disadvantages of ZigBee.

Also we developed a Windows based application to be able to simulate the navigation of a mobile robot in a sample area named SOLAN. This application simulates the navigation of a mobile robot in a sensor network and displays outputs instantly.

1.4 Thesis Organization

In this thesis, we'll talk about Wireless Sensor Networks and explain general aspects of WSNs, then what WSNs provides at the second chapter. The types of WSNs and the components of a WSN and its constraints.

In the third chapter, we'll talk about Localization problem.Localization is the main topic which answers the question "Where am I?" of a node/object in a WSN. The methods for localization can be found in this chapter. Localization is very important for good positioning results. Also we'll consider different types of environments and which localization method is feasible for which purpose. In the other hand, there will be information about Navigation. Which methods are being used and which types of communication provides a better result to decide the route of the mobile robot through the target.

At the other hand, the other important point is communication method that can be used in a WSN to localize and navigate the mobile vehicles. Today, we can say that Zigbee is a really prominent protocol because of the advantages. Zigbee protocol is a new one but comes very popular because of specifications such as needs low cost, easy to use and provides all requirements to send and receive positioning/localization infor over a definite area. That's easier to get, implement, program and use. In fourth chapter we'll talk about ZigBee.

At the fifth chapter, also we'll explain what is the purpose, methods and specifications and give information about the application. Also what was the outstanding constraints that should be considered while developing a network.

The sixth chapter includes a MS Windows based simulation application named SOLAN (Simulation of Localization and Navigation) that developed by me to simulate this experimental project that localization and navigation in a wireless sensor network with customizable parameters.

At the last chapter, we'll discuss the recent studies and a future work. We'll briefly talk about a future study which we named as "Spinning Antenna Approach".

2. WIRELESS SENSOR NETWORKS

A **wireless sensor network** (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations with the number of up to thousands. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control.

Wireless sensor networks are a trend of the past few years, and they involve deploying a large number of small nodes. The nodes then sense environmental changes and report them to other nodes over flexible network architecture. Sensor nodes are great for deployment in hostile environments or over large geographical areas.

2.1 Usage of Sensor Networks

Sensor networks have been useful in a variety of domains. The primary domains at which sensor are deployed follow:

- **Environmental observation**. Sensor networks can be used to monitor environmental changes. An example could be water pollution detection in a lake that is located near a factory that uses chemical substances. Sensor nodes could be randomly deployed in unknown and hostile areas and relay the exact origin of a pollutant to a centralized authority to take appropriate measures to limit the spreading of pollution. Other examples include forest fire detection, air pollution and rainfall observation in agriculture.
- **Military monitoring**. Military uses sensor networks for battlefield surveillance; sensors could monitor vehicular traffic, track the position of the enemy or even safeguard the equipment of the side deploying sensors.

- **Building monitoring**. Sensors can also be used in large buildings or factories monitoring climate changes. Thermostats and temperature sensor nodes are deployed all over the building's area. In addition, sensors could be used to monitor vibration that could damage the structure of a building.
- **Healthcare**. Sensors can be used in biomedical applications to improve the quality of the provided care. Sensors are implanted in the human body to monitor medical problems like cancer and help patients maintain their health.
- **Factories**. In factories human less carrier and loader vehicles can be navigated over whole factory, providing high reliability and low cost [26]

2.2 Network Model

A wireless sensor network consists of hundreds or thousands of low cost nodes which could either have a fixed location or randomly deployed to monitor the environment. Due to their small size, they have a number of limitations, an issue that I will discuss later. Sensors usually communicate with each other using a multi hop approach. The flowing of data ends at special nodes called base stations (sometimes they are also referred to as sinks). A base station links the sensor network to another network (like a gateway) to disseminate the data sensed for further processing. Base stations have enhanced capabilities over simple sensor nodes since they must do complex data processing; this justifies the fact that bases stations have workstation/laptop class processors, and of course enough memory, energy, storage and computational power to perform their tasks well. Usually, the communication between base stations is initiated over high bandwidth links.

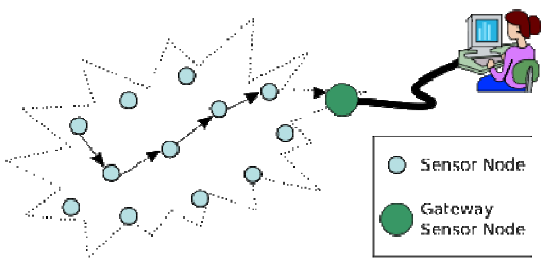


Figure 2.1: A typical sensor network

Note that one of the biggest problems of sensor networks is "power consumption", which is greatly affected by the communication between nodes. To solve this issue, aggregation points are introduced to the network. This reduces the total number of messages exchanged between nodes and saves some energy. Usually, aggregation points are regular nodes that receive data from neighboring nodes, perform some kind of processing, and then forward the filtered data to the next hop. Similar to aggregation points is clustering. Sensor nodes are organized into clusters, each cluster having a "cluster head" as the leader. The communication within a cluster must travel through the cluster head, which then is forwarded to a neighboring cluster head until it reaches its destination, the base station. Another method for saving energy is setting the nodes to go idle (into sleep mode) if they are not needed and wake up when required. Of course, the challenge is to find a pattern at which energy consumption is made evenly for all the nodes in the network.

In wireless sensor networks that we consider in this project, three main components are taken into account mainly.

1 – Sensor Nodes

2 – Central Information Processor

3 – Mobile Vehicle or Robot

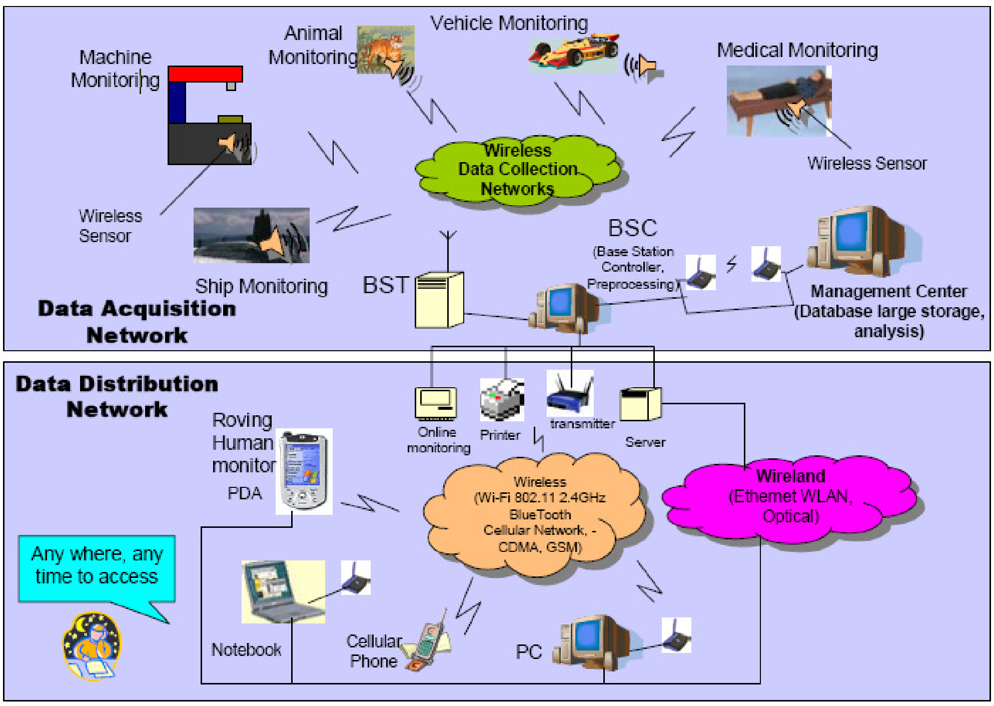


Figure 2.2: Wireless Sensor Network concepts and using areas

Main parameters which affect the quality of sensor network are number of sensors used and node density in the area, device quality for signal power, the algorithm which control and manage the network, 2 or 3 dimensional calculating capabilities and well-designed mathematical functions providing better accuracy, compact sensors that are able to locate many where needed and of course low cost.

Figure 2.2 shows the complexity of wireless sensor networks, which generally consist of a data acquisition network and a data distribution network, monitored and controlled by a management center. The plethora of available technologies makes even the selection of components difficult, let alone the design of a consistent, reliable, robust overall system.

The study of wireless sensor networks is challenging in that it requires an enormous breadth of knowledge from an enormous variety of disciplines. In this chapter we outline communication networks, wireless sensor networks and smart sensors, physical transduction principles, commercially available wireless sensor systems, self-organization, signal processing and decision-making, and finally some concepts for home automation.

(a)

(b)

Figure 2.3 : A sample Wireless Sensor Network located in ITU Ayazaga Campus

In Figure 2.3 (a), the red nodes representing the sensors distribute located to the area. This view was pictured to show an example of a WSN.

In Figure 2.3 (b), a mobile vehicle/robot needs to go to the position B from position A over wireless sensor network. When the sensor node which hears the destination position, propagates the signal to all network and the sensor node which can hear the robot sends the information that “ROBOT-N is HERE”. The calculation process is done by central information processor (CIP) and draws the route over the entire wireless sensor network.

In addition to one or more sensors, each node in a sensor network can be typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery. The envisaged size of a single sensor node can vary from shoebox-sized nodes down to devices the size of grain of dust, although functioning 'motes' of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few cents, depending on the size of the sensor network and the complexity required of individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth [12]

A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station).

In computer science and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year.

Unique characteristics of a WSN include:

- Limited power they can harvest or store
- Ability to withstand harsh environmental conditions
- Ability to cope with node failures
- Mobility of nodes
- Dynamic network topology
- Communication failures
- Heterogeneity of nodes
- Large scale of deployment
- Unattended operation

Sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. They usually consist of a *processing unit* with limited computational power and limited memory, *sensors* (including specific conditioning circuitry), a *communication device* (usually radio transceivers or alternatively optical), and a power source usually in the form of a battery. Other possible inclusions are energy harvesting modules, secondary ASICs, and possibly secondary communication devices (e.g. RS-232 or USB).

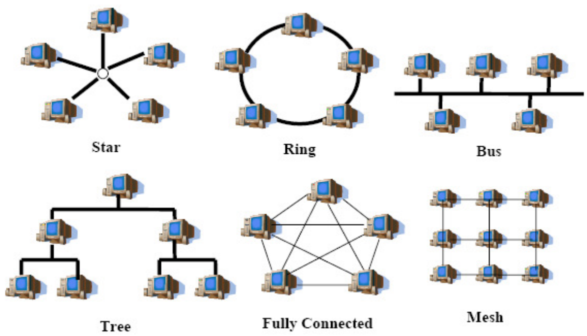


Figure 2.4*:* Several common topologies for WSN

The base stations are one or more distinguished components of the WSN with much more computational, energy and communication resources. They act as a gateway between sensor nodes and the end user.

The protocols which can be used in wireless sensor networks are varying by purposes of the projects. In large scales of terrains, higher data rate based technologies are used. High data rate based protocols and equipments provide more robust networks against to worse environmental conditions. Also signals between nodes can be transmitted over kilometers. But this technologies will be normally more expensive and harder to be built and maintenance.

Also depending on the area type and conditions, we will decide what the network type will be such as tree, mesh, star, etc. (shown in Figure 4 above)

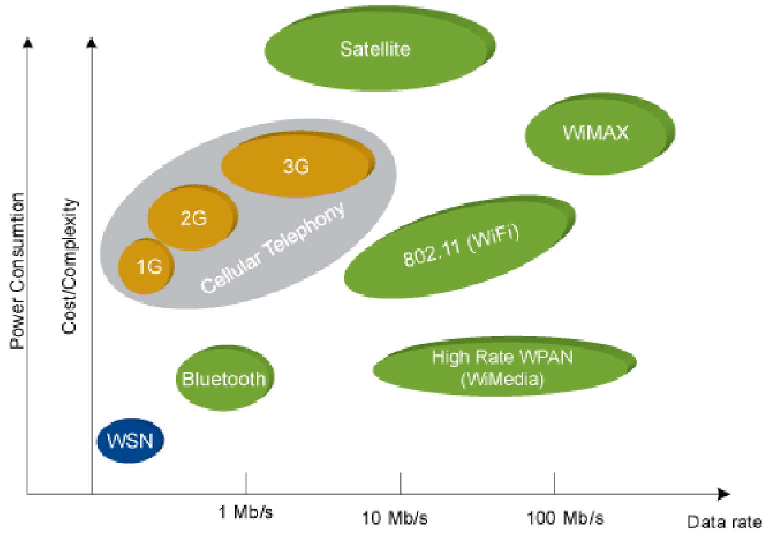


Figure 2.5: Communication protocols and sections

2.3 Constituent Layers of a WSN

Physical Layer: responsible for frequency selection, modulation and data encryption.

Data-link Layer: responsible for multiplexing of data streams, data frame detection, Medium Access control (MAC) and error control.

Network Layer: route the data supplied by the Transport layer, special multi-hop wireless routing protocols between sensor nodes and sink nodes.

Transport Layer: maintain the flow of data if the Application Layer requires it. Needed if End-User accesses the Sensor Network through the Internet.

Application Layer: Makes the hardware and software of the lower layers transparent to the End-User.

It would be better to make some explanations for the layers that should be known for the project in this thesis to be able to understand what could be next levels of this investigations and studies.

Physical Layer can have many kinds of devices varying by purpose of use. The main points that developer needs to consider are; power consumption of devices, robustness for environmental conditions, indoor or outdoor specifically, purpose of long term or short term using (because of maintenance or replication of components of devices)

Data link layer subject almost combined to physical layer subject if you are talking about a typical sensor network. Because the devices used in physical layer also involves its own northbound and southbound interfaces and components as well.

Again we can discuss the "network layer" and "transport layer" together. These layers are maybe the most important point after physical layer which provides better performance and to carry our soft-application/algorithms. Once again, the type of the network+transport layer will be choose by developer.

Wireless Sensor Networks became more important in recent years in robotic industry with high capability components were developed.

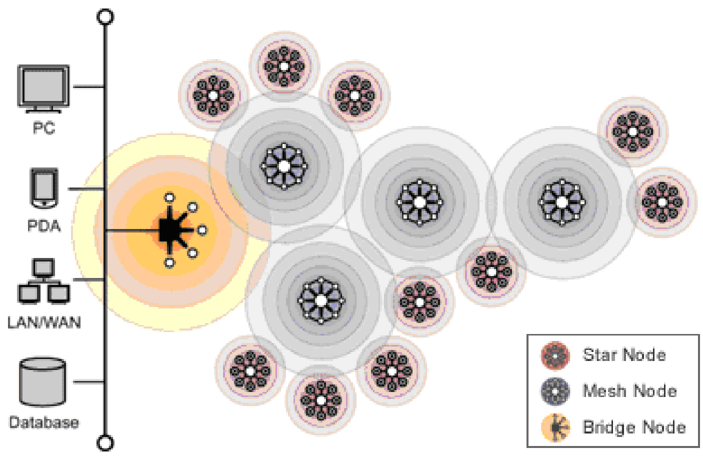


Figure 2.6: Cascading networks and sub networks

Sometimes it can be designed as cascaded network topologies including different types of networks. Every sub-network has a main communicator as an interface to other networks.

Each localization processes are applied in its own sub-network and all calculations are sent to upper level. Different approaches can be used in different sub networks but providing similar output.

3. A WIRELESS COMMUNICATION SPEC FOR WSN: ZIGBEE

ZigBee is considered as the communication protocol in this thesis. ZigBee protocol was preferred because of:

Standards based

Low cost

Can be used globally

Reliable and self healing

Supports large number of nodes

Easy to deploy

Very long battery life

Secure

3.1 The Outstanding Advantages of ZigBee

Home Automation [HA]: Defines set of devices used in home automation as Light switches, thermostats, window shade, heating unit

Industrial: Plant Monitoring as Consists of device definitions for sensors used in industrial control, Temperature, Pressure sensors, Infrared, etc.

Health and Hospitals: Patients receive better care at reduced cost with more freedom and comfort:

- Patients can remain in their own home

    Monitors vital statistics and sends via internet, Doctors can adjust medication levels

- Allows monitoring of elderly family member

Sense movement or usage patterns in a home, turns lights on when they get out of bed, notify via mobile phone when anomalies occur, wireless panic buttons for falls or other problems can also be used in hospital care

- Patients are allowed greater movement, reduced staff to patient ratio

3.2 A ZigBee Based WSN

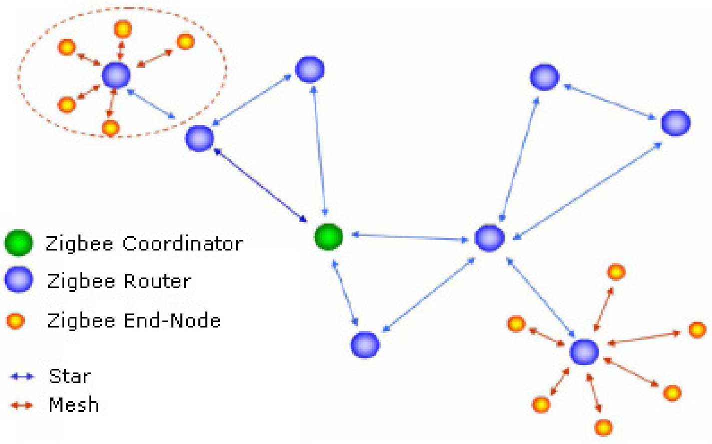


Figure 3.1: An example ZigBee network

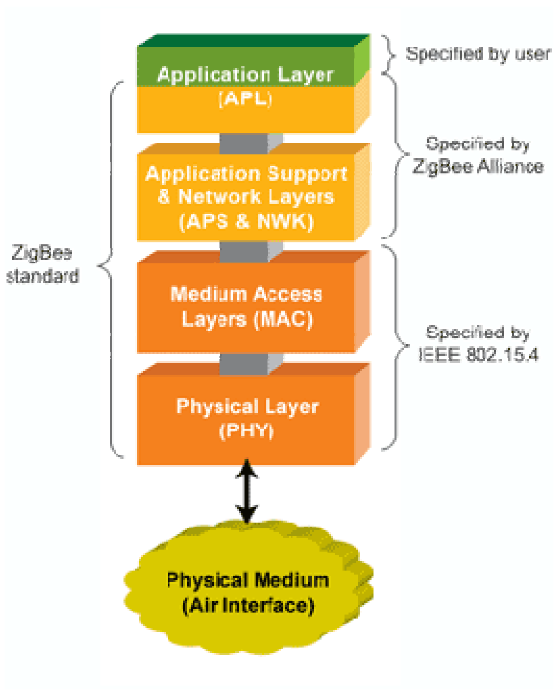


Figure 3.2: Physical schema of ZigBee

3.3 Application Concepts

This page introduces some concepts at the Application level that will help you to understand the more detailed software architecture presented in the rest of this module.

3.3.1 Endpoints

A node may have several applications running on it - for example, a node in an environment monitoring network may be measuring temperature and humidity, each of which is an application. These application instances on a node are said to be **endpoints**, where messages can originate and terminate.

