

A SELF-ORGANIZING HYBRID SENSOR SYSTEM WITH DISTRIBUTED DATA FUSION
FOR INTRUDER TRACKING AND SURVEILLANCE

by

RAVISHANKAR PALANIAPPAN
M.S. University of Central Florida, 2003

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Modeling and Simulation
in the School of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring 2010

Major Professor: Parveen Wahid

© 2010 Ravishankar Palaniappan

ABSTRACT

A wireless sensor network is a network of distributed nodes each equipped with its own sensors, computational resources and transceivers. These sensors are designed to be able to sense specific phenomenon over a large geographic area and communicate this information to the user. Most sensor networks are designed to be stand-alone systems that can operate without user intervention for long periods of time. While the use of wireless sensor networks have been demonstrated in various military and commercial applications, their full potential has not been realized primarily due to the lack of efficient methods to self organize and cover the entire area of interest.

Techniques currently available focus solely on homogenous wireless sensor networks either in terms of static networks or mobile networks and suffers from device specific inadequacies such as lack of coverage, power and fault tolerance. Failing nodes result in coverage loss and breakage in communication connectivity and hence there is a pressing need for a fault tolerant system to allow replacing of the failed nodes. In this dissertation, a unique hybrid sensor network is demonstrated that includes a host of mobile sensor platforms. It is shown that the coverage area of the static sensor network can be improved by self-organizing the mobile sensor platforms to allow interaction with the static sensor nodes and thereby increase the coverage area. The performance of the hybrid sensor network is analyzed for a set of N mobile sensors to determine and optimize parameters such as the position of the mobile nodes for maximum coverage of the

sensing area without loss of signal between the mobile sensors, static nodes and the central control station.

A novel approach to tracking dynamic targets is also presented. Unlike other tracking methods that are based on computationally complex methods, the strategy adopted in this work is based on a computationally simple but effective technique of received signal strength indicator measurements. The algorithms developed in this dissertation are based on a number of reasonable assumptions that are easily verified in a densely distributed sensor network and require simple computations that efficiently tracks the target in the sensor field. False alarm rate, probability of detection and latency are computed and compared with other published techniques. The performance analysis of the tracking system is done on an experimental testbed and also through simulation and the improvement in accuracy over other methods is demonstrated.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Parveen Wahid for her constant support and encouragement to complete this research work. I would also like to thank my committee members Dr. Peter Kincaid, Dr. Thomas Clarke and Dr. Bala Jaganathan for their time and advice. I would also like to thank the faculty and staff of Institute for Simulation and Training who helped me with hardware support and data collection. I would like to acknowledge the National Science Foundation, Japan Institute for Promotion of Sciences and Fukuoka Institute of Technology who sponsored a part of this research through their fellowship programs. On a personal level, I would like to thank my parents for encouraging me to finish this work.

TABLE OF CONTENTS

LIST OF FIGURES	viii
LIST OF TABLES	x
CHAPTER ONE: INTRODUCTION	1
Sensor Network Applications	1
Military Applications	2
Environmental Applications	2
Health Application	2
Factors influencing Sensor Network Design	3
Fault Tolerance	3
Scalability	3
Production Costs	4
Hardware Constraints	4
Transmission Media	5
Data Processing	6
Sensor Network Communication Architecture	6
Static Sensor Network	8
Mobile Sensor Networks	8
Deployment	9
Network Repair	9
Energy Harvesting	9
Localization	10
Sensor Devices	10
Mica Mote Sensors	11
Ultra Wide Band Sensor	11
Inertial Measurement Units	13
Sensor Fusion	14
CHAPTER TWO: LITERATURE REVIEW	17
CHAPTER THREE: TRACKING USING A HYBRID NETWORK	23
Design of Sensor Network – Static and Mobile Network	24
Mica Mote Static Sensor Network	24
Ultra-Wide Band Static Sensor Web	28
Sensor Unmanned Ground Vehicle Mobile Sensor Network	32
Accuracy of the Three Tracking Systems	36
Hybrid Sensor Network	37
Self Organizing and Localization by the Hybrid Sensors	37
Target Tracking in the Hybrid Sensor Network	41
Constant False Alarm Rate	43
Data Fusion using Kalman Filtering	44
Simulation and Testing	46
CHAPTER FOUR: RESULTS	47
Localization of Reference Nodes	47

Connectivity Metric	49
Received Signal Strength Model	50
Target Tracking with Static Nodes	57
Detection in the Presence of Noise	59
Fading Channels	60
Rayleigh Fading Effect	61
Rician Fading Effect	62
Gaussian Probability Density Function	63
Other Channel Related Issues	64
Cramer Rao Lower Bound on estimation of sensor node location	65
Probability of False Alarm	66
Simulation Results	68
Probability of Detection Vs. Number of Nodes in the Network	68
Detection Range and Signal-to-Noise Ratio	70
Constant False Alarm Rate	71
Hardware	73
Programming Aspects	75
TinyOS Components	76
Compiling and Installing with Mica2 Sensor	81
Connecting Multiple Nodes to form Sensor grid	83
Example of Target Tracking in Static Sensor Network	85
Target Tracking with Mobile Reference Nodes	87
Error Correction using Kalman Filtering	91
Evaluation of the Sensor System	95
Number of Reference Nodes	96
Number of Frequencies	97
Number of Transmit Power Levels	98
Number of Reference Signatures	98
Number of Data Samples	99
Cost Analysis for Static Sensor Deployment	100
Cost Analysis for Mobile Sensor Deployment	102
CHAPTER FIVE: CONCLUSION	104
REFERENCES	105