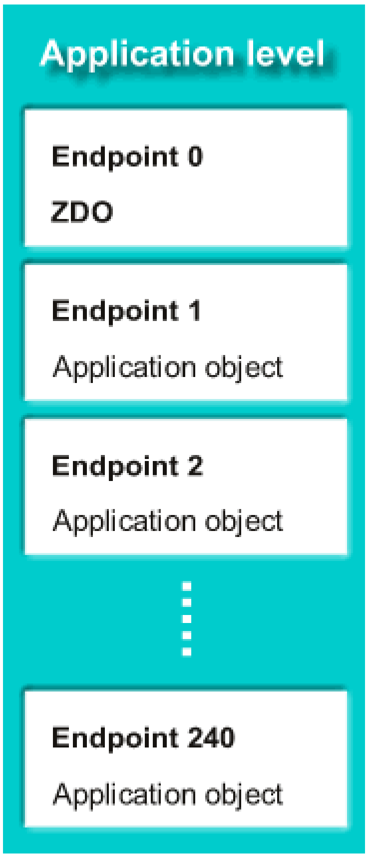


Figure 3.3: Application levels of endpoints ZigBee

In order to route messages arriving at the node to the appropriate application, each application on the node must be uniquely identified and is given an endpoint address. Endpoint addresses for user applications are numbered from 1 to 240. Therefore, to identify a particular application instance in a ZigBee network, you need to supply the relevant network address and then the required endpoint address on the node.

Note: Endpoint address 255 can also be used. This is the broadcast endpoint address - the same data can be sent to all applications on a node by sending the message to this endpoint address.

3.3.2 Application framework and SAPs

This part describes the Application Framework and associated SAPs (not shown in the architecture diagram on the previous page).

3.3.3 Application framework

The Application Framework (AF) contains the application objects and facilitates interaction between the applications and the APS layer. An application object interacts with the APS layer through an interface known as a Service Access Point (SAP).

3.3.4 Service access points

A Service Access Point (SAP) implements a set of operations to pass information and commands between layers. There are usually four types of operation implemented by a SAP:

Request: Typically, a layer using the services of another layer generates a Request to the lower layer.

Confirm: In general, the lower layer responds with a Confirm, which indicates whether it has accepted or rejected the request. A rejection could occur if the Request is invalid or the layer does not implement the operation concerned (the operation could be defined as optional).

Response: Normally, Requests result in some sort of Response from the lower layer. This may be a simple status message indicating that the Request has been performed, or may contain further information that the Request has asked for. Responses can be immediate or delayed:

- **Synchronous Response**: Responses may be generated immediately after the Request has been issued - for example, if the information or command is available on the local node.

- **Asynchronous Response**: A Request may require messages to be sent over the network to a remote node, in which case there will be a delay between issuing the Request and the arrival of the Response.

The SAP mechanism allows both types of Response to be handled and delivered to the higher layer.

Indication : An Indication is generated when the lower layer has unsolicited information or commands to be delivered to the higher layer, possibly as a result of a Request from a remote node for local information.

3.4 ZigBee Device Objects (ZDO)

Endpoint address 0 on each node is reserved for a special application called the **ZDO** (ZigBee Device Objects). This application has a number of roles, including defining the type of node (coordinator, Router or End Device), initializing the node and participating in network creation.

Heily (2004) defines ZigBee as “a rapidly growing, worldwide, non-profit industry consortium” whose mission is “to define a reliable, cost-effective, low-power, wirelessly networked, monitoring and control product based on an open global standard, illustrates the areas of interest for different wireless communication standards.

ZigBee, a new standard which became publicly available in June 2005, is based on the IEEE 802.15.4 standard. It expands the IEEE 802.15.4 by adding the framework for “the network, security and application”.

Craig (2005) mentions three networking topologies that the standard covers: the star, mesh, and cluster tree. The ZigBee standard works on top of the IEEE 802.15.4 addressing schema by using the standard 64-bit and the short 16-bit addressing. Kinney (2005) summarizes the ZigBee network layer responsibilities: the successful establishment of a new the network, and successful new device configuration, addressing assignment, network synchronization, frames security, and message routing.

ZigBee further distinguishes the concept of the physical devices (RFD, FFD) by using the notion of “logical devices.” “ZigBee Coordinator” is the first type of logical devices. It is responsible for initializing, maintaining, and managing the network. Under the coordinator in the network hierarchy is the “ZigBee router,” which is responsible for controlling the message routing between the nodes. Finally, the “ZigBee End Device” acts as the end point of the network structure.

The ZigBee specifications (2005) summarize the security services provided by ZigBee: “key establishment, key transport, frame protection, and device management.” ZigBee builds its security mechanism using symmetric key cryptography. The security services also depend on the associated layer, the security mechanism covers the network and the application layer. In addition, if a MAC frame needs security protection, the MAC layer is able to secure it. Moreover, the notion of end-to-end security is supported; the source and destination devices have access and use the same share key.

3.5 ZigBee Communication Protocol and Peripheral Units

ZigBee is a new low rate wireless network standard defined by the ZigBee Alliance and based on the IEEE 802.15.4. The standard is aiming to be a low-cost, low power solution for systems consisting of unsupervised groups of devices in houses, factories and offices. Expected applications for the ZigBee are building automation, security systems, remote control, remote meter reading and computer peripherals. The wireless spectrum in terms of two key performance characteristics wireless radio range and data transmission rate. Contrasted with other wireless protocols such as Bluetooth, 802.11, and 802.15.3, ZigBee shows a wide range in communication distance and excellent ability in low rate transmission, for example the quick, short text transmission [8]

ZigBee takes full advantage of a powerful physical radio specified by IEEE 802.15.4. As be showed in Fig. 2, the IEEE defines only the Physical (PHY) and Medium Access Control (MAC) layers in its standards.

Looking at ZigBee the key additions or differences in terms of the alliance mission statement are low power, networked, and open standard. The 802.15.4 standard also speaks of a ‘Personal Operating Space’ (POS) and 10m range but recognizes the possibility for greater range at lower data rates. So with the drivers of simplicity, long battery life, networking capabilities, reliability, and low cost, ZigBee should be widely used in building automation, personal health care, consumer electronics, PC & peripherals, industrial control, residential/light and commercial control

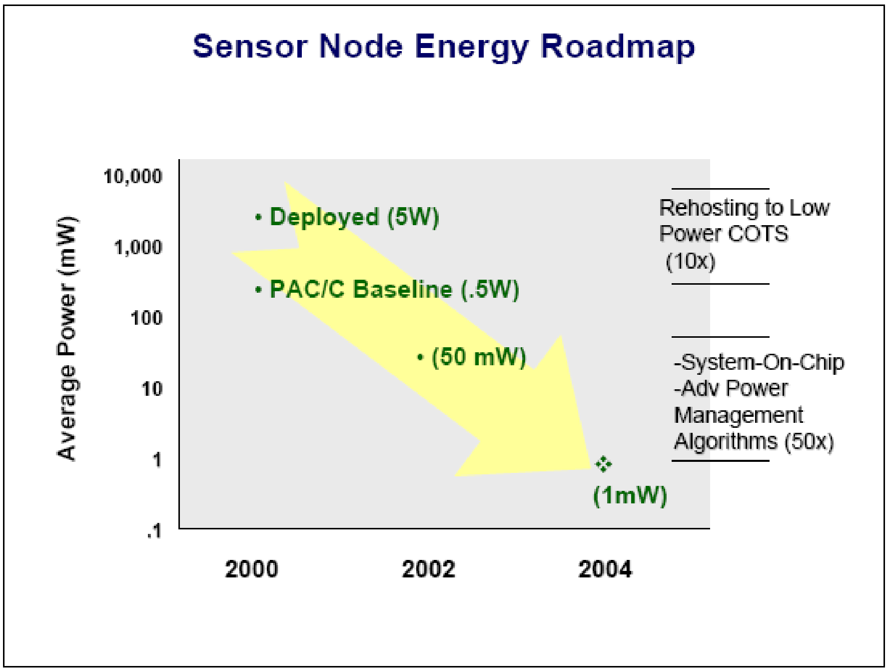


Figure 3.4: Power consumption for sensors and sensor components

ZigBee protocol ise really useful for sending text packets to other nodes. The carrier frequencies and frame type is designed for character strings with parity codes in one packet. That's also easy to parse the packet by receiver nodes and processors. No complex codec or similar algorithms that take more time are required.

3.6 The RSS Approach at ZigBee

Majority of the existing methods leverage the existence of IEEE 802.11 base stations with powerful radio transmit powers of approximately $100mW$ per base station. Such radios are in a different class from the low power IEEE 802.15.4 compliant radios that typically transmit at low power levels ranging from $52mW$ to $29mW$. The wide availability of larger number of IEEE 802.15.4 radios in the research community has revived the interest for signal strength based localization in sensor network. Despite of rapidly increasing popularity of IEEE 802.15.4 radios and signal strength localization, there is a lack of detailed characterization of the fundamental factors contributing to large signal strength variation.

3.6.1 Parameters that affect RSS

Different factors that affect the indoor RSS are the user's presence/absence, his/her orientation, time of the day, building type and material, distance from transmitter, and type of radio [17]. These factors are by no means exhaustive. Other possible factors are antenna orientation, directionality and type. Table 3.1 shows the classification of the parameters which affect RSS.

Table 3.1: Classification of factors that affect RSS

Effect On	Factors	Options
Data Collection	1.Proximity of user	User's presence or absence
	2. Orientation of user and terminal	North,East,South, West
	3. Make of Radio	
Statistics	4. Time of measurement	Time of day & days of week
	5. Period of measurement	Second, minute, hour
	7. Interference	Co-channel/adjacent co-channel
	8. Building environment	Small offices or large halls

3.6.2 Transmitter variability

Different transmitters behave differently even when they are configured exactly in the same way. In practice, this means that when a transmitter is configured to send packets at a power level of d dBm then the transmitter will send these packets at a power level that is very close to d dBm but not necessarily exactly equal to d dBm. This can alter the received signal strength indication and thus it can lead to inaccurate distance estimation.

3.6.3 Receiver variability

The sensitivity of the receivers across different radio chips is different. In practice, this means that the RSSI value recorded at different receivers can be different even when all the other parameters that affect the received signal strength are kept constant.

3.6.4 Antenna orientation

Each antenna has its own radiation pattern that is not uniform. In practice, this means that the RSSI value recorded at the receiver for a given pair of communicating nodes and for a given distance between them varies as the pair wise antenna orientations of the transmitter and the receiver are changed.

3.6.5 Multi-path fading and shadowing in the RF channel

In indoor environments the transmitted signals get reflected after hitting on the walls and/or on other objects in the room such as furniture. Both the original signal and the reflected signal reach the receiver almost at the same time since they both travel at the speed of light. As a result of this, the receiver is not able to distinguish the two signals and it measures the received signal strength for both of them.

4. LOCALIZATION

4.1 What Is Localization?

Localization answers the question: ***"Where am I?"***

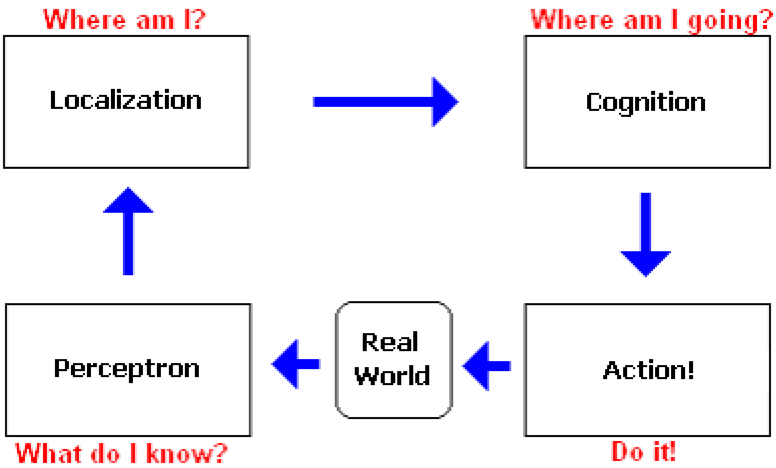


Figure 4.1: The flow of localization

Localization is the process of determining where a robot is in its environment. It could be an absolute position in the world (e.g. latitude and longitude), a position in a map (e.g. coordinates x, y), relative to the start position, relative to some landmark (e.g. at the kitchen doorway), relative to another robot, or some other way of identifying the robot's current position.

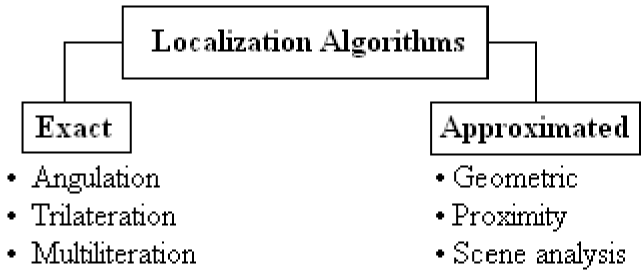


Figure 4.2: Classification of localization algorithms

The way of reporting position determines how it can be used by the robot's programs.

Several methods can be used for network localization:

– Constraint satisfaction/optimization (centralized)

– Joint estimation using ranging estimates (centralized)

– Multihop distance estimation (distributed)

– Iterative localization (distributed)

– Potential fields (distributed)

Localization can be done with or without a map of the environment. A map contains information about the environment that the robot can use to help localize itself. If you are in the middle of the woods, you might not know exactly where you are. But, if some other point identified on the map. Once you and such a point, you might be able to identify where you are in the woods.

Localization is the main problem for navigating an object in a sensor network. The best localization method provides the best input for an algorithm for navigation. Both sensor and mobile vehicle/robot need to be localized in the sensor network.

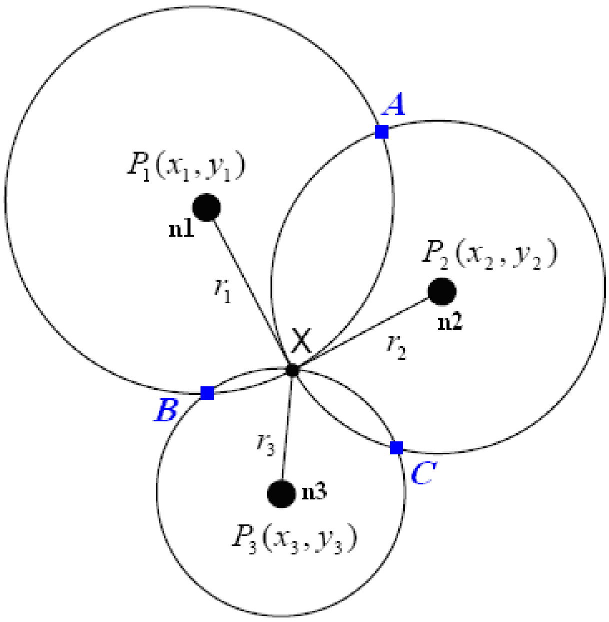


Figure 4.3 : A sample multilateration and determination of location of mobile robot.

$$\sqrt{(x_1 - x)^2 + (y_1 - y)^2} = |\vec{r_1}| \tag{4.1}$$

$$\sqrt{(x_2 - x)^2 + (y_2 - y)^2} = |\vec{r_2}| \tag{4.2}$$

$$\sqrt{(x_3 - x)^2 + (y_3 - y)^2} = |\vec{r_3}| \tag{4.3}$$

4.2 Common Localization Methods

4.2.1 Probabilistic localization

In probabilistic terms, localization is the process of determining the probability of the robot being at pose *l* given sensor inputs s_n and executed actions an (n=1...N) By assuming independence between observations and between executed actions, the following update rules can be formulated in the Markov localization framework Depending on the application and the type of robot, different motion models can be considered.

4.2.2 Kalman filter

Basically, a Kalman Filter is a recursive data processing algorithm that estimates the state of a noisy linear dynamic system. Kalman filtering emerges once representing all densities by Gauss. Kalman filtering has been successfully applied for mobile robot localization in many systems [19]. The inherent problem in this approach is that only one pose hypothesis can be represented making the method in general unable to globally localize the robot or to recover from total localization failures.

4.2.3 Grid based markov localization and ML-EKF

If p(*l*) is represented by a piece-wise linear function, we obtain grid-based Markov localization. Advantages of this method are its global search space and flexibility for different motion and sensor models. Depending on the dimension, resolution and size of the grid, the method might not be feasible for real-time applications without further optimizations.

The basic idea is that Markov localization is used for global search of the robot position providing high robustness on sensor noise and fast recovering from manual robot displacement, whereas Kalman filtering is used locally for precisely computing the robot pose.

A 2D Markov localization grid at coarse resolution represents Possible *(x,y)* positions of the robot but does not contain information about orientation. Because of being 2D, on observations only the distance to landmarks are considered and on motion all directions are treated with equal probability.

Landmark observations are first integrated into the Markov grid. If the given observation is plausible based on the Markov state, it is also integrated into the EKF. The plausibility check examines $p(s_n \mid \sim l)$, where $\sim l$ is the maximum likely cell in the grid. If this probability is smaller than a threshold t_{obs}, the observation is rejected for the EKF. After accepting and integrating an observation into the EKF, the distributions of Markov grid and EKF are compared using x_2 test. If the test exceeds a threshold t_x^2, the Kalman filter is reinitialized with $\sim l$ and the maximum likely orientation computed by projecting the last unfiltered observation to $\sim l$.

The output of the ML-EKF system is the EKF state. A limitation of this approach is that integration of observations into the 2D Markov grid must be feasible.

4.2.4 Monte Carlo localization

Monte Carlo Localization, also known as Particle Filtering, is a relatively new approach to the problem of robot localization - estimating a robot's location in a known environment, given its movements and sensor reading over time. Imagine a project that you are to solve the global localization problem, where the robot does not know its starting position but needs to figure out where it is. (This is in contrast to the position tracking problem, where the robot knows its starting position and just need to accommodate the small errors in its odometry that build up over time.)

To make things a bit simpler, you will solve this problem in a one dimensional world. Since the on-board computation abilities of the RCX are limited, we remote control the robot from a base computer. You are given skeleton programs for the robot and base computer, which are described below.

Imagine a long hallway with a set of open doors along one side. The doors are distributed unevenly along it, and the doors are not all of the same width. Your robot can move back and forth along the hallway, and at anytime it is either in front of one of the doors or is along a wall segment. The situation might look like this:

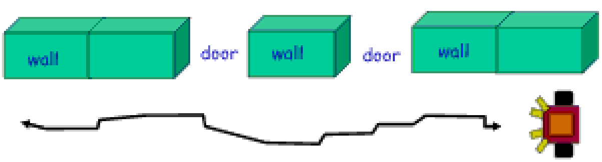


Figure 4.4 : The problem situation for MCL

The robot's task is move to a predefined goal point along the hallway. But it doesn't know its starting point. It does have a sonar unit that is aimed at the wall/doors. The unit is reasonably reliable, and can be used to determine whether the robot is currently in front of a wall or a door. To solve this problem, the robot repeated moves forward a fixed distance and takes a sonar reading. If it thinks it has reached the end of the hallway, it starts backing up instead of moving forward. It may need to move back and forth along the hallway a few times before it accurately knows where it is, but usually it will be able to reliably determine its location in one pass back and forth [2]

Not only does the robot not know its starting position, but there are two additional problems that need to be taken into consideration:

- The sonar reading may be wrong (e.g., the reading indicates an open door when in front of a wall segment).
- The distance the robot attempts to move may not the actual distance it moved (e.g. when it attempts to move 2 inches forward it really just moves 1.9 inches).

So you will need to develop probabilistic models for both of these sources of error.

Here is the MCL algorithm (based on the one presented by Dieter Fox in his paper listed below) for generating at time t the next set of samples S_{t+1} from the current set S_t. The x_t are the locations and the w_t the probabilities - so an (x_t, w_t) pair represents a sample. The distance traveled is u_t, and the sensor reading is z_t. We use superscripts to indicate the individual samples, so $x_t^{(i)}$ is the location of sample i at time t. n is the number of samples.

```
inputs:
Distance ut, sensor reading zt.,sample set St ={(xt(i),wt(i))|
i=1,...,n}
for i = 1 to n do        // First update the current set of samples
i.  xt  = updateDist(xt, ut)          // Compute new location
ii.  wt(i) = prob( zt | xt(i)  )          // Compute new probability
St+1 = null     // Then resample to get the next generation of
samples
for i = 1 to n do
```

```
i. Sample an index j from the distribution given by the weights in St
ii. Add (xt(j), wt(j)) to St+1    // Add sample j to the set of new samples
return St+1
```

4.2.5 Markov localization

Special case of probabilistic state estimation applied to mobile robot localization

a. Initial Hypothesis: Static Environment;

- Markov assumption and the robot's location is the only state in the environment which systematically affects sensor readings

b. Further Hypothesis: Dynamic Environment

c. Instead of maintaining a single hypothesis as to where the robot is, Markov localization maintains a probability distribution over the space of all such hypothesis and also Markov Localization uses a fine-grained and metric discretization of the state space.

Markov Localization has *Topological* (landmark-based, state space organized according to the topological structure of the environment) and Grid-Based (the world is divided in cells of fixed size; resolution and precision of state estimation are fixed beforehand). The latter suffers from computational overhead. Following is the notation of Markov Localization:

$\boldsymbol{l} = \langle x,y,z \rangle$ (location of robot) **(4.4)**

$\boldsymbol{l}_t$ is robot's true location at time t, L_t is random variable that expresses that robot's location and Bel(Lt) is robot's position belief at time t. $Bel(L_t=\boldsymbol{l})$ is the probability (density) that the robot assigns to the possibility that its location at time ***t*** is ***l***. The belief is updated in response to two different types of events: sensor readings and odometry data.

$d = \{ d0,d1, \ldots ,d_T\}$ **(4.5)**

$d_i = \{ a_i \text{ if odometry readings } | \ s_i \text{ if environment sensor readings } \}$ **(4.6)**

And the goal is, (by estimating the posterior distribution over LT conditioned on all avaliable data) :

$$P(L_T = l \mid d) = P(L_T = \boldsymbol{l} \mid d0, \ldots, d_T) \quad \textbf{(4.7)}$$

4.3 Three Questions for Robot Localization and Navigation

Robot navigation is the task of an autonomous robot to move safely from one location to another. The general problem of navigation can be formulated in three questions. "**Where am I**"?

The robot has to know where it is in order to make useful decisions. Finding out the whereabouts of the robot is called robotic localization. **Where am I going?** In order to full some task the robot has to know where it is going. It has to identify a goal and this problem is therefore known as goal recognition.

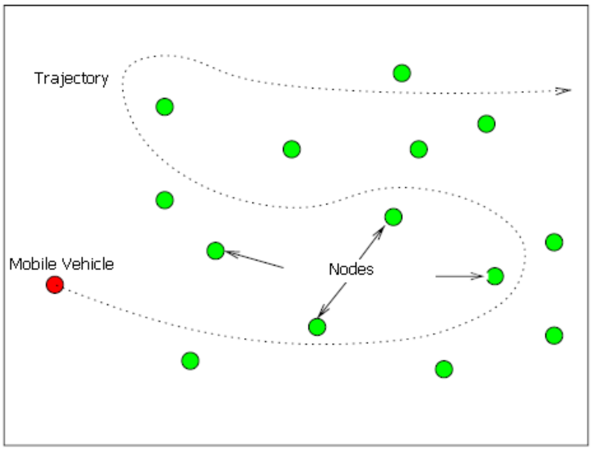


Figure 4.5 : Trajectory of a mobile vehicle navigated by sensor nodes of WSN

How do I get there? Once the robot knows where it is and where it has to go it has to decide on how to get there. Finding a way to get to the goal is known as path planning [20].

An interesting question is **"What is the optimum robot trajectory and when should the robot packets be sent?"** Notice that the problem is quite difficult since the position of the unknown nodes is not known *a priori*. Once the nodes are (at least partially) localized, the mobile robot can be steered to assist nodes with large uncertainties. We will not try to answer the optimality question in this thesis. Instead, we will make some remarks regarding some properties that the trajectory should have.

First, a node is best localized if the mobile robot trajectory is close to that node. This observation stems from the calibration data; the standard deviation for close range measurements is significantly lower than for higher ranges. Therefore, the trajectory of the mobile robot node should pass closely to as many potential node positions as possible.

Second, even if the mobile robot passes close to a node in a straight line, the proposed localization algorithm will not be able to determine on which *side* of the line the node lays. Indeed, positions symmetric to the line will be equally probable. The position estimate after four approximately collinear beacon packets have been received by the unknown node. To eliminate one of the candidates, at least one non-collinear beacon packet must be received.

Therefore, the mobile robot trajectory should be designed in such a way that all possible positions are fully covered by at least three non-collinear beacons, and the "grid" formed by the mobile robots should be as tight as possible (to increase precision).

4.4 Why Is Localization Hard?

A robot often needs to know its location to know how to act. Without knowing where it is, otherwise simple tasks such as "go to the doorway and beep" become impossible. How is the robot going to know when to beep if it does not know when it has reached the doorway? When a robot starts to do more complicated tasks, knowing where it is becomes even more important. You cannot have a robot go to a location if you do not know where the target location is, where the robot is, and where the robot is relative to the target location.

Localization is a big challenge and is one that current robot researchers are working on. Throughout the day you likely know where you are most of the time. You can look around the environment and make some observations based on past experiences. You may also know what caused you to get to that location. This is the history of your path. However, if you were dropped in the middle of an area that you had never seen before, you would probably have no idea where you were. You could start to walk around until you build some knowledge about the environment, or you could pull out a map and try to figure out where you are in the map based on observations.

A robot needs to do something similar. The problem is that the sensors a robot has are often not entirely accurate. Consider a sonar sensor. A sonar sensor will tell us the distance to some object by sending out a signal and measuring the time for it to react of the object and return. However, the sensor readings are noisy due to the physics of the signal. Some objects will absorb the signal. Sometimes the signal will bounce of a corner in such a way that it appears to be much further away. We will not get into the details of sonar signals here, but the important point is that a sensor does not always tell us exactly what we want to know. Furthermore, a sensor does not often tell the robot what it wants to know. The robot wants to know where it is. A sensor such as a sonar sensor tells the robot a distance to the nearest object in a particular direction. Localization software needs to take the raw sensor readings and date.

4.5 Previous Works

First ideas for localization in WSN have been developed in recent years mostly for military using. Numerous studies have been performed since then for a civil use. Researchers have pointed out the influence of noise on the localization process and the importance of various system parameters on the accuracy and efficiency of the localization process, but there is no consensus of a single best algorithm for localization in sensor networks. This depends on the environment and the specifications of the used motes [10]

Others showed that all the variations (transmitter frequency, acoustic hardware, etc...) lead to errors up to 300 % in the distance estimates and thus insisted in calibration issues instead of developing specialized hardware.

4.5.1 Some known techniques used at localization

a. Odometry

b. GPS (Global Positioning System)

c. Landmarks

d. Trilateration

e. Distance Sensors

f. Fixed Camera System

In the Odometry based map, the Edges represent the actions taken by the Roomba. In the topological map, the Edges represent the transition between two points of interest. In Odometry approach each node will be aware of its only one neighbor surrounding which is the closest. That will cause missing neighbors and black loops turning to its other neighbor.

4.5.2 Some methods using at localization in WSN

Receiver Signal Strength (RSS)

Time of arrival (TOA)

Time difference of arrival (TDOA)

Angle of Arrival (AOA)

4.5.3 Ideal localization scheme

Fine-Grained,

Ad-hoc deployment [22],

Distributed or Decentralized

5. LOCALIZATION AND NAVIGATION IN WIRELESS NETWORKS

If we are deploying mobile robots in wireless sensor networks and want to localize the nodes and navigate the mobile nodes (robot), we have 3 main subjects:

a- Determining the locations of all nodes (both mobile robot and sensors) by measuring the distance between nodes.

b- Directing the mobile nodes on the path/route throughout the network to the target node that mobile robot would like to reach.

At this chapter, you can find localization approaches, obstacle avoidance methods to be used if there are obstacles at the area and navigation approaches.

5.1 Localization by Range Approaches in WSNs

Today the algorithms can be classified in either *range* or *range-free* methods. The range methods are being the first ones to appear. Their principle in localization is mainly to estimate the distances between node pairs, and then to compute the position of individual nodes in the global network. Triangulation is for example the most basic approach for computing the position. Before giving the details of different range methods, we will discuss how one can estimate inter-node distances.

5.1.1 RSSI (Received signal strength indicator) approach

Changes in signal due to propagation indoors are difficult to predict because of dense environment and propagation effects such as reflection, diffraction and scattering [21]. The multi-path fading effect, which is the result of either constructive or destructive combination of multiple signal copies at the receiver, causes the received signal to fluctuate around a mean value at a particular location. The received signal is usually modeled by the combined effects of large-scale fading and small-scale fading. The large-scale fading component describes the signal attenuation as the signal travels over a distance and is absorbed by material such as walls and floors along the way to the receiver. This component predicts the mean of the RSS and usually has a lognormal distribution. On the other hand, the small scale fading component explains the dramatic fluctuation of the signal due to multi-path fading. On the other hand, the small scale-fading component explains the dramatic fluctuation of the signal due to multi-path fading. If there is no LOS component, the small-scale-fading is often modeled with a Rayleigh distribution. Such scenarios are commonly called NLOS. If there is a LOS component; the small-scale-fading is typically modeled by a Rician distribution. Note that when the RSS is measured, the measurement averages out the small-scale-fading effects.

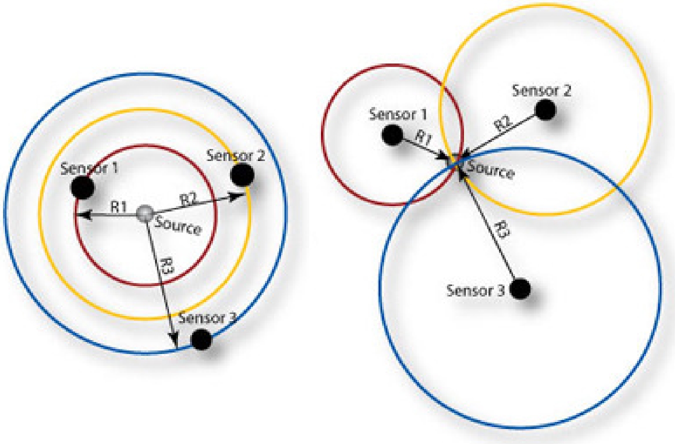


Figure 5.1 : RSS values are inversely proportional to distance.

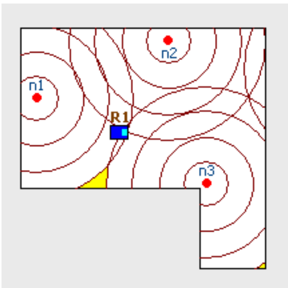


Figure 5.2 : The robot in a position that can be heard 3 or more nodes.

Each RSS value sent from a different sensor carries a specific number to indicate the "who I am" and "how I am close to the destination". This info is considered by sensor and robot to decide the route to follow. If the mobile vehicle or robot is in more than 1 sensor node sensing area, it runs a procedure to choose one to get closer.

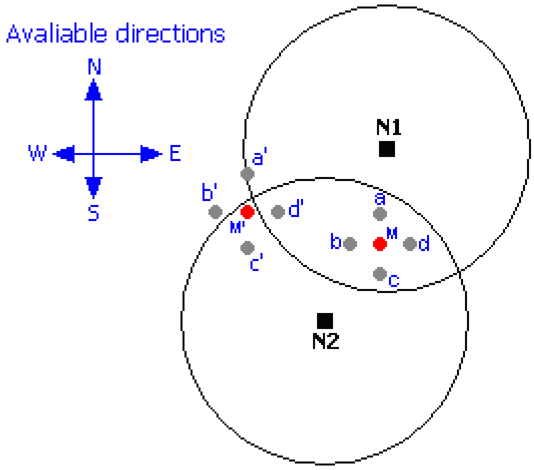


Figure 5.3 : The mobile robot in nodes' effecting area

In Figure 5.3, the mobile robot which shown in red is in a position represented by M and M'. In M position robot is able to step only a,b,c or d position only. This is considered as 1 cycle. Once robot moves to the other possible position, it will observe the difference between previous RSSI and new RSSI values and decide if it's getting closer to the node or not. Table 5.1 shows what the RSSI values in each step are at M and M' [16].

Table 5.1 : RSSI values at 2 different positions of robot that also shown in Figure 5.3

N1 Report	N2 Report	N2 Report
$\Delta_{RSSI}\{M\}$: 0.20	$\Delta_{RSSI}\{M\}$: 0.21	$\Delta_{RSSI}\{M'\}$: 0.0
$\Delta_{RSSI}\{M,a\}$: 0.09	$\Delta_{RSSI}\{M,a\}$: 0.50	$\Delta_{RSSI}\{M',a\}$: 0.01
$\Delta_{RSSI}\{M,b\}$: 0.25	$\Delta_{RSSI}\{M,b\}$: 0.18	$\Delta_{RSSI}\{M',b\}$: 0.0
$\Delta_{RSSI}\{M,c\}$: 0.30	$\Delta_{RSSI}\{M,c\}$: 0.08	$\Delta_{RSSI}\{M',c\}$: 0.0
$\Delta_{RSSI}\{M,d\}$: 0.11	$\Delta_{RSSI}\{M,d\}$: 0.22	$\Delta_{RSSI}\{M',d\}$: 0.13

The table above shows the RSSI values at a, b, c, d from M and a', b', c' and d' from M' at the Figure 5.3. The received signal measurement capability of the ZigBee hardware module specifies the quality of distance and positioning determination capability of system.

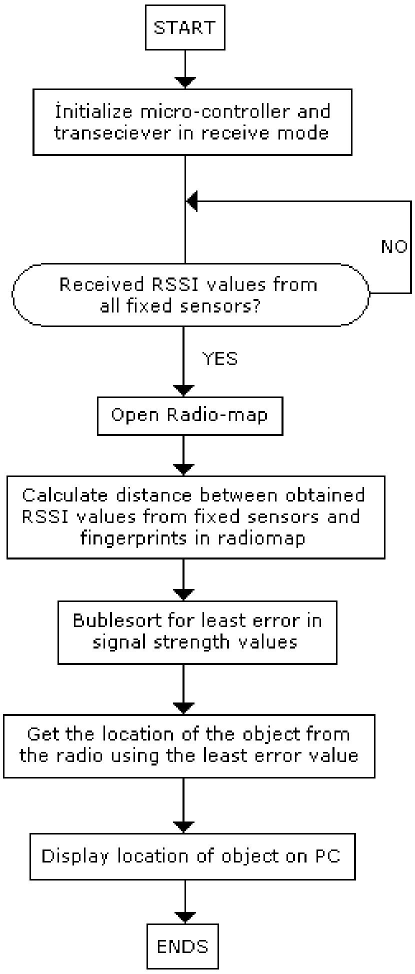


Figure 5.4 : The flowchart of localization in WSN algorithm.

Figure 5.4 is the flowchart that shows the process from mobile robot gets the signal from surrounding to it is shown in localization screen (for an observation application that's for user). During this process calculation of localization measurements can be done by a central computer or each sensor on its own.

Several localization systems have used received radio signal strength to estimate distance between transmitter and receiver. Perhaps the most well known of these is RADAR which uses existing 802.11 networks. Other commercial systems include PinPoint and WhereNet which deploy specialized RF infrastructure. Recent researches in ad-hoc localization using signal strength include SpotON and AHLoS. The current default radio protocol used with TinyOS measures signal strength with each radio message sent.

The energy of the radio signal, viewed as an electromagnetic wave, decreases as it propagates in space. By knowing the original emitted power and comparing it to the received signal power, one can estimate the attenuation g and deduce the distance via, for example, a free space path-loss model:

$$g = d^{-\alpha} \tag{5.1}$$

In this scheme the exponent α is around 2 in an open-space environment, but its value increases if the environment is more complex (walls, etc.) or less suitable for radio waves (metallic devices...). Another issue is that there is no unique path from the transmitter to the receiver. Any reverberations of the signal will influence the received strength, so it has to be measured at the appropriate moment. Some consider the first-peak, whereas others prefer an average of the first periods. RSSI can be calculated by:

$$r = \begin{cases} \sqrt{\dfrac{r_{Max}{}^{2}\left(\left|E_{Max}\right| - \left|E_{Measured}\right|\right)}{\left|E_{Max}\right|}} & 0 < \left|E_{Measured}\right| < \left|E_{Max}\right| \\ 0 & sonst \end{cases} \tag{5.2}$$

And the theoretical signal strength progression:

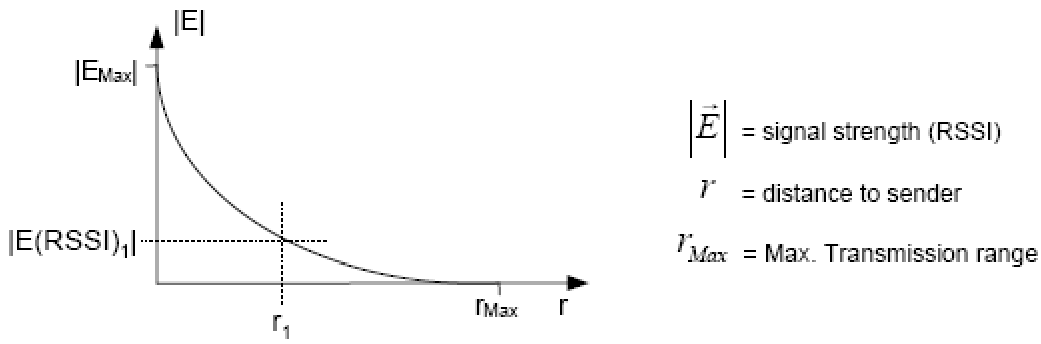


Figure 5.5 : The relation between signal strength and radius

5.1.2 Position estimation calculation at the sink node

We use a maximum-likelihood (ML) estimation to estimate the position of a target by minimizing the differences between the measured and estimated distances. ML estimation of a target's position can be obtained using the minimum mean square error (MMSE) [18], which can resolve the position from data that includes errors. We explain the calculation for a two-dimensional case as follows. MMSE needs three or more sensor nodes to resolve a target's position. First, the sink node searches for the same data in terms of a target ID and a packet number by collecting data from sensor nodes. The difference between measured and estimated distances is defined by:

$$f_i(x_0, y_0) = d_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \quad \textbf{(5.3)}$$

where (x_0,y_0) is the unknown position of the target node, (x_i,y_i) for i= 1,2,3…N is the sensor node position, and $N \geq 3$ is the total number of data that the sink has collected, and d_i is the distance between sensor node i and the target. The target's position (x_0,y_0) can be obtained by MMSE. By setting $f_i = 0$, Eq. (5.3) is transformed into

$$-x_i^2 - y_i^2 + d_i^2 = (x_0^2 + y_0^2) + x_0(-2x_i) + y_0(-2y_i) \quad \textbf{(5.4)}$$

After getting Eq. (5.4), we can eliminate the $x_0^2 + y_0^2$ terms by subtracting kth equation from the rest, as follows.

$$-x_i^2 - y_i^2 + d_i^2 - (-x_k^2 - y_k^2 + d_k^2) = 2x_0(x_k - x_i) + 2y_0(y_k - y_i) \quad \textbf{(5.5)}$$

Then Eq. (5.5) is transformed into Eq. (5.5), which can be solved using the matrix solution given by Eq. (5.6). Position (x_0,y_0) can be obtained by calculating Eq.(5.6).

$$y = Xb \quad \textbf{(5.6)}$$

$$b = (X^T X)^{-1} X^T y \quad \textbf{(5.7)}$$

where

$$X = \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) \\ \vdots & \vdots \\ 2(x_k - x_{k-1}) & 2(y_k - x_{y-1}) \end{bmatrix} \quad \textbf{(5.8)}$$

$$y = \begin{bmatrix} -x_1^2 - y_1^2 + d_1^2 - (-x_k^2 - y_k^2 + d_k^2) \\ \vdots \\ -x_{k-1}^2 - y_{k-1}^2 + d_{k-1}^2 - (-x_k^2 - y_k^2 + d_k^2) \end{bmatrix} \quad \textbf{(5.9)}$$

$$b = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad \textbf{(5.10)}$$

5.1.3 ToF (Time of flight)

When the environment is supposed coherent enough for the propagation of a signal being at constant speed, knowing the speed and measuring the time of propagation will give an estimation of the distance. This is the basic principle of the Time of Flight (ToF), which could also be applied to a radio signal. Since the propagation speed of radio signals is very high (indeed, equal to the speed of light), time measurements must be very accurate in order to avoid large uncertainties. For example, a localization accuracy of 1m requires timing accuracy on the level of 3.3 nanoseconds.