LIST OF FIGURES

Figure 1: Components of a sensor node.....	5
Figure 2: Sensor network communication architecture	7
Figure 3 Mica mote sensor with different sensing capabilities.....	11
Figure 4 Ultra-wide band transceivers.....	13
Figure 5: Inertial Measurement Unit.....	14
Figure 6: Decision fusion type of sensor fusion system	15
Figure 7: Data fusion type of sensor fusion system.....	15
Figure 8: Target detection algorithm using mote sensor network	26
Figure 9: Sensor node data being collected by the central computer and the webcam	27
Figure 10: Sensor network configuration	28
Figure 11: Background scan amplitudes for the range of 800 bins	29
Figure 12: Threshold for varying Probability of False Alarm	31
Figure 13: Block diagram of Unmanned Ground Vehicle for reconnaissance and tracking.....	34
Figure 14: Unmanned ground vehicle used for the sensor network	34
Figure 15: GPU and INS feedback system for a mobile platform.....	36
Figure 16: Stage 1: Static nodes report position to central node	39
Figure 17: Stage 2: Central node dispatches mobile nodes	40
Figure 18: Stage 3: Central node organizes the static and mobile nodes into a grid	40
Figure 19 A hybrid static mobile intruder detection sensor network.....	42
Figure 20: Target tracking through the sensor web	43
Figure 21 Sensor data fusion using Kalman Filter.....	45
Figure 22: BER for BPSK Channel with Rayleigh and Rician fading	50
Figure 23: Location of initial anchor nodes on the sensor grid	53
Figure 24: Reference nodes that form the sensor grid	53
Figure 25: Sensor location estimation.....	55
Figure 26: Steps involved in tracking of target node.....	59
Figure 27: Simulated Rayleigh fading channel at 915 MHz.....	61
Figure 28: Simulated Rician fading channel probability function.....	63
Figure 29: Gaussian probability density function.....	64
Figure 30: Cramer-Rao Lower Bound on variance of position error.....	66
Figure 31: Normalized detection threshold versus probability of false alarm.....	67
Figure 32: Signal-to-Noise ratio versus Probability of Detection	69
Figure 33: Detection range versus Signal-to-Noise Ratio	71
Figure 34: Constant false alarm rate detection	72
Figure 35: CC1000 Chipset components	74
Figure 36: Message from Transmitting Sensor.....	75
Figure 37: Receiving Code for measurng RSSI.....	75
Figure 38: TinyOS Structure Implementation	76
Figure 39: Varying the frequency on the Mote sensor	78
Figure 40: Varying the frequency on the mote sensor	79
Figure 41: Varying frequency and transmit power on the reference nodes.....	80

Figure 42: MIB510 Serial Interface board with Mica2 Sensor.....	81
Figure 43: Compile messages on Computer indicating successful compile and install	82
Figure 44: Serial forwarder interface to computer.....	83
Figure 45: Sensor web grid with reference nodes and the target node	84
Figure 46: Target node tracked in the Sensor grid.....	85
Figure 47: Target out of range for detection the static reference node grid	86
Figure 48: Target node traversing the sensor grid but unable to be tracked.....	88
Figure 49: Mobile reference node used to continuously track the target.....	89
Figure 50: Second mobile sensor node movement to track target	89
Figure 51: Three mobile reference nodes for target tracking	90
Figure 52: Kalman filter process.....	91
Figure 53: Kalman filter operation in target node estimation.....	92
Figure 54: Kalman filter used for target tracking in presence of noise	93
Figure 55: Filtered error from sensor.....	94
Figure 56: Measured error on the sensor	95
Figure 57: Error distance (in feet) as the number reference nodes are increased	96
Figure 58: Target error distance as a function of frequency.....	97
Figure 59: Error distance of target as a function of transmit power	98
Figure 60: Target Error as a function of number of reference signatures.....	99
Figure 61: Latency as a function of Number of Data Samples.....	100
Figure 62: Cost function for increasing the number of sensors	101
Figure 63: Cost of Deployment for Mobile sensor node	103

LIST OF TABLES

Table 1: Types of sensors and their sensing modalities.....	10
Table 2: UWB sensor web simulation	30
Table 3: Number of False Alarms for given PFAs	31
Table 4: Type of sensor network and accuracy of intruder location.....	37
Table 5: The message format of the Mote sensor	52
Table 6: Various Parameters used in Simulation Model	54
Table 7: Frequency and power advantage and disadvantage.....	79
Table 8: Output from tracking Sensor software.....	87

CHAPTER ONE: INTRODUCTION

A wireless sensor network can be defined as a group of spatially distributed sensors wirelessly linked by a RF connection. These distributed sensors collectively gather sensory information such as light, temperature, pressure etc., which is then transmitted through the wireless link to a base or data collection station that processes the raw sensory data and gathers the desired information about the environment. With the recent development of micro-electro-mechanical (MEMS) devices, sensors have become ubiquitous in various fields for sensing and tracking. An advantage of sensors is that they can be deployed in inhospitable terrain, e.g. by randomly dropping them from an aircraft. On the other hand, this random deployment means that the sensors should be capable of organizing themselves into a wireless network in order to collect and transmit the data to a base station for information gathering. One of the primary constraints of sensor nodes is the need for low power consumption. Sensor nodes should be functional for many months if not years and must be capable of switching to power saving mode when not needed. All these requirements make sensor network technology a very interesting area for research and development.

Sensor Network Applications

Primarily developed for the military as a surveillance tool, wireless sensor networks have now proliferated to civilian applications such as environment and habitat monitoring, healthcare applications, home automation, and traffic control.

Military Applications

Sensor networks are integral part of the Command, Control, Communication, Computers and Intelligence systems (C4I). The rapid deployment, self organizing and fault tolerance capabilities make them an ideal candidate for many applications. Some of the major military applications for sensors include monitoring friendly or enemy forces, ammunition movement, tracking intruders, and reconnaissance in hostile terrain. Sensors can be attached to each entity in a battlefield and can be tracked for logistics and planning. In chemical and biological warfare, sensors can be used to detect changes in the chemical balance of the air and therefore set off alarms on a possible attack. One of the primary applications of sensors in military scenarios is for setting up virtual fences that can detect an intruder trespassing through a protected zone with hostile intent.

Environmental Applications

Environmental applications of sensor networks include tracking wild animals, livestock, growth of crops, planetary exploration, under water mapping, forest fire detection and various other applications. The sensor system will need to be tailored to the need of the specific application. For example, for monitoring crop growth the primary requirement would be to report on the growth pattern, therefore the sensor can operate in “sleep mode” for a majority of the time [1].

Health Application

Sensor networks can be used to provide tele-monitoring capabilities for keeping track of patients’ movements in hospitals and nursing homes. They can also be used to send reminders to patients for taking medication and for detecting vital signs such as heart rate, etc [2].

Factors influencing Sensor Network Design

Some of the challenges that need to be addressed in wireless sensor networks include the number of sensors needed to monitor a given area, size of the networks, fault tolerance, battery life, data collection and transmission methods, post processing, user display and decision making using the gathered data.

Fault Tolerance

Sensor nodes are very small and fragile devices that often fail due to physical damage or power loss. However the sensor network formed by the nodes themselves should be resilient enough to continue the data gathering and perform the required functionalities without interruption. Previous research in the area of sensor node reliability has estimated that the probability of sensor node not failing is given by the relation

$$R_k(t) = \exp(-\lambda_k t) \tag{1}$$

where λ_k is the failure rate of the sensor node k and t the time period [3]. Sensor network systems should be designed keeping this fault tolerance ratio in mind.

Scalability

The number of sensors deployed for studying a phenomenon is very dependant on the need and area to be sensed. High density sensor nodes can have high rate of failures without losing information. In other cases the sensors can be widely spaced apart to avoid detection or if the sensors have large ranges. The sensor density can be determined as

$$\mu(R) = (N\pi R^2) / A \tag{2}$$

where N is the number of scattered nodes; R is the range and $\mu(R)$ gives the number of nodes within the transmission radius of each node in given region A [3].

Production Costs

Since sensor networks typically consist of a large number of individual sensors the costs of each sensor node should be low. The cost of each node increases based on the functionality available on it such as transceivers, range of operation and the underlying hardware.

Hardware Constraints

A sensor node is made up of four basic components i). a sensing unit, ii). a processing unit, iii). a transceiver unit and iv). a power unit. The sensing unit usually consists of the sensor and an analog-to-digital convertor; the sensing data is converted to digital data and sent to the processing unit. This processing unit hosts the software responsible for connecting the sensor node to other nodes on the network. The transceiver unit contains the RF hardware module that wirelessly connects the node to the sensor network and finally, the power unit supplies power to all the other units. Figure 1 shows the various components of a sensor node in detail.

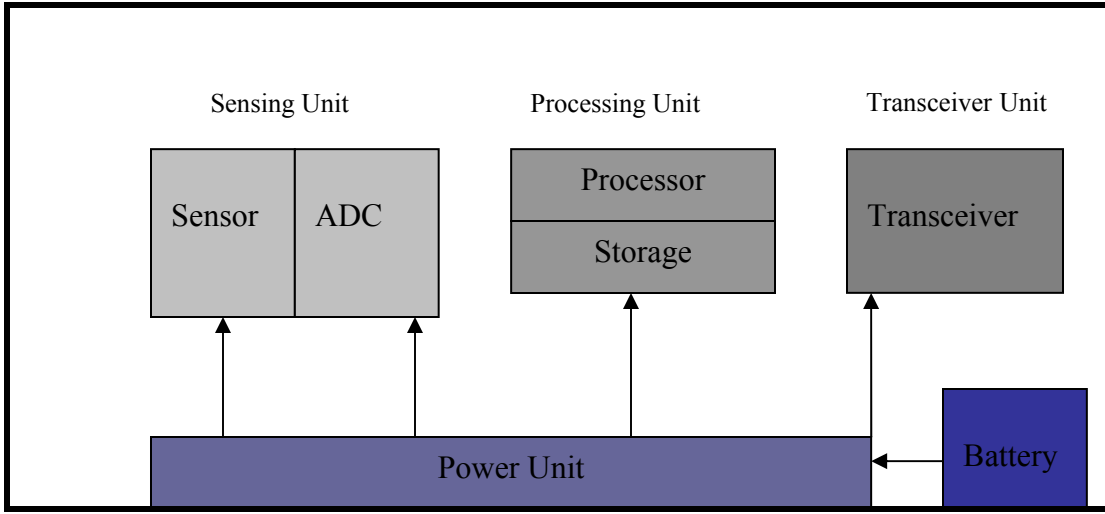


Figure 1: Components of a sensor node

Transmission Media

Most of the current sensor networks operate in the unlicensed Industrial, Scientific and Medical (ISM) band at 2.4 GHz. Although the ISM band is open for all electronics and modulation schemes, there are transmission power constraints and interference levels to adhere to. Some sensor systems like the Berkeley Mica mote devices use the 433 ISM band or the 915 MHz band. Of the three units in a sensor system the transceiver unit consumes the most power. There has been considerable research done on calculating the power consumption in sensor nodes. The standard formulation for radio power consumption in a sensor node given by

$$P_C = N_T [P_T (T_{ON} + T_{ST}) + P_{OUT} (T_{ON})] + N_R \{P_R (R_{ON} + R_{ST})\} \quad (3)$$

Where P_T is the power consumed by the transmitter P_R is the power consumed by the receiver, P_{OUT} is the output power of the transmitter, T_{ST} and R_{ST} is the start-up time, T_{ON} and R_{ON} is the number of time the transmitter and receiver are switched on per unit time [4] .

Data Processing

The energy spent on processing data is considerably smaller than for data communication. Current sensor technology is based on CMOS devices that switch between different modes to process the data. Using adaptive power schemes the power drawn by these devices can be controlled to the minimum required for processing the data. A simple expression has been formulated to estimate the power consumed in a sensor and is given by [5]

$$P_p = CV_{dd}^2 f + V_{dd} I_o e^{V_{dd}/nV_T} \quad (4)$$

where C is the switching capacitance, V_{dd} voltage swing and f is the frequency. The second term indicates the power loss due to leaky currents with I_o the current and V_T the threshold voltage.

Sensor Network Communication Architecture

Sensor nodes are usually scattered in a sensor field and each of the nodes has the capability to collect and route information to a central base station. All sensor nodes use the same network protocol stack for communication through the wireless medium. The protocol stack consists of the application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane and task management plane. Figure 2 below shows the sensor network communication architecture and the different layers in the system.

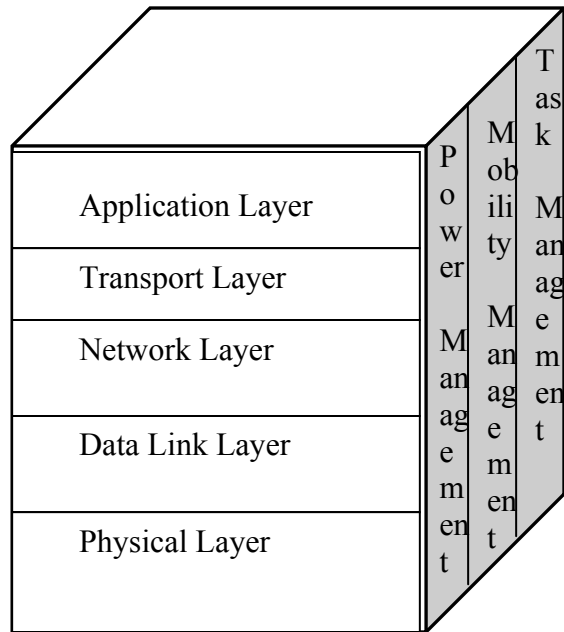


Figure 2: Sensor network communication architecture

Depending on the tasks different kinds of applications can be built on the application layer; the transport layer helps maintain the data flow; the network layer does the routing of the packets in the network; the data link layer makes the connection from one sensor node to the other and checks for errors and finally the physical layer is used for the modulation and transmitting the data. All these different layers are controlled by the management layers during the working of the sensor. The power management plane regulates the power supply to the different operations such that only the minimum power is utilized. The mobility management plane keeps track of the nearest neighboring sensor nodes and the basic topography of the sensor network. The management plane regulates the tasks the sensor is programmed to perform.