In the case of the Global Positioning System, a synchronization of the atomic clocks in the satellites gives a great accuracy (thus depending on the clock of the receiver), but in the case of wireless sensor networks, the achieved accuracy is very poor: the Telos motes used at the Automatic Control Group have a time stamp of the radio packet with an accuracy of 1 millisecond. Using an acoustic signal will decrease the propagation speed, and thus increase the accuracy. With a precision of 1 ms, the localization accuracy is 35 centimeters. Unfortunately, the motes we are using were not primarily designed for localization purposes, and have neither acoustic transceivers nor receivers. The NADA department at KTH is conducting researches with embedded sensor boards, and their motes have both beepers and microphones. They could be used with acoustic time of flight algorithms, but we have been unable to conduct measurements on the software platform. One restriction could be on how much they are sensitive to their own signals.

An advantage of acoustic time of flight is the multipath avoidance, as the signal suffers less interaction with its reflections. Both time of flight and acoustic time of flight are more expensive than received signal strength (hardware, power), but are more accurate and almost computation free.

5.1.4 Using both methods together: Calamari approach

Calamari is a good compromise and a solution to the calibration problem. The authors showed that normal variations (in, for example, transmit frequency, acoustic hardware, etc) between sensor nodes from the same manufacturer may lead to an error up to 300% in the distance estimates. Although these errors could potentially be remedied via higher tolerances on hardware components, calibration would certainly be a much more cost efficient approach. A traditional calibration technique would be to map the device response to the desired one [16]. But this procedure has to be performed for every *pair* of devices, thus it is order n^2. This *pairwise calibration* is too expensive.

5.1.5 Calibration

The mass-produced, analog components used in Calamari provide a cheap, low-power solution but also introduce high variability between nodes, which often has as much effect on distance estimates as distance itself. Without calibration, ranging estimates in Calamari are nearly useless.

For example, a radio may transmit at up to twice the power of another radio, leading to distance errors of up to 100%. Variations in transmitter frequency also affect the observed RSSI, the RSSI values of one receiver as the transmission frequency was varied over the observed range of transmitter frequencies.

The Figure 5.6 below shows uncalibrated TOF readings are always greater than the true distance and are highly erroneous due to transmit- and receive-delays.

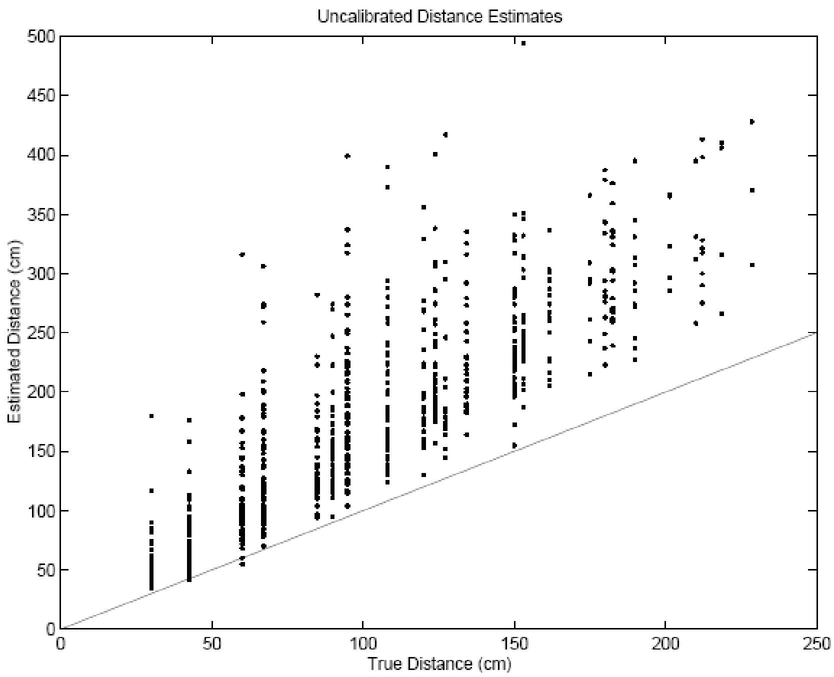


Figure 5.6 : Uncalibrated TOF readings

Here is several solutions are explained below:

Iterative calibration: One transmitter is said to be the reference, and all the receivers are calibrated, and the other way round. The problem is that it is valid for a single frequency. As the frequencies may vary from a transmitter to another, this is still a valid scheme for a single pair, in a way.

Mean calibration: avoids the pair problem by a simple assumption: the variations in the devices are normally distributed. Each receiver is then calibrated using all transmitters, but transmitters are not calibrated...

Joint calibration: is used in Calamari, to calibrate each device by optimizing the *overall system response*. The ToF estimation is affected by hardware issues, mostly Bias (time for starting oscillating) and Gain (volume of the emitter, sensitivity of the receiver).

The sensor model is then:

$$\mathbf{r_* = B_T + B_R + G_T \cdot r + G_R \cdot r} \quad \mathbf{(5.11)}$$

Frequency and sensor orientation also affect the output, but are considered as included in the error term. This relation for each sensor pair will give 4n variables and

$$\boldsymbol{n^2 - n} \textbf{ equations} \quad \mathbf{(5.12)}$$

There is no way to solve each of them separately (i.e. to decide if the error is due to the transmitter or the receiver), but it can be solved globally.

All the calibration process was with known distances. A proposition of **Auto calibration** for a completely uncontrolled environment is to take advantage of symmetric pairs ($\boldsymbol{d^*_{ij} = d^*_{ji}}$) and triangle inequality

$$\boldsymbol{(d^*_{ij} + d^*_{jk} - d^*_{ik} \geq 0,} \textbf{ if } \boldsymbol{i, j} \textbf{ and } \boldsymbol{k} \textbf{ are connected)} \quad \mathbf{(5.13)}$$

If some anchor nodes are used, the known distances can replace the estimates in the above equations, thus reducing the estimation error. Just a few information about the mixed use in Calamari:

TinyOS [Url-6] current default radio protocol measures signal strength with every sent message. And the message is time stamped with micro second accuracy. Both radio and acoustic messages are sent simultaneously. When the acoustic impulse is received, the processor is toped with a time stamp with micro-second accuracy. The difference between the two stamps, multiplied by the speed of sound gives the distance. Technically, it seems that RSSI is not used, but only the time stamp included in it.

5.1.6 ToA (Time of arrival)

Time of Arrival (ToA), also named Time of Flight (ToF), which both means the travel time of a radio signal from a single transmitter to a remote single receiver. By the relation between light speed in vacuum and the carrier frequency of a signal the time is a measure for the distance between transmitter and receiver. However, in some publications the fact is ignored, that this relation is well defined for vacuum, but is different for all other material when radio waves pass through.

Similar to the TDOA technique, this TOA or Time of arrival called technology only differs in the fact that it uses the absolute time of arrival at a certain base station rather than the measured time difference between departing from one and arriving at the other station. The distance can be directly calculated from the time of arrival as signals travel with a known velocity. Time of arrival data from two base stations will narrow a position to two circles and data from a third base station is required to resolve the precise position with the third circle when matching in a single point [17]. There are many ToA-based localization systems, including GPS.

As with TDOA, synchronization of the network base stations is important. This synchronization can be done in different ways:

- With exact synchronous clock on both sides. Inaccuracy in the clock synchronization translates directly to an imprecise location.
- With two signals which have different spreading speed. Distance to a lightning strike can be measured in this way (speed of light and sound velocity).
- Via measurement to a third reference point.

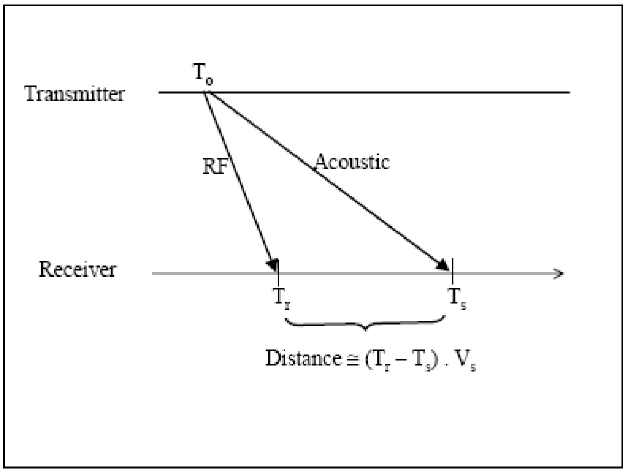


Figure 5.7 : The illustration of ToA

5.2 Range-Free Methods

Contrary to the first ones, those methods never compute the distances to the neighbors. They use hearing and connectivity information to identify the nodes and beacons in their radio range, and then estimate their position.

5.2.1 Can any node hear me?

This idea of only using the information of the immediate neighbors fits perfectly the distributed approach of the localization problem. In those type of schemes, every node only uses direct communications to refine his position estimates, and when it succeeds to achieve a given accuracy, it broadcasts the result. The big advantage is that it saves a lot of traffic, but an overload of the radio channels can occur. This has to be carefully studied, and the rules for priority clearly established. Another drawback is the fact that those techniques usually require a great amount of nodes.

5.2.2 APIT approach

The APIT idea is to divide the environment into triangles, given by beaconing nodes. An individual node's presence or absence in each of those triangles will allow reducing the possible location area. This goes until all the possible sets are exhausted, or the desired accuracy reached. The APIT algorithm is then ran at every node:

1. Receive locations from n anchors.

2. For each possible triangle, test if inside or not.

3. If yes, add it to the InsideSet.

4. Break if accuracy reached.

5. Estimate position as *CenterOfGravity* (∩Ti {Element Of} InsideSet).

For testing if the node is inside or not a triangle according to the Point-in-Triangle (PIT) test, it needs to move. To cope with situations where nodes are static and unable to move, an *Approximate* PIT test is defined according to:

If no neighbor of M is further from/closer to all three anchors A, B and C simultaneously, M assumes that it is inside triangle ABC. Otherwise, it is outside.

This is of course subject to errors, especially if the node is close to one of the network's edges, or if the neighbors have an irregular placement. The authors of [9] have performed extensive simulations and claim that the error has never exceeded 15 % (on their particular scenarios).

To minimize such errors, there is an aggregation of the algorithm's results, not only an intersection. A grid represents the possible location for the mote. Initially filled with zeros, it is incremented for every triangle that had a positive APIT test, and decremented for others. The area with maximum overlap has then the highest numbers, and its center of gravity will be the estimated position.

An important aspect of this solution is that APIT uses indeed signal strength, but not as an approximate for a distance. It just assumes that signal strength decreases monotonically with the distance (usually valid). Thus it is used to *compare* distance, and APIT is still a range-free algorithm.

5.2.3 Multi-Hop

Multi-hop methods are mainly range-free, but can also use estimation of the distances. Their purpose is to compute a connectivity graph, and then trying to make it fit the known positions as good as possible.

5.2.4 Multi dimensional scaling

In a large sensor network, Multi Dimensional Scaling (MDS) only uses connectivity information, *i.e.* which nodes are within communication range of which others. The process has three steps:

1. Rough estimation of the distance between each possible pair of nodes.
2. MDS to derive locations fitting the estimated distances.
3. Optimization by taking the known positions into account.

The system is modeled by a connectivity graph, the edges having the value 1 (if the distances are known, the values are used instead). This gives a symmetric matrix, which is run in a classical all-pairs shortest-path algorithm. The resulting distance matrix is used in classical MDS, and gives a relative map locating each node.

Linear transformations (scaling, rotations, reflections, translations) are used to fit the anchors' estimated positions to the correct ones, and perhaps all other known positions, if any. There are many types of MDS techniques: metric/nonmetric, classical/ replicated, weighted, deterministic/probabilistic. Classical MDS, where the proximities are used as being distances, seems to be the best choice in this issue. The Euclidian distance has then to be as close as possible to the proximities (least squares).

5.2.5 N-Hop multilateration

Multihop multilateration's technique is aiming to give to give nodes that are several hops away from beacons the possibility to collaborate in finding better position estimates. By allowing this type of collaboration, the ratio of beacons to nodes can be decreased. The algorithm could be centralized or decentralized, see [16] for a detailed account of the distributed version, fitting best sensor networks (communication costs distributed, accepts node failures).

5.3 Measuring Distance

Sensor nodes receive packets from targets, measure the power of the packet, and transform the RSSI into distance for use in theoretical or empirical models. The packet includes a target ID and a packet number. By reading the packet, a sensor gets the target ID, packet number, and the distance between the sensor and the target. It then sends the following data to the sink: sensor ID, target ID, packet number, and sensor-to-target distance [24]. Figure 5.8 below shows the RSSI values received from different nodes (which each nodes sends own IDs in binary format as 00001, 00010, 00011 etc.) Mobile robot considers the greatest value as RSSI

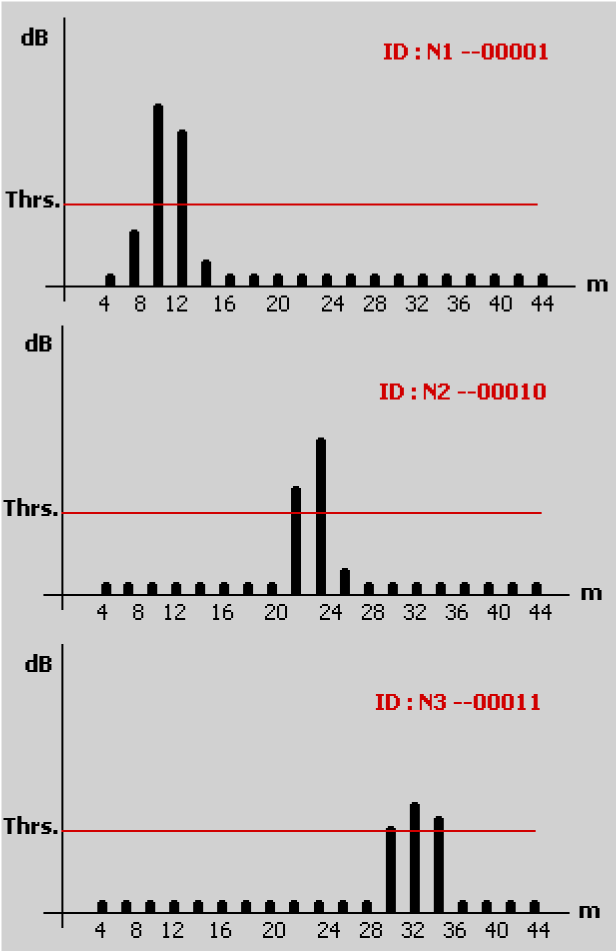


Figure 5.8 : Different signal strength values received from around

5.4 Determining the Direction and Angle

In this condition a minimal cycle step helps the robot to realize if it is getting closer to the node or not. Robot can compare the received signal strengths (RSS) of two positions/locations. If RSS increases when robot moves from A to B position, robot understands that it's getting closer to the desired node. If not, robots change the direction by 45 degrees and keep moving and calculating if RSS is increasing.

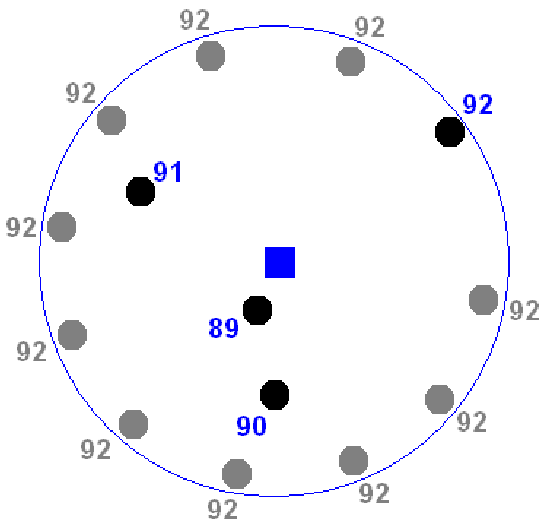


Figure 5.9 : Robot hears a signal from a node but doesn't know in which direction

The figure 5.9 illustrates a very important question for localization/navigation a mobile vehicle/robot in a wireless sensor network study. It was easier to hear a signal from surroundings and calculate which sensors are around and which has the greatest number. Let's assume, robot can hear "92" as the greatest number and wants to step to that node. The goal question is: "The node is 7 meters away but in which direction? "

In fact there are several methods to fix this issue with specific accuracy methods. With the help of two or more nodes, that's easier to find the exact position of mobile robot. Because of the other nodes have already RSSI values that are indicating the distance between mobile robot and itself. By determining both values together, it will provide us the elimination of probabilities of the location of the mobile robot. Namely we will be able to eliminate the probable positions of mobile robot that is not current. Because, robot can be at only one position.

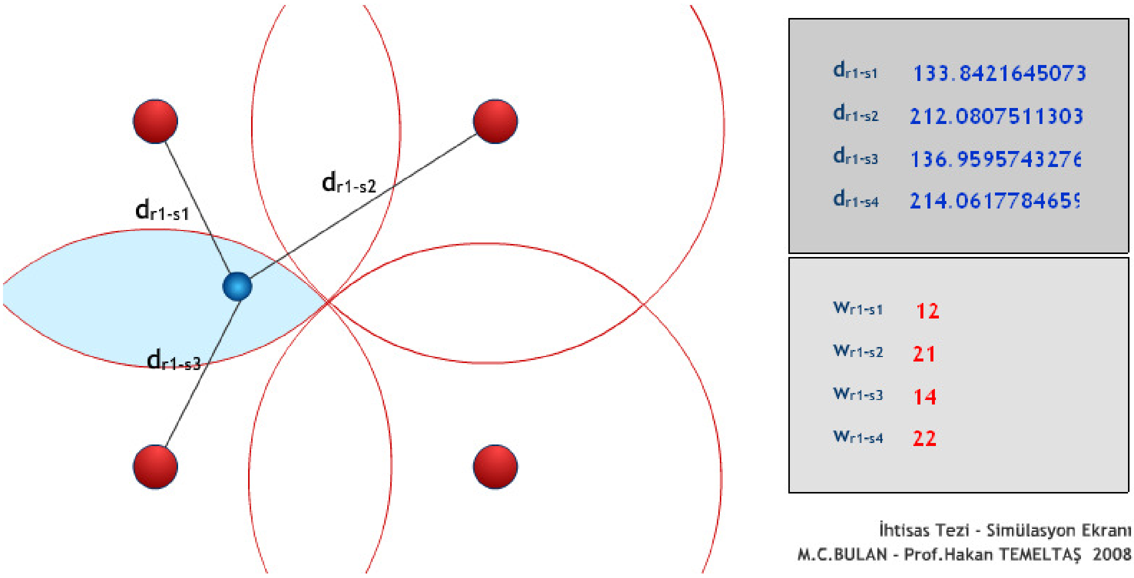


Figure 5.10 : The distance between mobile robot and three of nodes

Figure 5.10 illustrates how 3 of nodes can be enough to eliminate the possible positions of the robot. If mobile robot hears 3 signals from 3 different nodes, it will measure the RSSI value for each of them. Also nodes will measure the RSSI between their selves and get the neighbor node distances. Finally mobile robot will be able to localize itself by multilateration algorithm.