Static Sensor Network

Wireless sensor networks are expected to be extensively utilized in the future since they can greatly enhance our capability of monitoring and controlling the physical environment. Data collection and efficient routing are the primary operations of a sensor network. In most sensor networks the sensor nodes are static and have no mobility. They are placed in their location by some form of disposal method in either a random distribution or in a specific pattern to cover the entire region of interest. However if the environment is a hostile terrain or a disaster area deploying the static sensors cannot be done manually and they might have to be air-dropped from high altitudes. As a result the static sensor nodes are prone to failure due to mechanical defects and loss of power.

Mobile Sensor Networks

In a mobile sensor network each node is mobile with capability to move to a certain location on the sensor web as needed. The sensor nodes are normally mobile robots that have sensing, computation and communication capabilities. Using mobile nodes the sensor network coverage area of a static sensor network can be maximized. Mobile nodes can be unmanned ground vehicles (UGVs) that can be equipped with sensors such as wireless cameras, inertial sensors etc enabling the monitoring of the coverage area in a variety of ways. The ability of actively changing location can be used to mitigate/solve many of the problems of static sensor networks and for the optimal placing of a sensor for monitoring. The movement of the sensor nodes in wireless sensor networks is termed as “controlled mobility”. This type of mobile sensor is

different from traditional mobility that is associated with Mobile Ad-hoc Networks (MANET) in which the mobility is assumed and not predictable. In mobile sensor networks the movement of the nodes is user-controlled to achieve a goal or perform specific tasks. Mobile sensor networks can solve many fundamental problems associated with static networks. A few of them are described here:

Deployment

The mobile sensor nodes can be relocated at optimal locations for continuous monitoring of the environment. They can also align themselves such that the deployment is optimal to monitor the events under consideration.

Network Repair

One of the major problems with static networks is the lack of connectivity due to node failures. Having even a few mobile nodes in the network can solve the issue by relocating a node to an area where there is loss of network connectivity.

Energy Harvesting

There has been considerable interest in the area of finding renewable energy sources for powering the individual sensor nodes. Energy sources such as solar power, vibration and heat are being combined to supply the sensors with small amount of electricity to power them over long periods of time [6]. This is of particular interest for mobile sensors such that they can power themselves to move to a specific location.

Localization

Localization is a major research area in distributed robotics and network systems and many algorithms such as ‘Simultaneous Localization & Mapping’ (SLAM) [7] have been developed for static networks. It is possible to apply some of these algorithms to mobile sensor nodes.

Sensor Devices

There are many sensor devices that are being marketed for various applications in the military and civilian field. Some of the qualities of a sensor node include ease of installation, durability, large mean-time before failure and small size. A typical sensor node consists of a sensing device along with a RF trans-receiver to send or receive data from other nodes or the base station. Table 1 shows the different types of sensors and parameters that they measure

Table 1: Types of sensors and their sensing modalities

	Measured Quantity	Principle Technique
Physical properties	Pressure	Capacitive
	Humidity	Resistive
Motion properties	Position	GPS, contact sensor
	Velocity	Optical Encoder
Presence	Contact/proximity	Hall effect
	Vibration	Ultrasound

Mica Mote Sensors

The Crossbow mote sensors are one of the most popular sensor devices currently in the market. The operating frequency is at either 433 MHz or 916MHz with a data rate of 40 Kbits/sec with a range of 30 ft to 100 ft. Each node has a low power microcontroller processor with speed of 4MHz, a flash memory with 128 Kbytes, and SRAM and EEPROM of 4K bytes each. The operating system is Tiny-OS, a tiny micro-threading distributed operating system developed by UC Berkeley, with a NES-C (Nested C) source code language. The mote sensors are equipped with sensors for measuring light, temperature, pressure and humidity levels in the environment. Figure 3 shows a Mica2 Mote sensor with the various sensing modules.



Figure 3 Mica mote sensor with different sensing capabilities

Ultra Wide Band Sensor

An ultra wideband signal is an electromagnetic signal with a fractional bandwidth of greater than 0.25 with respect to the center frequency. Most narrowband systems carry information, also

called the baseband signal, as a modulation of a much higher carrier frequency signal. The important distinction is that the UWB wave form combines the carrier and baseband signal. Ultra wideband is also sometimes referred to as impulse or carrier-free transmission with large relative bandwidth. There is a distinct difference between spread spectrum and ultra wideband systems. Spread spectrum systems have a transmitted signal that is spread over a frequency band much wider than the minimum bandwidth required to transmit the information being sent [9]. A spread spectrum system takes a baseband signal with a bandwidth of only a few kilohertz and distributes it over a larger bandwidth but in the case of UWB systems the transmitters emit ultra-short monocycle wavelets with tightly controlled pulse-to-pulse intervals. The wavelet pulse widths are between 0.2 and 1.5 nanoseconds, corresponding to center frequencies between 5 GHz and 600 MHz, and pulse-to-pulse intervals of between 25 and 1000 nanoseconds. The systems typically use pulse position modulation. The pulse-to-pulse interval is varied on a pulse-by-pulse basis in accordance with two components: an information signal and a channel code. The TM-UWB receiver directly converts the received RF signal into a baseband digital or analog output signal. Typical data rate at distances of 10 meters is 9.6 Mbps. Figure 4 shows UWB devices that operate at 2.4 GHz frequency.



Figure 4 Ultra-wide band transceivers

Inertial Measurement Units

Inertial Measurement Units (IMUs) are capable of measuring acceleration in all three planes of motion and is a very useful for calculating direction of motion and speed of a node. While operating over the full 360 degrees of angular motion the IMUs provide orientation in matrix and quaternion formats. Figure 5 shows a MicroStrain 3DM-G device that is used for measuring different parameters.

The IMU incorporates:

3 accelerometer sensors to measure the acceleration on the X-Y-Z plane;

3 magnetometer sensors to measure magnetic fields;

3 rate inclinometer sensors to measure the rotation about their sensitive axis;

A temperature sensor



Figure 5: Inertial Measurement Unit

Sensor Fusion

Acquiring complete or adequate information about a target by a single sensor in a highly cluttered environment is a difficult task. On the other hand if multiple sensors are used to locate and track a target there is a higher probability of sensing, locating and discriminating the target. In a multi-sensor fusion system individual data from each sensor is processed to form an opinion on the target and information from the individual sensors is combined to form an aggregated situational awareness of the target location. This is known as decision fusion. The disadvantage with decision fusion is that the individual decisions at the sensors nodes are made from limited available information. Unlike the decision fusion system, a data fusion system involves combining and integrating sensor data from multiple sources to form a comprehensive view of the target location.

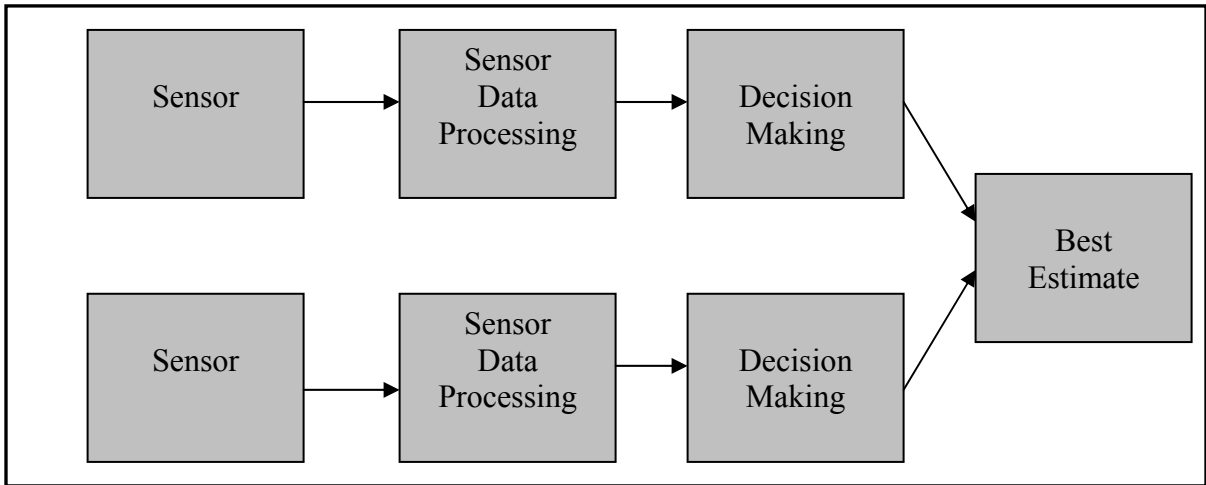


Figure 6: Decision fusion type of sensor fusion system

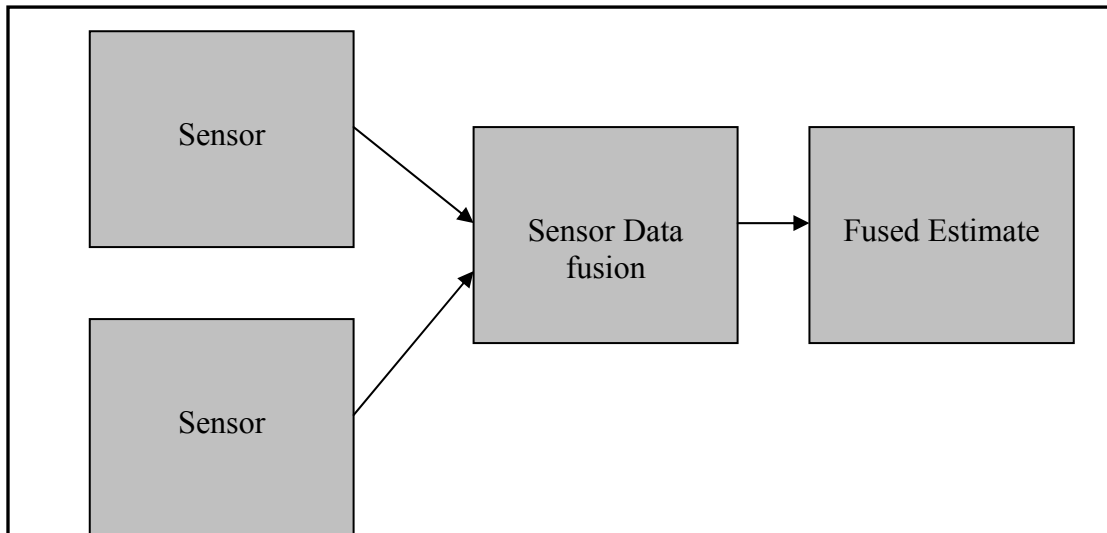


Figure 7: Data fusion type of sensor fusion system

In this dissertation we address the issue of sensor coverage using a hybrid static-mobile sensor network. Using a novel tracking methodology we demonstrate the feasibility of using static sensors integrated with mobile sensor nodes to accurately track dynamic targets over a large period of time. Some of the major challenges addressed in this research include integrating data

from static and mobile sensors, developing tracking algorithms that can process and track targets in near-real time and displaying the dynamic data on a graphical user interface.

The dissertation is organized as follows. The literature review of related works is presented in Chapter 2. In Chapter 3, the design methodology for integrating the mobile and static networks and the tracking algorithm is presented. Chapter 4 gives the highlights of the evaluation and a comparison with other tracking algorithms. Chapter 5 presents the conclusions of the dissertation.

CHAPTER TWO: LITERATURE REVIEW

There are many challenges in deploying wireless sensor networks in a static configuration such as the range of each node, area of coverage, battery life and data collection schemes. Since static sensor networks have been well established over many years for target detection and tracking, current research over the last few years has focused on improving the features of the sensor networks by using new sensor deployment methods, improved signal processing techniques and data fusion methods. Sensor deployment strategy based on path exposure as a measure of the goodness of a deployment was presented as an approach in [10]. It illustrated that the cost of deployment can be minimized by appropriately choosing the number of sensors deployed in each step to achieve the desired detection performance.

Some researchers have considered two approaches to detect the presence of a target: Value fusion and decision fusion. In value fusion a master sensor node gather the received signal Strength Indicator (RSSI) values from all the sensor nodes and compares it to a threshold to determine the presence of an intruder while in decision fusion each individual node does the comparison function. The simulation scheme associated a cost function to the number of sensor deployed for a given area of sensor field and an algorithm was used to minimize the cost function.

Mica2 sensors have been used as wearable badges to track movement of people in an indoor environment. In [11] a static sensor network consisting of Mica2 motes was placed inside a building and used to track entities. Issues such as battery life, multi-path fading and ad-hoc management issues were addressed in this research. In [12] received signal strength indicator

measurement was proposed as a very practical approach to localization of nodes. This method did not require prior information on the location of the nodes.