5.5 Obstacle Avoidance

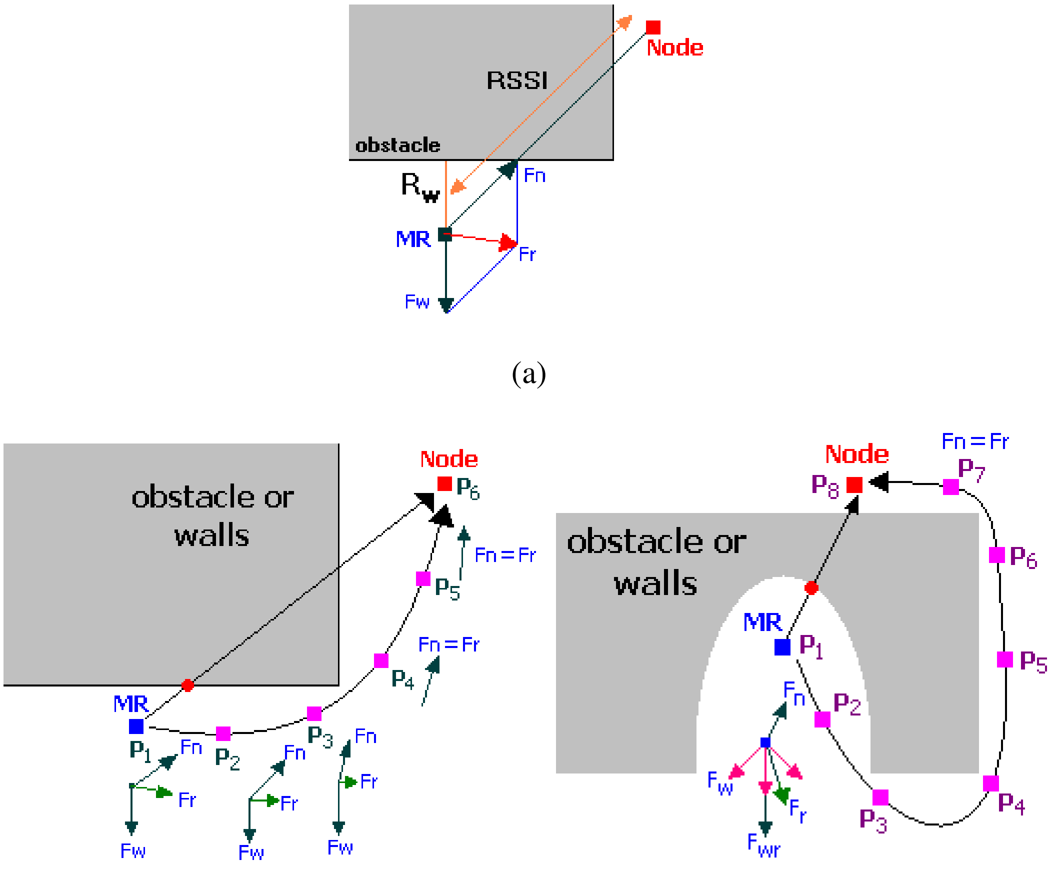
At all costs robots should avoid colliding with the obstacles in their environments. In dynamic environments, that is, where there are multiple moving obstacles, obstacle avoidance is a difficult problem. Although the dynamics of moving obstacles can sometimes be predicted, like when trains run on a track, sometimes the dynamics may be more uncertain, like playing children running around. A robot uses path planning techniques to plan a collision free path from one location to another location. If the obstacles in the environment are dynamic, the path planning problem is NP-hard.

At the other hand, artificial potential fields method is used to provide obstacle avoidance. The potentials of the obstacles are added to the potential of nodes and generates resultant force vector including force value and direction angle.

5.6 Heading Towards Sensor Nodes

Once mobile vehicle decides to move towards to a node, it calculates the direction easily by localization methods and moves by resultant vector. But there is an obstacle on mobile robot's way; it should re-calculate the new route to follow.

Figure 5.10 illustrates two samples (b and c) conditions and resultant vector forces that determine mobile vehicle's route at every step:



(a)

(b) (c)

Figure 5.11 : Three conditions that there is an obstacle on MR's way

In this sample, mobile robot would head towards F_n direction and move, but crashes the wall if keep moving. Also we assumed that the signals are directly received by the mobile robot over the obstacles.

The distance sensors which are located on the mobile robot provide obstacle information around the vehicle by the distance between obstacle and mobile robot. This information is used to generate F_o (o: obstacle) and this force is added to the F_n (n: node) force vector to get resultant vector. The obstacle force is only considered when the obstacle is between heading node and mobile robot. Otherwise, system doesn't put obstacle force vector into account while evaluating the directional vector. The obstacle force vector is normal to the obstacle surface. Figure 5.11 a, b and c illustrates the condition where there is an obstacle on robot's path and how to calculate the resultant force to creates the new path for mobile robot.

5.7 Path Quality Value (PQV)

At the same time, once the network was settled by hop-count numbering there may be some positions of mobile vehicle that it received more than 1 same signal. This situation causes a collision or overriding the information on the network. For example mobile vehicle may receive "95" signal from two different nodes.

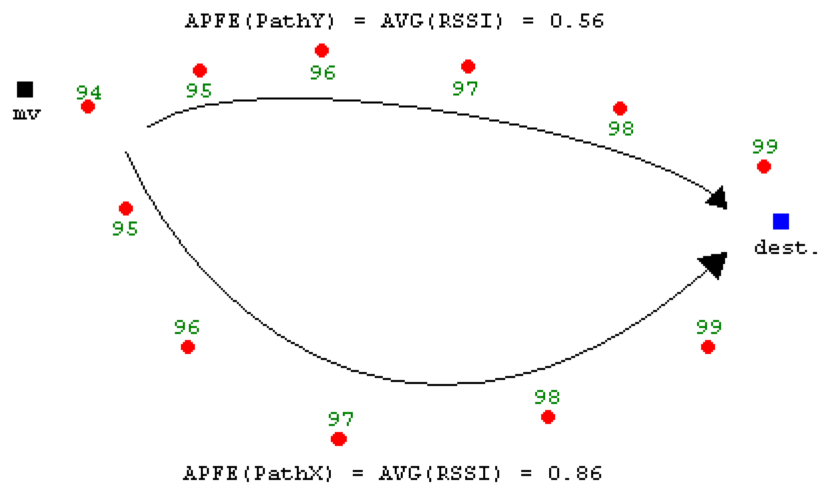


Figure 5.12: Two paths that have same HC numbering order, but have different lengths

Figure 5.12 above illustrates this condition. There are 6 nodes at both paths and the end nodes broadcast the same signal even received by the last node of the paths. When mobile vehicle receives the "95" signals from both nodes (even may be more), it takes the PQV value into account to decide which one is better way to follow.

PQV is calculated as an average indicator of the RSSI values of constituent nodes. This value totally depends on the distance between the nodes of the path. Figure 5.12 shows that how this value was determined below:

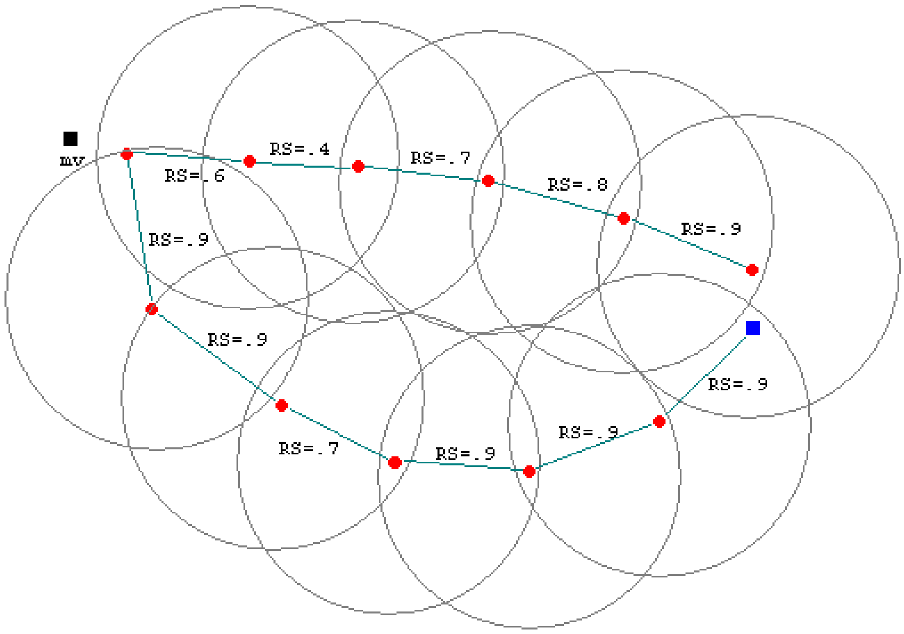


Figure 5.13 : Determining the PQV of a path

The PQV value provides information to mobile vehicle to choose the shorter path to follow and preventing time and energy loss. Especially time efficiency is very important for indoor projects. The greater PQV indicates the higher quality of path.

To calculate a PQV for a specific path, we use the following equation (5.14) by considering the nodes only at the path:

$$\mathrm{PQV(P)} = \begin{cases} \sum_{i=f}^{l} RSSI(Ni; Ni-1) \;/\; (1-\mathrm{f}) & \text{NodeCount} > 2 \\ \mathrm{RSSI\ (Ni, Ni\text{-}1)} & \text{NodeCount} \leq 2 \end{cases} \qquad \textbf{(5.14)}$$

where P is the specific path, RSSI(N_i, N_{i-1}) is the RSSI value between nodes i and i-1, l stands for last and f stands for first node.

PQV value can be determined by collecting the RSSI values from each node. If any node stops collaborating with the neighbor nodes to share / exchange data at a node-chain, there would not be a connection between the neighbors of the missing node. This is called “Broken Node-Chain” and each piece of main chain generates a new chain. There must be at least 2 nodes to calculate a PQV value, because at least 2 nodes can form a chain after all.

If the PQV value may change at the next steps of mobile vehicle's travel because of nodes, the algorithm of the MV is adaptive and chooses the best PQV value (largest one) at each T_n time.

6. EXPERIMENTAL STUDY and SIMULATION SOFTWARE

6.1 Methodology and Purpose

In our project we have about 10 sensor nodes randomly located at the area and 1 sensor node which is located on mobile robot as dynamic node. We have our mobile robot to be navigated in the network by the help of sensor nodes and finally find the target node.

6.1.1 Setting up the networks by hop-count-numbering

Let's assume that we have a network with less than 10 randomly located sensors and 1 mobile vehicle/robot and 1 destination. For this thesis, we planned to consider this project in an indoor area. All nodes/sensors have the same specifications such as sensing area radius, signal processing unit. Our plan is starting the mobile robot's travel from a random position and let it to be navigated over the network by static sensors. Let's describe the idea by figures;

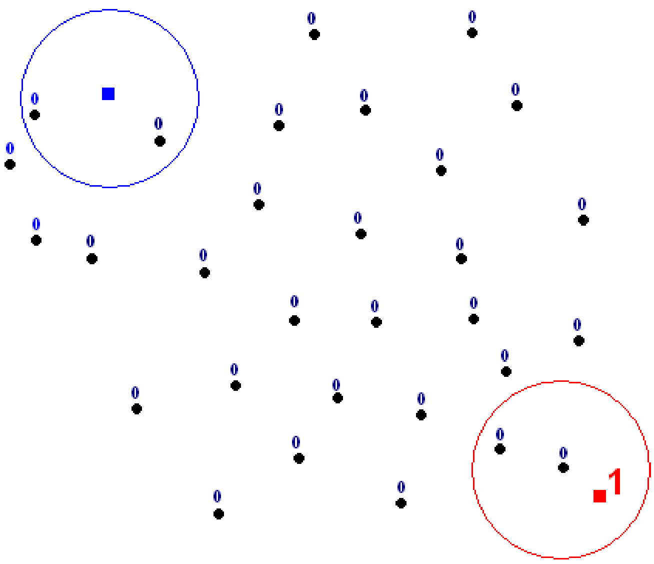


Figure 6.1: Initial position of the network

The initial situation of sensor network is shown in Figure 6.1. The starting position of the mobile robot is shown as blue square and position of the destination is shown as red square. The circles around the robot and sensors are radius of the affecting area. Initially, all sensors have "0" (zero) as propagation signal. After process starts, all sensors get the prop. signal number by neighborhood.

The destination propagates "1" as "***I AM THE DESTINATION***" signal. And the one which can hear the destination gets the signal number "99" as maximum number. Each sensor nodes which can hear the other one, get the 1 less signal number of its neighbour.The maximum starting number depends on the sensor count of the network. For example if there are 1000 sensors at the network, recommended maximum starting number is will be "1001".

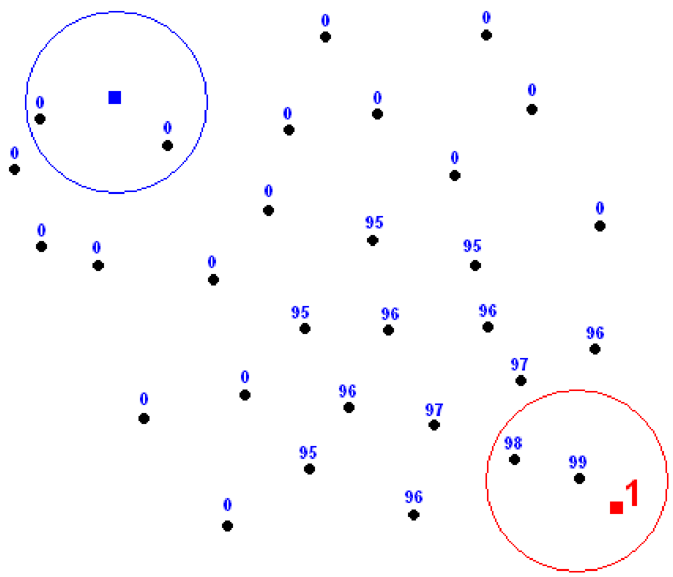


Figure 6.2 : View of the WSN at t instance while process continues

Figure 6.2 illustrates how the nodes getting numbers of propagation throughout the wireless sensor network. If a node's nearest neighbor has the propagation number N, this node will take N-1 as propagation number. This will propagate over the wireless sensor network and all nodes which can hear any node take a number and start to send.

For example, if the node can hear N, N-3 and N+2 values from the nodes in range, it will choose N+2 as reference and will get (N+2)-1 = N+1 value as hop count order.

The number “0” indicates that the node cannot hear any node which is in the chain to the destination node (target).

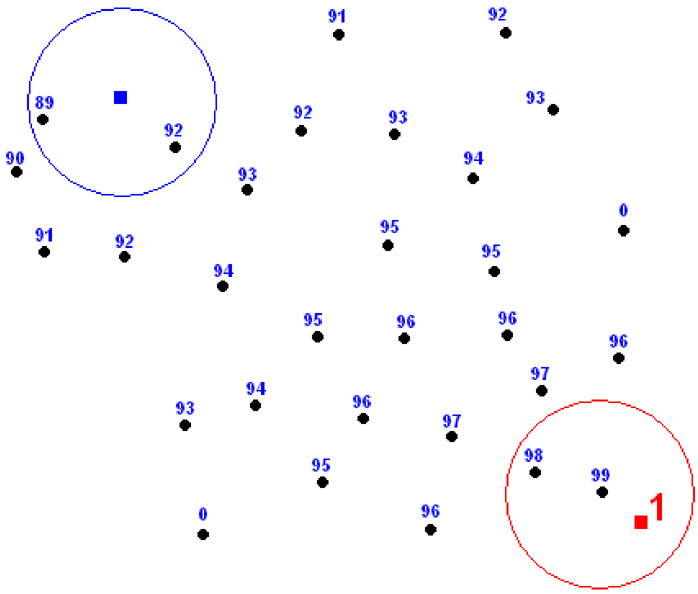


Figure 6.3 : After propagation process has finished

Figure 6.3 shows the status of the sensor network after propagation finished and all nodes got the number. The nodes which have “0” are the ones which don’t hear any node. The number counts down from the max one. The least one carries “89” for the network of this project. If a node can hear more than one neighbor sensors, this node gets the max number of the incoming signals. For example if a node can hear, 93, 92, 91 and 90 from neighbors this node chooses “92” as propagation signal number.

Also mobile vehicle/robot behaves the same. Meanly, it chooses the maximum number from the signals from its surroundings and tends to this node by tending algorithm defined before.

If mobile robot hears more than 1 node which have the same number, this time, the application in the robots considers the node which has a greater path quality value (PQV). PQV will be described at the next pages.

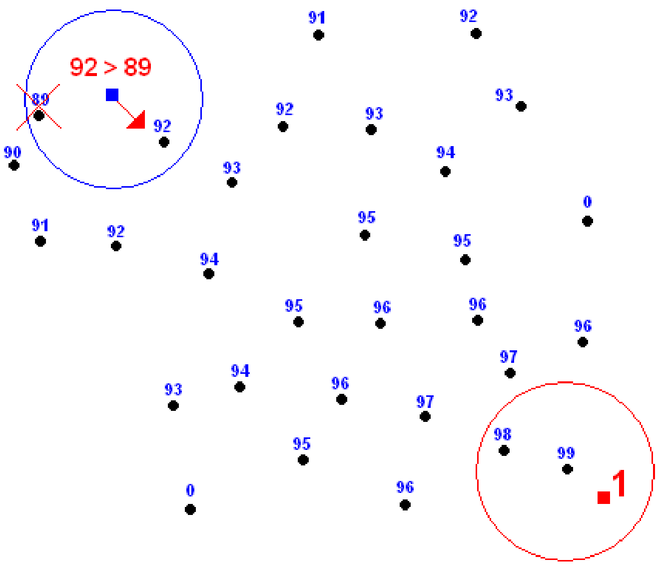


Figure 6.4 : Robot/Vehicle tending to the destination in mapped network

Figure 6.4 shows the main principle of the behavior of the mobile vehicle/robot in the wireless sensor network. Robot listen all signals coming from its surrounding and calculates the maximum number of signals. Robot tends to the node which has the maximum number while moving ahead.

If robot doesn't hear any signal from its surrounding, it will follow the default direction that was set before process was started. In this project, +315 or -45 degrees in X-Y plane was decided to be default direction.

If robot hears more than 1 signals, it chooses the node which sending the maximum number. But in this condition, however robot can measure the distance to the node, it cannot realize in which direction that node is located. This question is figured out in the Figure 6.4.

The nodes which cannot communicate with the other nodes in the network get *sleep mode* after a defined time is passed, that is called Time-Out duration. At sleep mode, the node only broadcasts "I AM HERE" signal around in a very low frequency (may be 1 time per hour or day).

6.2 Using PQV to Choose the Best Path

Path Quality Value (PQV) provides the best path to the mobile robot by calculation of inter-node RSSI relation. Mobile robot follows the route by listening PQV value from sensor nodes at every 10 second interval.

Mobile robot considers the PQV only if it hears the same sensor distance numbers from different sensor nodes at any position. Otherwise mobile robot keeps considering the largest sensor number.

6.3 Developed Application: SOLAN (Simulation of Localization and Navigation)

This project was developed by author with the help of supervisor and in following developing environment:

- Programming Language : **Java and Delphi**
- Test PC : **Pentium Cre2Duo 2.0 GHz and 2 GB RAM**
- Additional Requirements: An 3rd party tool which named **Active Comport** must be installed on the computer before running SOLAN. This additional middleware helps us to use USB ports as ethernet ports with the emulator code included.