Dan Li et al have outlined a framework for collaborative signal processing for multiple target detection and classification in a distributed sensor network [13]. Their research highlights some of the key steps involved in tracking intruders such as event detection, target classification, and estimation and prediction of target location. They developed algorithms with emphasis on classification and experimented with real data which provided useful insights into the problems and issues with target tracking in a wireless distributed sensor network.

There has also been considerable research focused on using mobile nodes as sensor networks. While some of the papers have analyzed mobile sensor deployment strategy other researchers have tried to address network issues such as congestion, throughput and delay parameters. Few other papers have also addressed the idea of using mobile nodes to improve network coverage of a static sensor network

In [14] Xiaojiang Du and Fending Lin published a paper that simulated a mobile sensor network that improved the coverage of a static network. They simulated a method in which the sink attempts to improve the coverage of the sensor network by positioning the mobile sensor nodes in places where there is lack of coverage by the static networks. The performance evaluation of this method was tested by measuring and comparing the throughput and delay performance of the network with the static nodes and with introducing the mobile nodes to improve the coverage.

“RaceMote” a mobile node for sensor networks was tested for mobility and its capability to enhance a static wireless network. Using a GUI the user can control the location of the mobile

node such that it can be positioned at different inaccessible locations in an indoor area for data collection [15].

The University of Texas at Arlington researchers formulated methods to combine robot navigation issues such as obstacle avoidance, environment mapping and coverage with sensor network issues such as congestion control, routing and node energy minimization [16]. They developed algorithms for adaptive sampling (AS) for distributed field estimation, potential fields (PF) for communication bandwidth optimization, and a discrete event controller (DEC) for mission planning.

Sensor data fusion is used to improve the detection capability but this adds to the complexity of the sensor network in terms of data collection and computation. Therefore researchers have developed sensor fusion algorithms that attempt to minimize the data flow from the nodes. Yin Zhenyu et al [16] researched a multi-sensor data fusion algorithm based on data reduction technique. Their assumption is that sensor data can be divided into two categories, emergent sensor data that changes rapidly with time such as vibration and usual sensed data like temperature that can be sampled at a slower rate because it does not typically show significant changes over a period of time. They used a “data-fusion” layer that compares the usual sensed data to the previously stored value and forwards the data to the sink only if there is a significant change in the sensed value. This reduces the data flow in the network thereby reducing the power consumption.

Douglas A. Walquist used a sensor data fusion algorithm in a simulated environment that was used for an automatic target recognition system [17]. Using particle filter method they could detect and track targets by using multiple target estimates. Each target estimate, or particle, was

updated using target kinematics and compared to the measured data. These estimates are weighted based upon how closely they reflect the given data. Particles that closely reflect the measured data are given higher weight, while those that differ greatly are given smaller weight. The overall target estimate is calculated as the weighted sum of each particle estimate.

Andrew M. Ladd et al, described a robotics-based location sensing system using wireless ethernet that performed probabilistic inference of position and sensor fusion from noisy signals to determine the location of a wireless Ethernet card using a 802.11b wireless connection [18]. Using a hidden Markov model they try to interpolate the mobile position out of the stream of measurements coming from the inference engine. Their method could accurately locate a mobile with one meter accuracy in a building.

Min Ding ,Terzis, A, I-Jeng Wang and Lucarelli, D published findings on integrating imaging and non-imaging sensors to detect and track targets in a complex urban environment [19]. They used low cost mica mote sensors and infra-red cameras for the imaging sensors. Their simulation used non-imaging sensors for preliminary detection and then the imaging sensors to track the target. This method reduced the power requirements of the sensor systems and improved accuracy.

Konrad Lorincz and Matt Welsh researched the use of Mica2 sensors for a robust tracking system that uses RSSI measurement for estimating the location of a target node. They have presented detailed analysis of the tracking system under varying conditions including node failures, receiver sensitivities and number of nodes available [20]. This work however uses static sensor nodes and does not include the benefits of mobile nodes in the system.

A review and tutorial of radio propagation channels was presented Homayoun Hashemi [21]. Theoretical studies on large scale path losses, spatial and temporal variations of signals due to interference, mean excess delay spread and RMS delay spread were explored. The paper also highlighted the differences between indoor and outdoor channel propagation models.

Jorge Cortez et al, presented a control and coordination scheme for optimal area of coverage for a set of mobile sensor network nodes in autonomous motion [22]. Their approach is closely related to the classic Lloyd algorithm from quantization theory. Their proposed coverage algorithm considers many constraints of mobile sensor nodes such as limited range and non-centralized decision making.

Sonia Martinez's work in [23] studied the optimal placement and motion coordination for mobile sensor networks for 2D and 3D application. The work also investigated the Cramer-Rao Lower Bound (CRLB) that decreases the uncertainty and the covariance of error. One of the drawbacks in this work was it did not account for propagation noise in indoor conditions.

Tatiana Bokareva et al, investigated the use of wireless sensor networks for battle ground surveillance in outdoor conditions. Using Mica2 mote sensors they demonstrated the tracking of multiple targets using data fusion and filtering methods in noisy conditions [24].

Another set of research papers by P. Enge, P. Misra, N. Bulusu, J. Heidemann, and D. Estrin describe use of GPS to locate the sensors. Although these systems work in outdoor conditions they do not address the case of indoor localization problems [25][26].

The review of previous work and the survey of the literature provide ample information on methods and approaches of solving various challenges faced by wireless sensor networks. As can be seen, the majority of the previous and recent works have in general focused on individual

issues such as solving coverage problems, network security, false alarm rates and network integration. They also used homogenous sensor systems that did not entirely reflect real scenarios where it might not be possible to field a homogenous sensor system. Our proposed work tries to fill this technology and application gap by developing a hybrid sensor system that uses static and mobile sensors working over a set of different frequencies and a range of transmit power to effectively track a target in a sensor web, improve coverage and reduce false alarm rates.

CHAPTER THREE: TRACKING USING A HYBRID NETWORK

This chapter outlines the building of three types of sensor networks and finally a hybrid sensor network that combines the functionality of the three networks. The three sensor networks are i) Mica mote static network, ii) Ultra-Wide band static network and iii) Unmanned ground vehicle mobile sensor network. The basic elements of the design of these three networks are presented including the intruder detection and tracking mechanism. The results from the test and evaluation of the networks including the probability of false alarm and the accuracy are also presented. Finally the design of the self-organizing mechanism of the hybrid sensor network is described. The determination of target location at successive time intervals is equivalent to tracking a target in the sensor web. There are many methods for target tracking such as time difference of arrival (TDOA) and beam-forming methods. Sophisticated TDOA methods involves accurate synchronization between the nodes, which require high energy and also sophisticated hardware such as an accurate global clock and beam-forming methods require assumptions such as plane wave propagation which might not be the case when the target is too close to the sensor. A method to overcome these drawbacks through the use of simplistic signal strength measurements from multiple sensors located at known points on a sensor grid is also presented for the location and tracking of a target.

Design of Sensor Network – Static and Mobile Network

Our initial studies that was done using suites of static mica mote sensors, UWB radios and mobile sensor platform as independent systems have yielded promising results for intruder detection and tracking application [27]. Although each of these systems was homogenous we were able to measure the accuracy of these different systems that led us to develop a hybrid sensor network. These networks that were designed for intruder tracking are presented below.

Mica Mote Static Sensor Network

Experiments were conducted with Mica2 mote radios using light sensors for a simple detection system [28]. When an intruder crosses the monitored environment, the system detects the changes in the light intensity. Any change greater than a pre-defined threshold, indicates the presence of an intruder. An integrated web cam was used to take snapshot of the intruder and transmits the pictures through the network to a remote station. This project was done as part of a large sensor web project for the Office of Naval Research to develop long term monitoring systems for border security.

The experiments involved

- Using the motes in an ad-hoc network mode to collect sensor data from environment for the application
- Monitor sensor data from the mote sensors using a Star-gate central computer in wireless mode

- Configure the Star-gate computer to receive sensor data and operate the camera to capture image of the intruder.

Each Mica2 mote sensor node was connected to a MTS 300 sensor board that was used for the experiments. . Figure 8 is a flowchart of software design. The motes send raw messages encoded with the sensor readings that are read through the wireless link. A raw message format is given by

```
FF FF 007D 1D 84010500 A801 F2011602 EF 0138007802290327030000000000000000
```

Each data packet contains several fields of data. The overall message format is as follows:

Destination address: 7D 1D Message handler ID: 84 Group ID: 01 Source address: 0500

Temperature: F201 Light: 1602 Microphone: EF 01 accelX: 3800 accelY: 7802 magX: 2903

magY: 2703

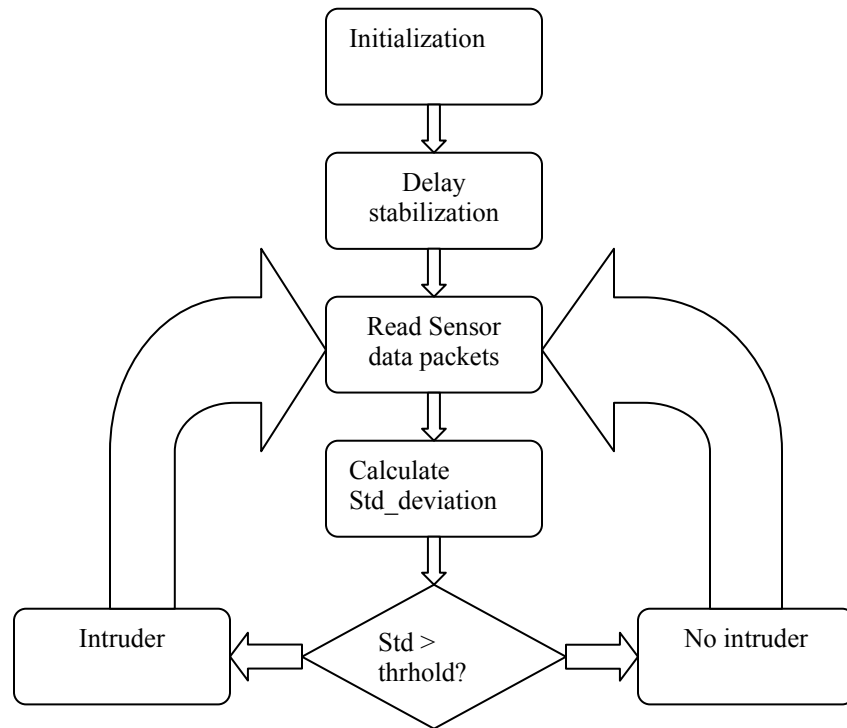


Figure 8: Target detection algorithm using mote sensor network

The MTS 310 mica sensor board has five sensors: accelerometer, magnetometer, microphone, light, and temperature. Only the light data was used since the change of light data was being used in detecting an intruder. The software algorithms reads the in-coming data packet stream and extracts the light sensor values to calculate the standard deviation over a pre-set set of sensor readings. If the light change (standard deviation) exceeds a pre-defined threshold, the code will start to write a value of “1” to the data-file file, meaning that there is an intruder. Otherwise, a “0” is written to indicate an absence of the intruder. After detection is recorded, the system delayed the next set of data collection for 5 seconds for the system to time synchronize and convey the information of the presence of the intruder to the Star-gate computer to record a

picture of the intruder. Using the apache web-server enables users to monitor the system from a remote location. If a user is connected to the internet, they would be able to access the apache web-server on the Star-gate to retrieve the intruder pictures from any location. Figure 9 shows the webcam, sensor nodes and Star gate computer in the intruder detection system.



Figure 9: Sensor node data being collected by the central computer and the webcam

The sensor network setup used for the detection is shown in Figure 10 with the sensors, data collector, local computer and the Star-gate.