General capabilities of the program:

- The background map is from Istanbul Technical University, Electronics Faculty 2nd floor plan
- There are 16 static sensor nodes, 1 destination, 2 mobile vehicles/robots
- The coordinates of destination can be set by user
- The moving speed of the robot can be configured by user, both step size and step frequency.
- Number mapping (hop-count) by neighborhood view
- The report of RSSI relations between neighbor nodes
- Mobile vehicle parameters at each instance (dynamically updated)
- Observing all values from the panel at the bottom-right

- Observing all actions of the mobile vehicle and communication with sensors
- Observing the instant distance between mobile vehicle and sensors
- Momentary observing the calculated position/location estimation
- Evaluating the sensor network behavior by sensor locations and user defined located obstacles.

The deficiencies of SOLAN;

- In this software you cannot add the noise effects depending on the application environment.
- The unknown obstacles are not considered in this simulation

Please find the software in attached CD to test. There are no pre-requisites to run the software except Active ComPort plug-in, and no setup is required as well. Please double click the application.exe file and run.

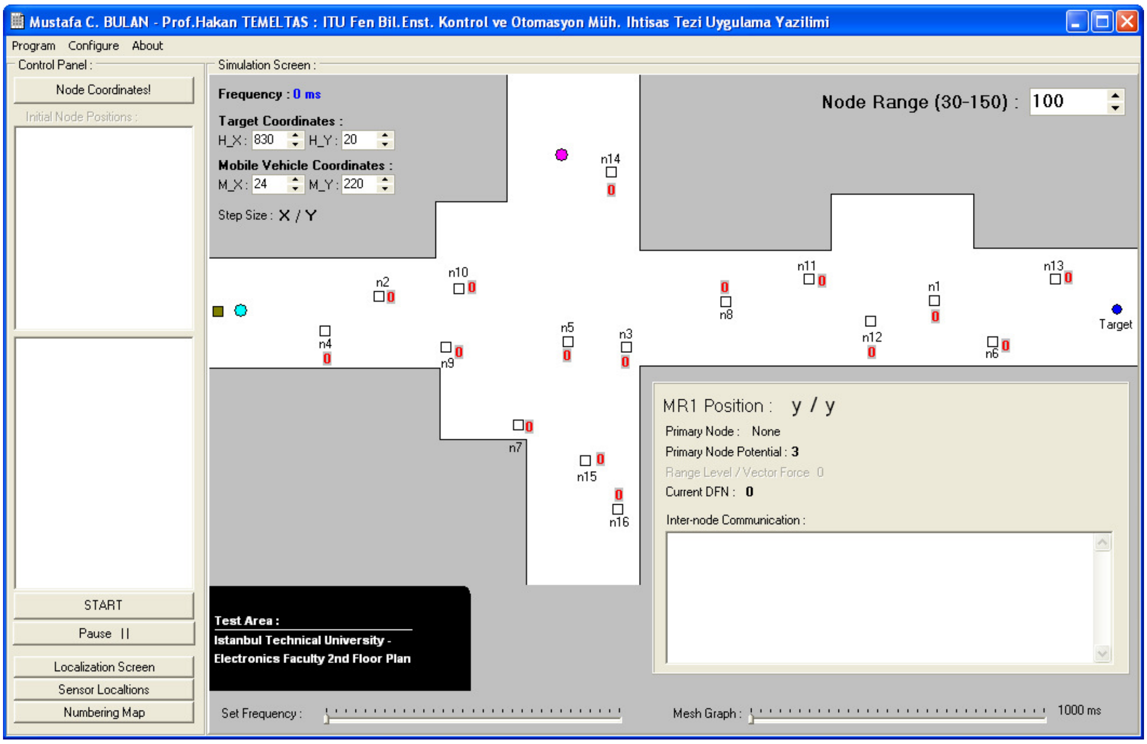


Figure 6.5 : Main window of SOLAN

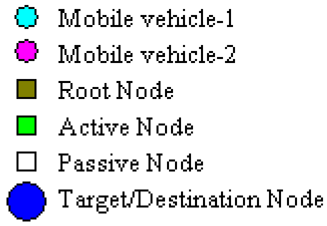


Figure 6.6 : The legend of the network

The main window of simulation software is shown at Figure 6.5 You can see nodes as n1, n2, n3... as white squares, "target" as blue circle, mobile robot as aqua circle, sink node as green square. Figure 6.6

You can set the initial position of target by the numerical spin edits at the top of the area. The background is the map of the Istanbul Technical University Electrical and Electronics Faculty, middle part of second floor plan.

We start simulation by clicking on the "Start" button. The icon colored Aqua is our mobile robot which will be navigated thought area. Once press the "Pause" button, all timer intervals are set to 0 and system freezes.

We can see the each RSSI values between the nodes in their range at the right bottom text box, see Figure 6.7.

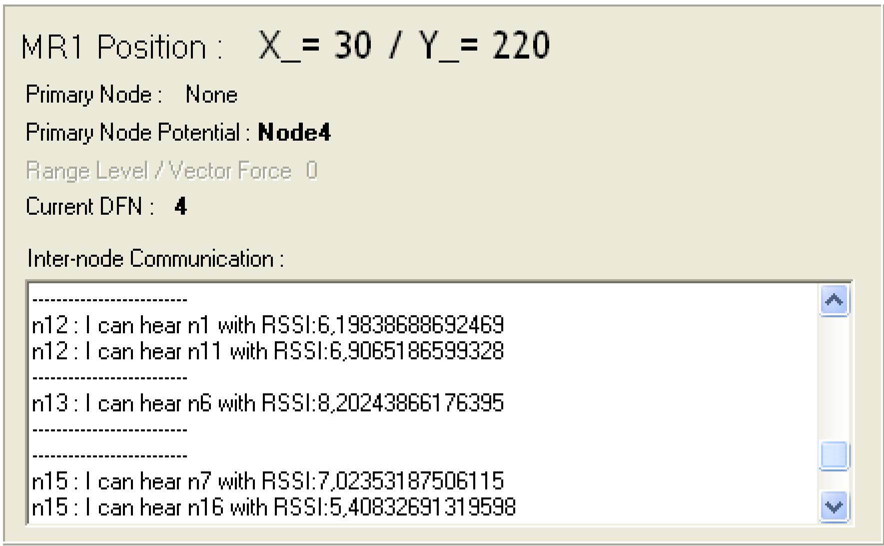


Figure 6.7 : RSSI report of WSN and instant communication data of mobile vehicle

All nodes are distributed randomly to the area and don't know their own locations. But when the simulation process starts the mapping and node/sensor localization algorithm will build the network number mapping, every node become aware of its neighbors. By this way, sensor nodes will realize how far it is away from the node which can hear the target. Please note that target has a specific numeric signal that represents “I AM THE TARGET”.

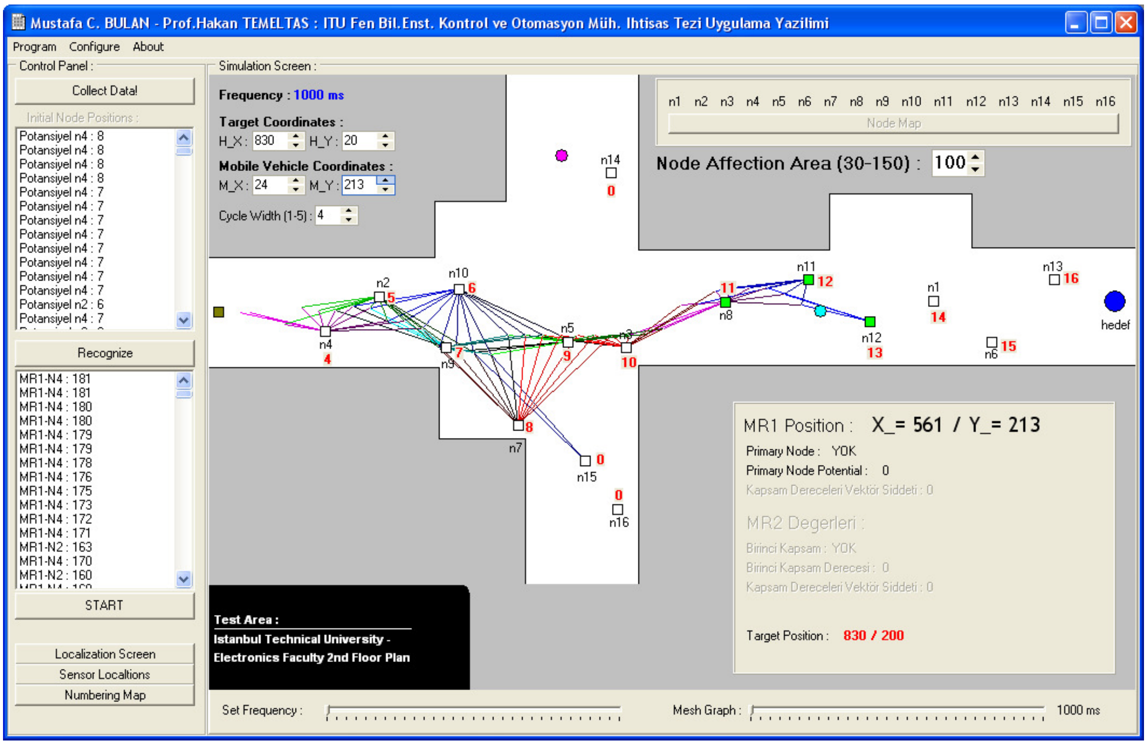


Figure 6.8 : SOLAN at runtime

You see the software after process started and mobile vehicle/robot is moving ahead. Robot has an initial direction angle set by user before process was started. This will prevent robot to keep in stuck positions if it doesn’t hear any signal from surrounding. In the screenshot got from application window has the robot with 0 degree default direction. Mobile robot listens the signals coming from its surrounding and calculates the signal strength by embedded devices. So mobile robot is able to measure the distance between the signaling node and itself.

Mobile robot also sends signals around as a unique signal code representing “I AM MOBILE VEHICLE/ROBOT” as number. The nodes which can hear this signal can share this info to the neighboring sensor nodes, thus the central information processor can calculate the route and tending algorithms.

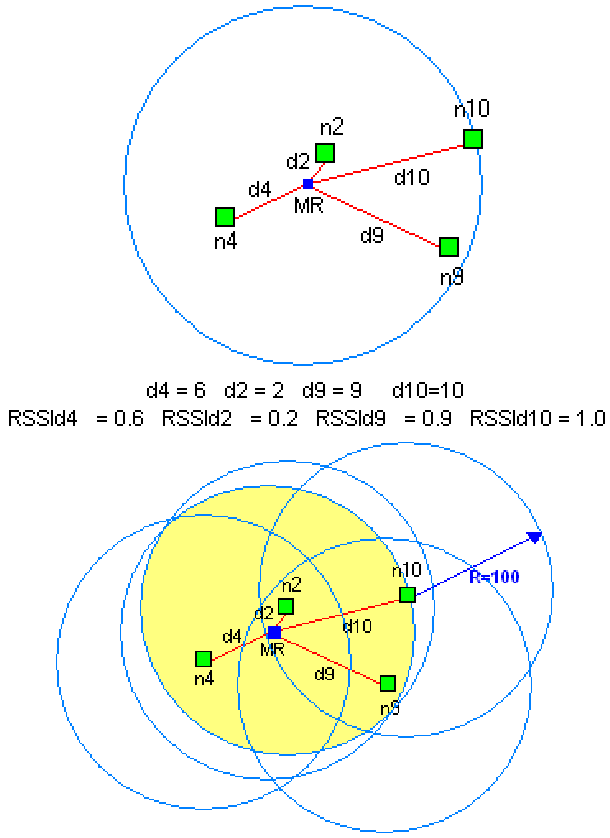


Figure 6.9 : Mobile vehicle is in the range of 4 nodes

The sensor nodes which hear the mobile robot, their color turns to “lime” and turns back to white when lost the signal from robot. At the same time, in this software, the line is drawn between the robot and nodes which can hear the robot in a user defined frequency.

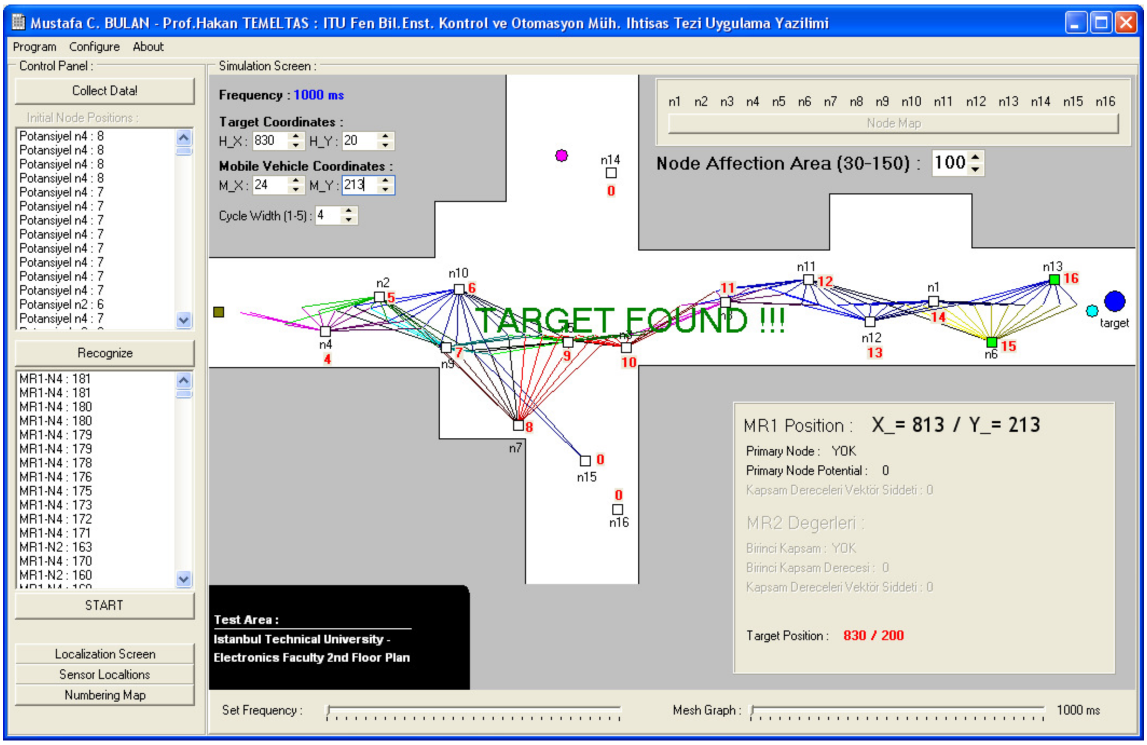


Figure 6.10 : SOLAN screen after process was completed

The screen which also graphically shows the final situation after robot found the destination node over the sensor network by help of sensor nodes. In the same screen, we can also see which nodes robot heard and tracked by sensors during its travel until last step.

The lines are kept to be able to see which sensor node get signal from robot during which part of travel.

Also all numerical calculations and values can be found in the listboxes aligned left side of window. We can scroll the listboxes and see the RSS values (RSSI) and affecting potential values robot got around in each t time.

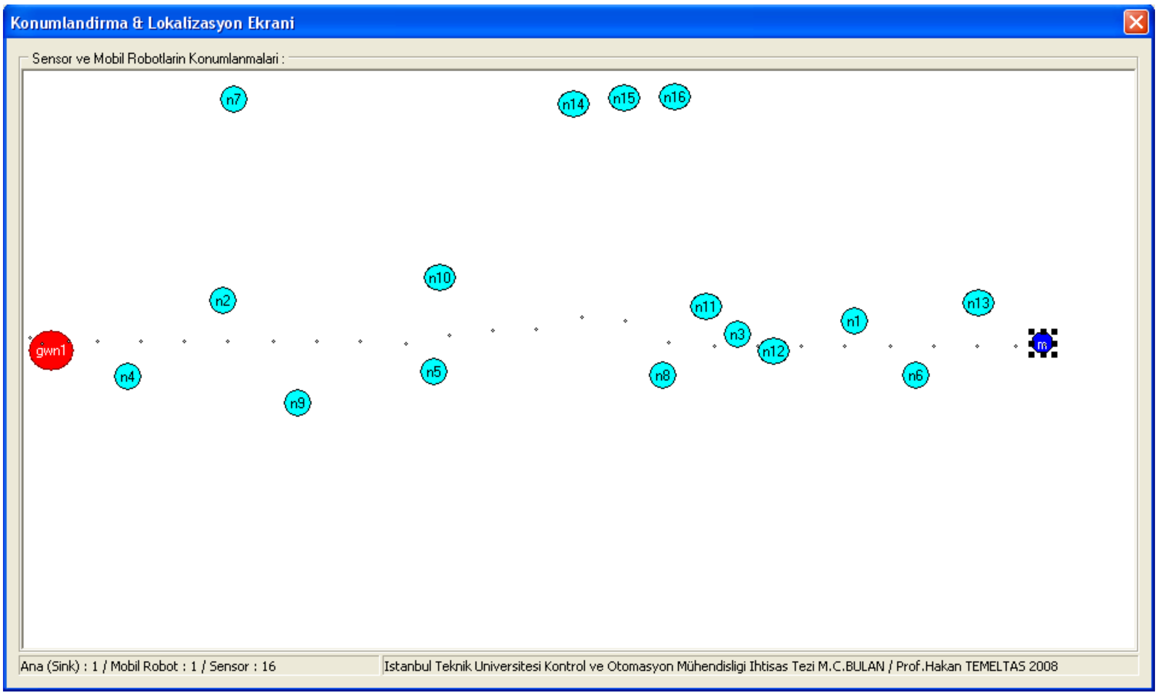


Figure 6.11 : Sensor node proximity window by mobile robot

The window above shows only the real time distances between mobile robot and nearest nodes. The red one is the main node and aqua ones are sensors randomly located in the area. The blue one is the mobile robot which puts a node at the points it passed. Program puts the related node shape to the position by the distance from the mobile robot and sets the position in every cycle.

In this window, we cannot see the angles and directions that the signals coming from. We can only see the distances. The nodes at the top of the window are the ones which did not recognized by mobile robot. So no distance could be calculated.

Also the robot, sensor and sink node counts are shown at the information bar, at the bottom of the window.

6.4 Communication SW and HW of Mobile Robots and Nodes

In this project, I developed a communication system based on ZigBee specification which includes data exchange between nodes, data transferring to the next nodes and mobile vehicles as well.

There are 3 main parts of the communication system. First part is for nodes which consist of ZigBee antenna and host modules. The second part is for communication peripheral that located on mobile vehicles and the third part is a software application that also including a GUI for a ground station which is a PC. In each communication part, different procedures are performed by specific algorithms which embedded to the components of WSN. Figure 6.12 shows the node-node and node-mv communication and a main station that collects data from the network.

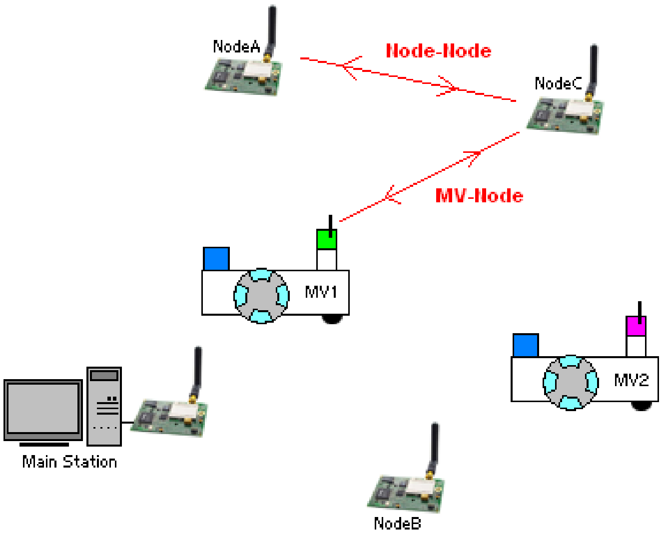


Figure 6.12 : Node-node and node-MV communication and central station

6.4.1 Node-node and node-MR communication

In this part, there is a quite little software embedded to the main processor of the wireless node module. The module is shown in Figure 6.13 below.

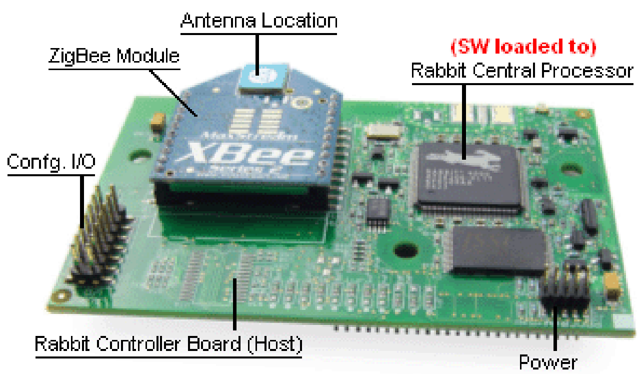


Figure 6.13 : ZigBee node with antenna module

The algorithm which is performed by wireless ZigBee node consists of RX and TX parts. RX part is used for receiving and TX part is used for transmitting. The software that is embedded to the nodes is performs the following procedure:

Node-Node communication procedure flow:

- Listen node signals around
- Record the information that received from neighbor nodes
- Calculate the own value in the hop-count network by received information
- Record the RSSI value of the neighbor nodes
- Set the broadcast signal frame as following :
    - **NodeID%HC#.RSSI%PQV…000**
- Broadcast the own values around to other nodes in range

Node-MV communication procedure flow:

- Listen MV signals from around
- Record the information that received from a MV
- Set the transmit signal frame as following :
    - **NID%HC#.RSSI%Neigh1ID:HC#:RSSI% Neigh1ID:HC#:RSSI...000**
- Transmit the node data to the MV

Table 6.1 : Received signals received by mobile robot at initial three steps

T	**MR Received**	**Description**
1	N/A	No signal received / "I CAN NOT HEAR ANY NODE"
2	N1%6.43%9:7:51%2:8:14000	**MR Hears :** Node-1 with H.C no: 6 and RSSI value is: 4.3 and N-1 Hears : Node-9 with H.C. no: 7 and RSSI value is: 5.1 and N-1 Hears : Node-2 with H.C. no: 8 and RSSI value is: 1.4
3	N1%6.29%9:7:51%2:8:14000 N9%7.68%1:6:51%4:3:60%14:10:89	**MR Hears :** Node-1 with H.C no: 6 and RSSI value is: 2.9 and N-1 Hears : Node-9 with H.C. no: 7 and RSSI value is: 5.1 and N-1 Hears : Node-2 with H.C. no: 8 and RSSI value is: 1.4 **MR Hears:** Node-9 with H.C no: 7 and RSSI values is: 6.8 and N-9 Hears: Node-1 with H.C..no: 6 and RSSI value is : 6.0 and N-9 Hears : Node-14 with H.C..no: 10 and RSSI value is : 8.9

Three of received data samples in order are shown and described in Table 6.1. At the first step (we can say that is first second if we set frequency as 1 second) mobile robot did not hear any signal from its surroundings. At the next step mobile robot started to hear only one signal from Node-1 in range. Received data includes node id as "1" and hop count number as "6" and RSSI value as 43 that also represents distance from mobile vehicle to the Node-1. In addition, Node-1 sent the received data from neighbor nodes around. That is called as "*RSSI array of a node*". In this array mobile robot achieves the neighbor nodes of Node-1, hop count numbers of neighbor nodes and RSSI value between Node-1 and neighbor nodes. That also represents the distance between Node-1 and its neighbor nodes.

The data exchange between nodes is applied only one time, once the network is initialized. When the numbering and calculations are finished up at each node which is called as "network settlement"

After initialization of the WSN setup, nodes in the WSN broadcast an "OK" signal and then mobile vehicle can start to travel through WSN from the initial position that is user defined or randomly.

During the travel of the mobile vehicles, nodes guide them to decide next step and follow the best route. Each node is only aware of its neighbor nodes around and can communicate with these nodes. Every node chain including several ZigBee nodes produces a specific RSSI average value that indicates the path quality value (PQV)

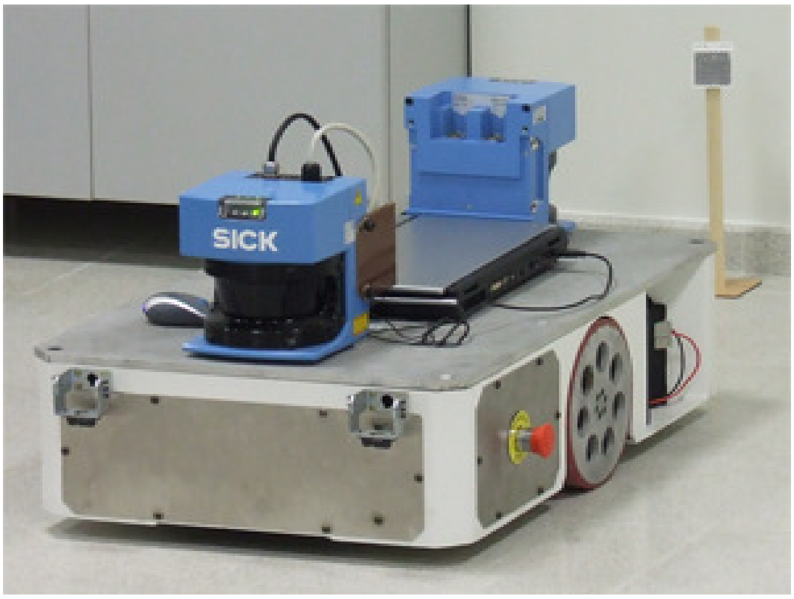


Figure 6.14 : A mobile vehicle (AGV) that can be used in an indoor WSN

During the travel of mobile vehicle, it communicates with the nodes in its range until it reaches the destination. The nodes in MV's range provide hop-count values to let MV choose largest number in the received numbers.

The communication frequency of mobile vehicle is set by user such as 1 second up to 1 minute or more. The frequency can be set large due to the task that mobile vehicle acts. If mobile vehicle needs to specify each step in a short time and network has a high density of the nodes, communication frequency would be higher.

The nodes in the wireless sensor network continuously listens signals that may come from mobile vehicles but only broadcasts signals once any request is received. This method also prevents undesired power consumption and data collision as well.

The software which is embedded at the MV provides direction information to physical controllers to move the vehicle. The software which is responsible to control vehicle wheels runs on the processor retrieves parameters from communication part.

Software which is embedded to the MV runs continuously and has a frequency defined by user, 10 seconds is chosen in this application study.

6.4.2 Communication geometry

In this project, there were no obstacles assumed. Mobile vehicles are moving on 2 dimensional indoor areas. The modules with antennas were hosted are located at the ceiling and this method created a new 2 dimensional surface parallel to the main floor that mobile vehicles are moving on.

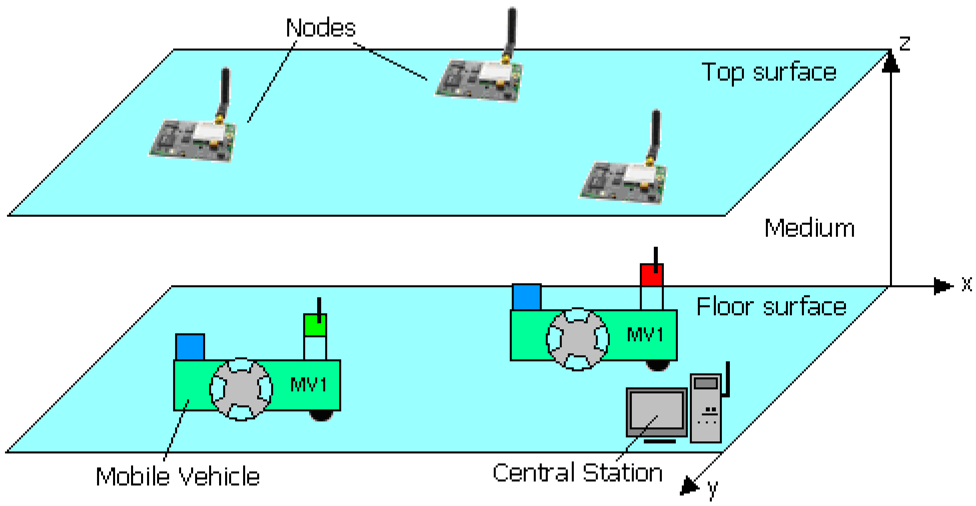


Figure 6.15 : Two surfaces of system

Although, there are two surfaces the system works on the medium between these surfaces. Because the transmitted and received signals move in this medium. But, because of level difference, localization algorithms must consider three dimensional calculations. If the nodes were located on the floor level which is the same with mobile vehicles, coverage area and distance calculations by both RSSI and others would be 2 dimensional systems.

The node-node communication is done at the top (ceiling) surface and mobile vehicles move on the floor surface as 2-dimensional. The RF signals are transmitted in the medium between floor and top surfaces as 3-dimensional (Figure 6.15).

R_{tn} is the real position of mobile vehicle at t_n time and R'_{tn} is the projection position of mobile vehicle at the same t_n time. Since the ground clearance of the antenna which is located on mobile robot never changes, we can consider the distance between antenna and nodes is constant. This helps us to avoid complex 3-dimensional calculations done in the processor of mobile vehicle.

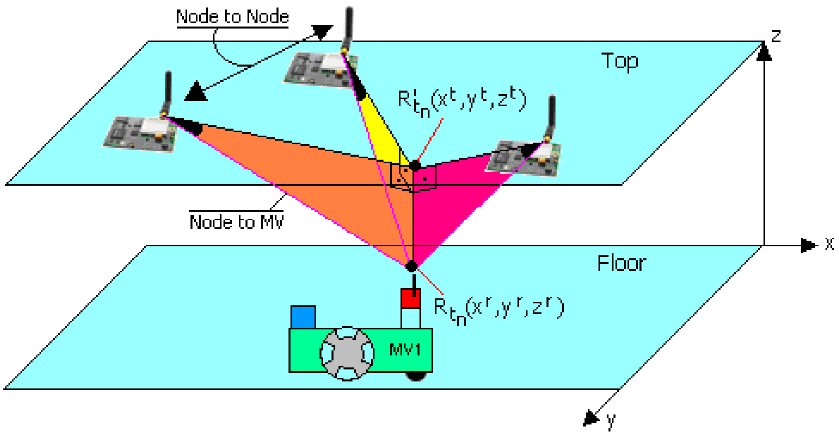


Figure 6.16 : Communication lines in 3-dimensional volume

As seen in Figure 6.16, distance to MV antenna only depends on the distance to R' position. So we can evaluate all calculations and algorithms based on $\mathbf{d_{NaR'}}$ value.

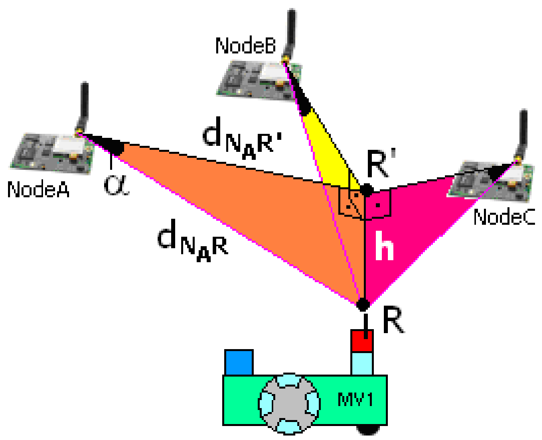


Figure 6.17 : Geometrical view of MR in range of three nodes

Please note that because of h is constant , the software that developed as the simulation of the system considers only 2-dimensional platform and gets the values from network relative by R' coordinates, without putting z into play. $d_{NAR'}$ value can be used instead of d_{NAR} value.

6.5 Determining the Direction of Mobile Robot

The direction of the mobile vehicle is not considered and put in calculations. Only the coordinates of antenna of mobile vehicle is considered as MV node. To determine the direction of the mobile vehicle we need to locate more than 1 antenna which provides reference point for each other. Thus the direction of the mobile vehicle can be determined by the signals of each antenna.

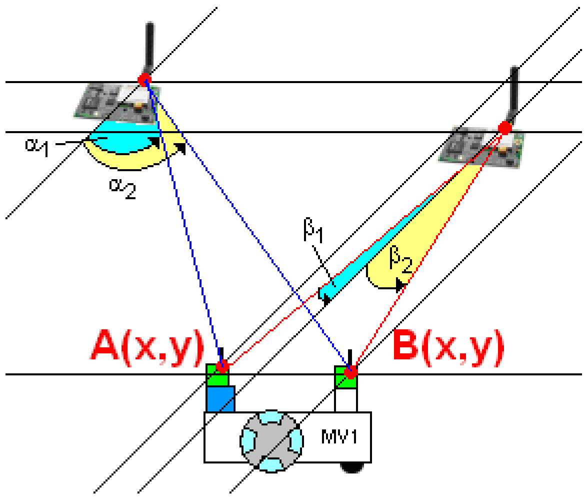


Figure 6.18 : Geometrical view of mobile vehicle with 2 antennas

Figure 6.18 shows the geometrical view of two antenna mobile vehicle communicating with 2 nodes. *A(x,y)* and *B(x,y)* indicates the antenna coordinates in the 2 dimensional surface.

Since each antenna sends different signals, nodes can recognize which is the front side and which is the back side antenna of mobile vehicle. By the distance between each antenna of the mobile device to the node, the direction can be calculated. One of the antenna on the mobile vehicle only sends "*I AM REAR* (Data string is '*REAR*')" and the other one which is at the front side sends all information in packets.

This information that received by the sensors are used to determine the direction of the mobile robot and processed in each sensor node individually. The output of the calculation is added to the packet frame in the broadcasting signal of each sensor node as an integer value representing *x* angle.

6.6 Application Results

Following graphs show the experimental results of the application such as functional relation between RSSI and distance at Figure 6.14. At the other hand Figure 6.15 is the standard deviation of the ranges as a function of the signal strength. Figure 6.16 shows the change of distance (a) and RSSI value (b) on the timeline. Each color indicates different node.

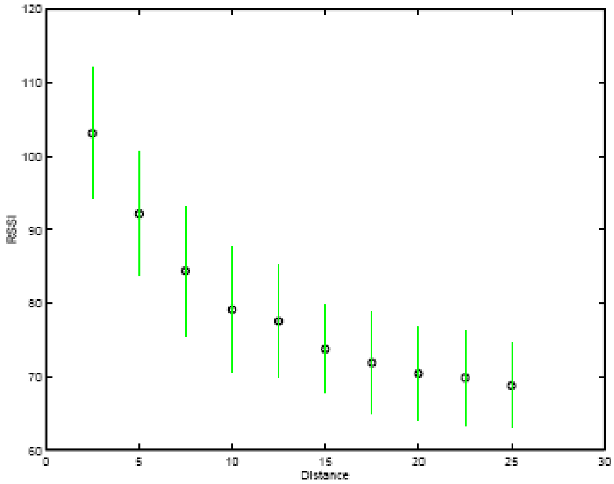


Figure 6.19 : Signal strength measurements as a function of the distance.

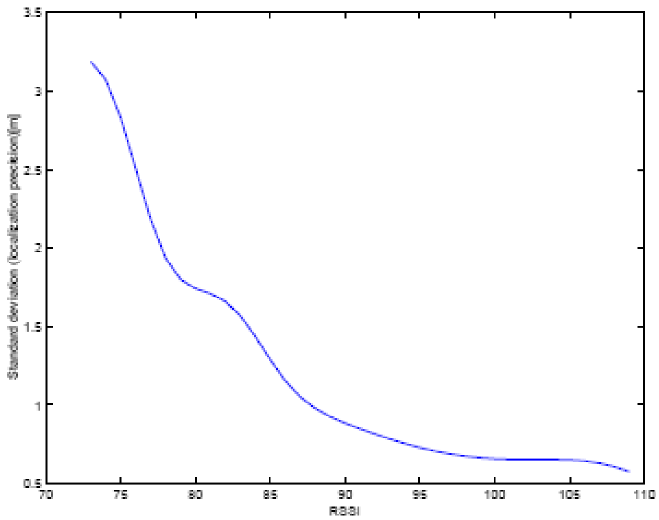


Figure 6.20 : Standard deviation of the ranges as a function of the signal strength

At figure 6.21 below, the approximation of mobile robot to the sensor nodes is shown. Mobile robot moves closer to the node which has a greater hop count number and ignores the nodes which have smaller numbers. That's why Node-2 (N2) was ignored and mobile robot did not move towards to that node.

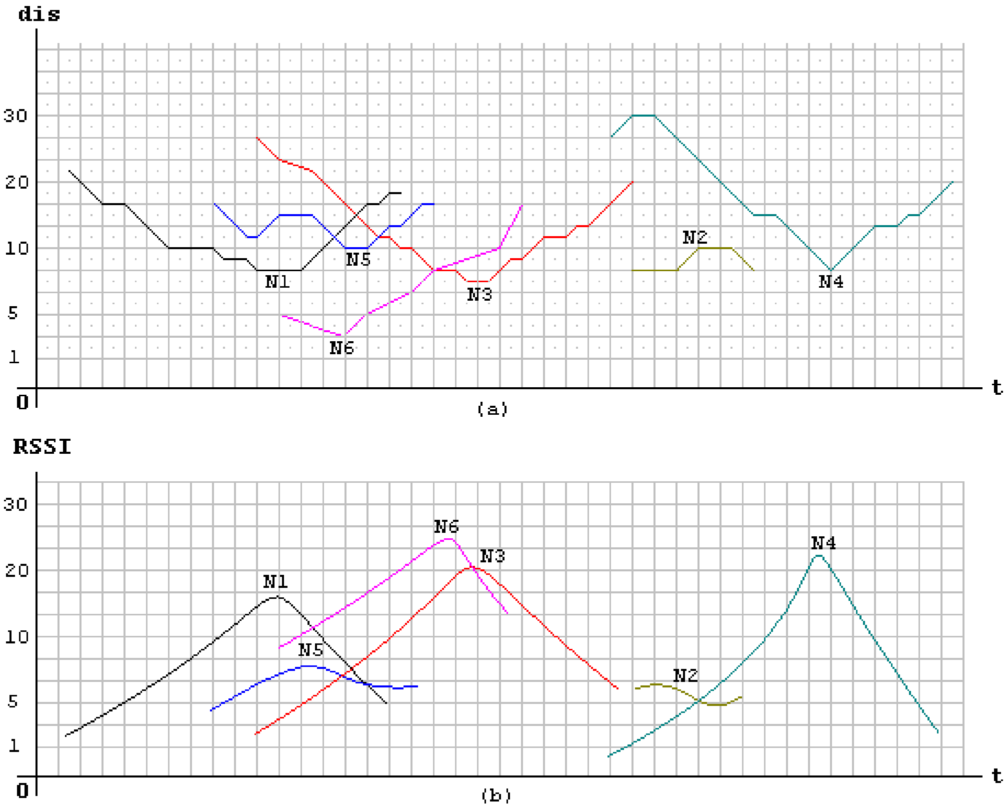


Figure 6.21: The distance and RSSI values during mobile robot's travel

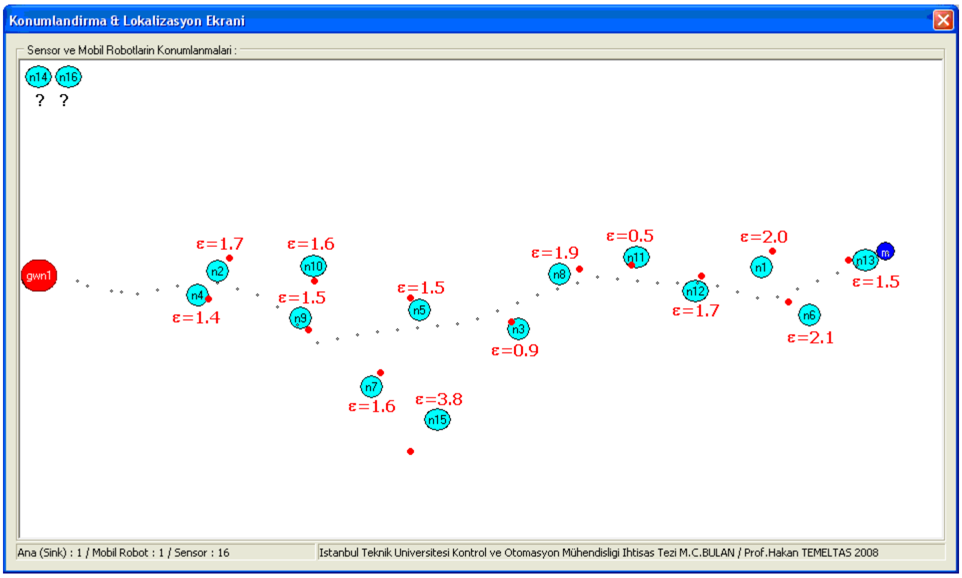


Figure 6.22: The estimated and exact positions of nodes determined by mobile robot

Figure 6.23: The actual and robot's paths to the target position

Figure 6.22 and 6.23 shows the exact and estimated locations of the nodes. The node locations which are out of range (node 14 and 16) could not be determined in this sample. The accuracy of determining the node positions depending on the density of nodes in range. The locations of the nodes are determined relatively to mobile robot's position. If mobile robot can hear only one node, this node's location cannot be determined.

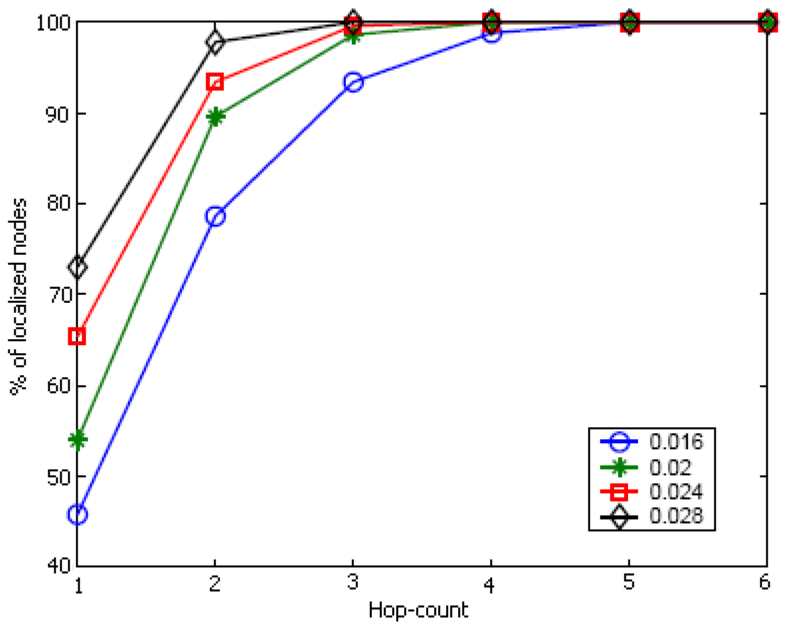


Figure 6.24 : Percentage of nodes that can be localized for different node density

Figure 6.24 shows the percentage of node localization by hop-count. Depending on the density of the nodes, localization algorithm can eliminate the possible errors and could find the node locations with a better accuracy.

The change of accuracy, in other words the localization error by node count in the same range (density of nodes in the range of mobile robot) is shown in Figure 6.25 below. Average localization error reduces by more node density and settles after 4 or 5 nodes. Because 4 or 5 nodes are enough to complete localization estimation algorithm. Different lines represent different network densities including 0.016, 0.02, 0.024 and 0.028, which correspond to average degrees of 6, 8, 10 and 12, respectively. Figures also show that, regardless of the network density, the accuracy of the localization improves along with the hop length for the first several hops. However, the improvement is not significant after 5–6 hops.

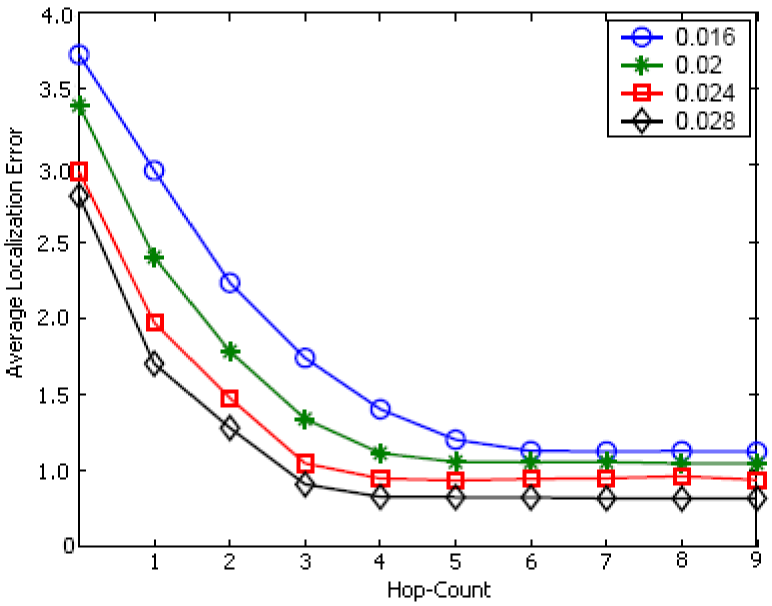


Figure 6.25 : Average localization error by number of hops

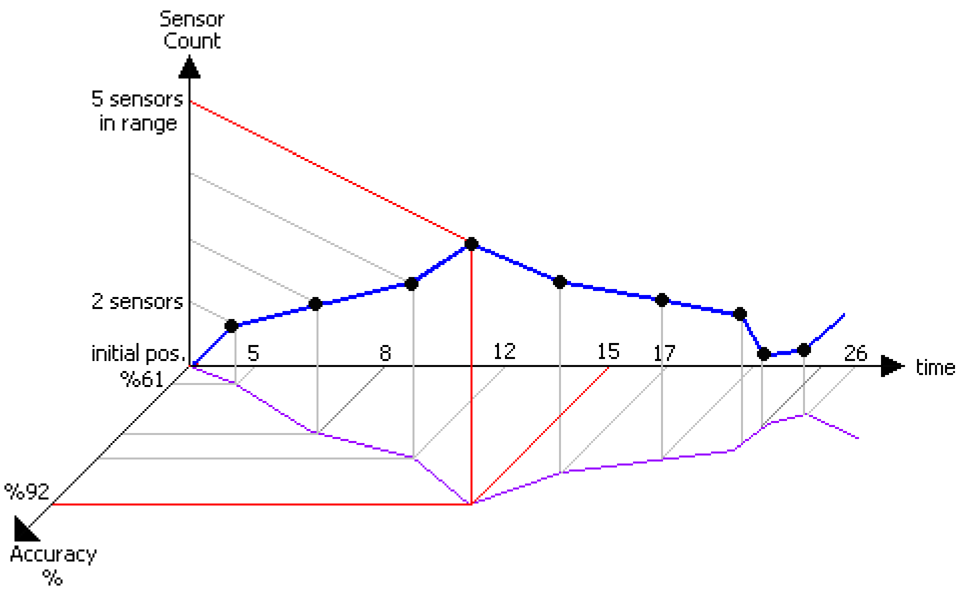


Figure 6.26 : Sensor Count in range and obtained accuracy of localization by time

7. CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

We considered an indoor project in this thesis that nodes are randomly distributed to the area and initially don't know their positions also the area is unknown. We saw that navigating a mobile robot or vehicle in an indoor wireless sensor network needs a robust signaling system and localization method to be able to create the optimum path. The destination object was also a static node which propagating a specific signal (means "I AM DESTINATION") different from other nodes in the network. The signals will be disturbed by the other effects and communication algorithm should be well designed that including confirmation protocols in data frames and ability to confirm by whole data. Also in our project we used sensor nodes which can process their own algorithms that cause additional power consumption by the way, because each sensor node includes power unit (batteries). Centralized sensor networks are more difficult to deploy for ad-hoc networks or might be impossible for military applications.

We developed a small code to let the nodes which are out of range (alone nodes) would get sleep mode and broadcast only base signal ("I am here") in very low intervals.

We saw that the nodes should be closer (in range) to be aware of neighbor nodes. That means the density of nodes highly affects the accuracy at the localization and needs at least 4 nodes around without mobile robot or 3 sensor nodes and one mobile robot to create multilateration. Otherwise, there can be some nodes that out of serve, because they cannot hear any signal from its surroundings and cannot help robot to decide its route. Also the accuracy of node localization depends on the density of nodes in the same range.

However we did not consider obstacles in the area, we need proximity sensors that also measuring distance to the obstacle on the robot to avoid walls at 2D bounded surface. The resultant force vector that is generated by both sensor signal and obstacle signal provides mobile robot direction and speed rate.

Also if we planned to develop a WSN for a factory, we would need to localize the nodes and robots with high accuracy up to 5-10 centimeters and nodes would need to know their positions. Because there will be some actions critical for short distances. That would enforce us to use the devices which have high quality of signal capabilities and very frequently update mobile robot location data. ZigBee devices have required enough quality of signals to be processed for localization up to miliseconds of update times. Also ZigBee devices provide a well communication channel with the communication protocol that we developed. Other communication protocols would include more payloads and would cause data collision. Regarding to small packet size of data that are exchanged between nodes and mobile robot, ZigBee spec became the best solution for these kinds of applications. Also cost advantage is the other important goal.

At the other hand, if we would like to determine the direction of mobile vehicle in the wireless sensor network, we would use more than 1 antenna on the vehicle. Instead of that, we may need sector antenna on the vehicle. But at this time, we would miss entire range area of radius of mobile vehicle. Although we used omni directional antenna on the mobile robot, position estimation by looking at the previous location algorithm helps us to locate our mobile vehicle in a better accuracy.

In addition, PQV provides our mobile robots to choose the better (shorter) one to follow in their route when mobile robot hears more than 1 signal which are same. For static nodes PQV can provide a great help to drawing best routes in the application area. Also PQVs can be dynamically updated at every time to iterate the best solution.

Also if we would perform localization experiments with a mobile robot it has been ascertained that grid-based Markov localization is more robust than Kalman filtering while the latter (given good inputs) is more efficient and accurate than the former. A combination of both approaches is likely to inherit the advantages of the underlying techniques. More recently, new localization methods are being employed with particle filters are presented.

In addition, the system was considered as an absolute 2D surface in this project, but in an outdoor system that sensors have different altitudes causes signal collisions and wrong calculations. So, z coordinate (altitude) would be put into play in calculations. But we may not need if the locations of nodes were known.

We aimed mobile vehicle only to find a target in the wireless sensor network in this thesis. In addition mobile robot can collect data from the network and go back to the initial position with the collected data.

As another scenario, if we assume that sensor nodes which randomly distributed on the area would have not create a chain from target to the initial robot position because of possible big gaps somewhere in the network. We offer to distribute some new sensors randomly to the area at that situation. For example in a big terrain, additional sensors would be thrown from a plane if sensors could not create a full path to the target position or mobile robot could not successfully keep move ahead. But however, that never guarantee an exact solution for a stuck situation.

7.2 Future Works

We prepared a prototype project that may be base for large projects, can be used in military or disasters. Also this would be good example for the technologies that trending to human-less factories.

By developing better skilled mobile robots, some difficult tasks may assign to the robots in seriously dangerous or hard issues. Hardware capabilities of mobile vehicles and also sensor nodes can accomplish works in factories or huge outdoor area.

In our project we talked about a mobile vehicle/robot which is being navigated in a wireless sensor network with an omni directional. This put us in a situation that we must use more than 2 static sensors to localize the robot by geometric calculation and RSSI. But if we did put the sensing area pieces, and give a unique code for each piece, that would provide us a really easier way to position the robot or sensors.

7.3 Spinning Antenna Approach

In this approach, the RF antenna located on the mobile robot is turning clockwise and sends different signal codes at each slice of 90 degrees. Thus localization builder will only parse the related codes and vehicle will be able to get the sensor in which part of range. This will save more processing time and reduce sensor could used in this sensor network. Furthermore, this can be upgraded to more sides using systems. Namely, if we use more codes and more steps for motors which is spinning the RF antenna, we will have more accuracy for positioning. Through to ideal system, a 360 degree step motor will send 360 different codes around and will get the exact sensor position.

REFERENCES

[1] **Whitehouse, C.D.,** 2002. The Design of Calamari: an Ad-hoc Localization System for Sensor Networks, University of California at Berkeley, USA

[2] **Dellaert, F., Fox, D., Burgard, W. and Thrun, S.,** 1999. Computer Science Department, Carnegie Mellon University, Pittsburgh PA,USA

[3] **Lee, W., Kim, M., Yee, W. and Eom, D.,** 2000. Indoor Navigation System for Mobile Robot using Wireless Sensor Network, Korea University, Korea

[4] **Moses, R.L., Krishnamurthy, D. and Patterson, R.M.,** 2002. A Self-Localization Method for Wireless Sensor Networks, Dep. of Electrical Engineering, The Ohio State University, 2015 Neil Avenue, Columbus, OH 43210, USA

[5] **Kortenkamp, D. and Weymouth**, T., 1994. "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Proceedings of the AAAI.*

[6] **Simmons, R. and Koenig, S., 1995.** "Probabilistic robot navigation in partially observable environments" in *Proceedings of the Int.Joint Conf. on Artificial Intelligence*.

[7] **Holger Karl.,** 2005. Ad hoc and Sensor Networks, Paderborn University

[8] **Priyantha, N.B., Balakrishnan, H., Demaine, E.D. and Teller, S.,** 2005. Mobile-assisted Localization in Wireless Sensor Networks, MIT Computer Science and Artificial Intelligence Laboratory, USA

[9] **Bah, B., Jungabel, E., Kowalska,M., Leithäuser, C., Pandey, A. and Vogel, C., 2009.** Odometry for train location.

[10] **Ahmed, A. Ahmed., Shi, Hongchi. and Shang**, Yi., 2006. A New Hybrid Wireless Sensor Network Localization System, Missouri, Columbia, USA

[11] **Jamieson, Phil., 2007.** ZigBee Application Profiles, ZigBee Open House, Paris, France

[12] **Patwari, N., Hero, A.O. and Kyperountas, S.,** 2005. Cooperative Geolocation of Wireless Sensors, University of Michigan: Ohio State Univ., Motorola Labs., USA

[13] **Chiti, F., Fantacci, R., Menci, S. and Zappoli, A.,** 2007. Cooperative Localization Protocols for Wireless Sensor Networks, University of Florence, Italy

[14] **Prof.Dario, P., Dr.Canelli, N., Prof. Fantacci, R. and Dr. Fontanelli, R.,** 2005. DUSTBOT Project, EU (See also http://www.dustbot.com)

[15] **Kotay1, Keith., Peterson, Ron. And Rus, Daniela.,** 2004. Experiments with Robots and Sensor Networks for Mapping and Navigation, Massachusetts Institute of Technology, Cambridge, MA, USA

[16] **Golatowski, F.,** 2005. Localization in Wireless Sensor Networks and its Applications, Center for Life Science Automation, Bilbao Spain

[17] **Huang, H., Huang, Y. and Ding, J.W.,** 2006. An Implementation of Battery-aware Wireless Sensor Network Using ZigBee for Multimedia Service, National Cheng Kung University, Tainan, Taiwan
[18] **Tadakamadla, S., 2006.** Indoor Local Positioning System For ZigBee, Based On RSSI, Mid Sweden University, Sweden
[19] **Sugano, M., Kawazoe, T., Ohta, Y. and Murata**, M., 2006. Indoor Localization System Using RSSI Measurement Of Wireless Sensor Network Based On ZigBee Standard, Osaka Prefecture University, Japan
[20] **Reichenbach, Frank. And Timmermann, Dirk.,** 2006. Indoor Localization with Low Complexity in Wireless Sensor Networks, Institute of Applied Microelectronics and Computer Engineering 18119 Rostock-Warnemuende, Germany
[21] **Ahn, H. and Yu**, W., 2007. Indoor mobile robot and pedestrian localization techniques, Department of Mechatronics, Gwangju Institute of Science and Technology, Korea
[22] **Negenborn, Rudy.,** 2003. Robot Localization and Kalman Filters.
[23] **Chraibi,** Y., 2005. Localization in Wireless Sensor Networks- Graduation Thesis, Sweden 2005
[24] **Grossmann, R., Blumenthal, J., Golatowski, F. and Timmermann**, D., 2006. Localization in ZigBee-based Sensor Networks, CELISCA, Center for Life Science Automation, Germany
[25] **Sarigiannidis**, Georgios., 2006. Localization For Ad-Hoc Wireless Sensor Networks, Technical University Delft, the Netherlands
[26] **Prof. Dr. Zuehlke**, Detlef., 2008. SmartFactory – from Vision to Reality in Factory Technologies. Technology Initiative e.V. Kaiserslautern, Germany
[27] **Sugano, M., Kawazoe, T., Ohta, Y. and Murata**, M., 2006. Indoor Localization System using RSSI Measurement of Wireless Sensor Network based on ZigBee Standard, Japan
[28] **Borenstein, J., Everett, H.R. and Feng, L., 2002.** Navigating Mobile Robots: Sensors and Techniques (Book).
[30] **Woodman**, **Oliver.J**., 2007. An introduction to inertial navigation, University of Cambridge, Computer Lab.
[31] **Eom, D., Jang, J., Kim, T. and Han**, J., 2006. A VR Game Platform Built Upon Wireless Sensor Network, Korea University, Seoul, Korea.
[32] **Suau, P.,** 2005. Robust Artificial Landmark Recognition Using Polar Histograms, Departamento de Ciencia de la Computaci´on e Inteligencia Artificial, Spain.
[33**] Luo**, **R. C.,** 1996. Outdoor Landmark Recognition Using Hybrid Fractal Vision System and Neural Networks, North Carolina State Univ.
[34] **Zanella, A., Menegatti, E. and Lazzaretto**, L., 2007. Self-Localization of Wireless Sensor Nodes By Means of Autonomous Mobile Robots Book, Wireless Communications 2007 CNIT Thyrrenian Symposium ,US.
[35] **Bulan, M.C. ,Temeltas, H.**, 2009. Localization of Mobile Robots in a Wireless Sensor Networks Based on Hop-Count and Artificial Potentials Fields, ICCA09.
[Url-1] <http://www.zigbee.org> accessed at 18.02.2009
[Url-2] <http://www.circuitcellar.com> accessed at 19.02.2009

[Url-3] <http://en.wikipedia.org/wiki/UCAV> accessed at 19.02.2009
[Url-4] <http://en.wikipedia.org/wiki/MQ-1_Predator> accessed at 21.02.2009
[Url-5] <http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle> accessed at 1.11.2008
[Url-6] <http://www.tinyos.net> accessed at 12.03.2009

CURRICULUM VITA

Candidate's full name : Mustafa Çağlar BULAN

Place and date of birth : Bingöl 31.10.1980

Parmanent address : Namık Kemal Mah. Türbedar Sok. No:16 Ümraniye İSTANBUL

Universities and Colleges attended : M. Kemal University Electrical and Electronics Engineering B.Sc.