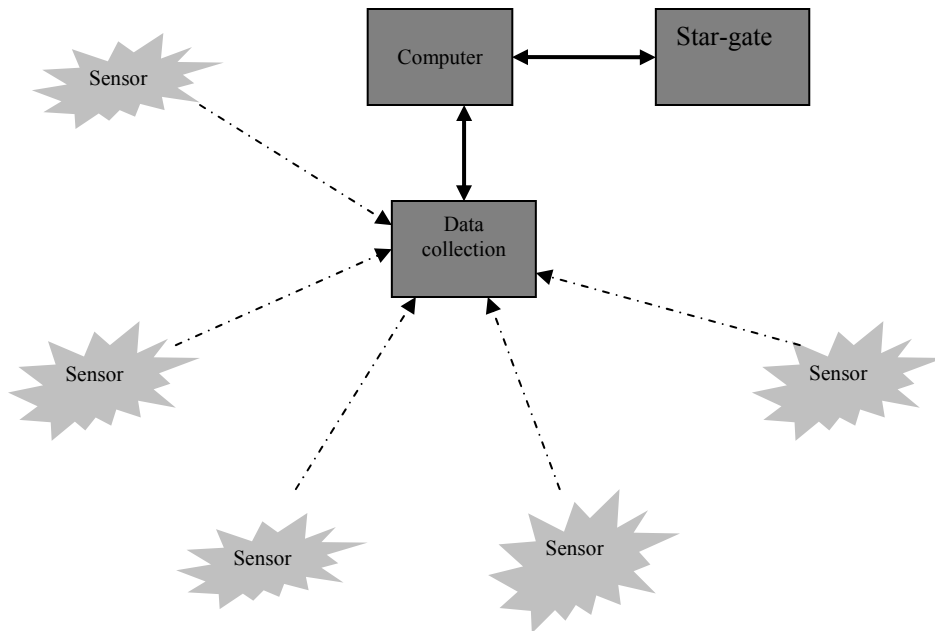


Figure 10: Sensor network configuration

Ultra-Wide Band Static Sensor Web

An Ultra-Wide band (UWB) sensor web system that estimated and tracked a dynamic target was developed to demonstrate the capability of using wireless network for detection and tracking [32]. UWB signals are often defined as having 3-dB bandwidth greater than 25% of their center frequencies of the radiation. The major advantages of UWB is the wide bandwidth which allows high resolution position location and tracking, reduced multi-path due to fine delay resolution and higher penetration through material. Experiments were setup inside an office building using Time Domain P201 PulsON UWB radios. These are low power devices operating over a wide band of 3.2 GHz and a center frequency of 4.7 GHz. The UWB radios were placed at varying distances from each other and the background noise scans were measured. Figure 11 shows the

amplitudes for different samples in a single scan. The large early peak is due to the line-of-sight from the transmitter to the receiver and the later signal oscillations are due to the multi-path reflections. The background noise essentially consists of ground bounce, noise from the various analog components on the electronic board and noise due to various environmental factors. Each scan data contains the received radio signal for 800 range bins with each range bin corresponds to 32ps or 0.032ft. The noise is normally distributed and a threshold is set for detecting the presence of an intruder. The parameters used in the simulation are tabulated in Table 2

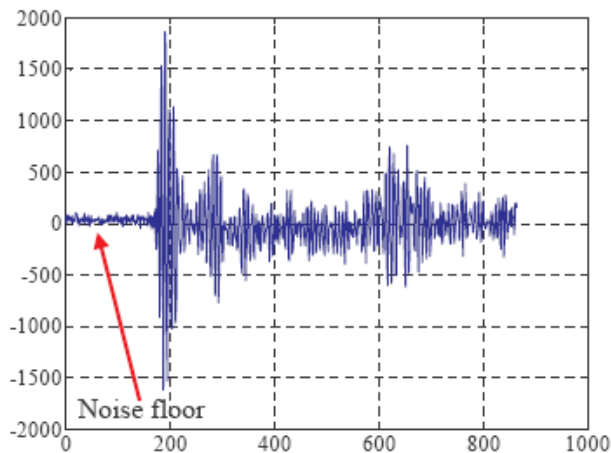


Figure 11: Background scan amplitudes for the range of 800 bins

Table 2: UWB sensor web simulation

Parameters used for Simulation	Values
Coding Level	Binary (1/0)
Carrier Frequency	3.1 GHz
PN sequence length	7
Data Length	100
Fading Channels	Rician, Rayleigh
Receiver type	BPSK
Number of Receivers	3 (1 Master, 2 slaves)
Algorithm Used	Chan's algorithm
Transmitter Location	(6,8)
Receiver Nodes	(1,1) (1,8) and (9,1)
Simulated Location of the Transmitter	(6.2,8.3)
Distance between nodes	100 feet

For purposes of intrusion detection, a threshold is set for each range bin and an alarm is sounded whenever this threshold is exceeded [32]. When an intruder is present, an (true) alarm is sounded if $S+N>t$, where S is the signal, N, the noise, and t the threshold and a (false) alarm is sounded when $N>t$ when no intruder is present. The threshold levels were varied to estimate the number of false alarms triggered for a given threshold. If the threshold is too low the noise will exceed the threshold and trigger false alarms. Figure 12 shows the threshold levels for various values of probability of false alarms.

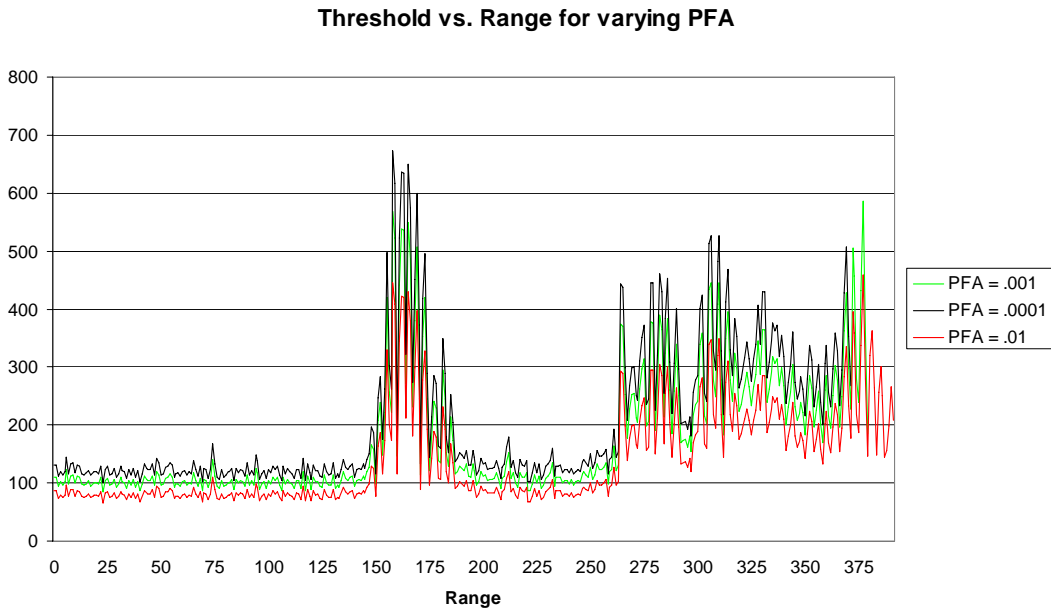


Figure 12: Threshold for varying Probability of False Alarm

Table 3: Number of False Alarms for given PFAs

Probability of False Alarms	Number of False Alarms
.01	93
.001	23
.0001	0

From Table 3 we see that by setting a setting the P_{fa} value to $10e-3$ we could reduce the number of false alarms to zero hence demonstrating the capability of the UWB radio sensor network to accurately track an intruder in a sensor field.

The experiment conducted consisted of setting up UWB radios at 6 feet from each other with one of the UWB devices acting as a transmitter (requestor) and others as receivers (responders).

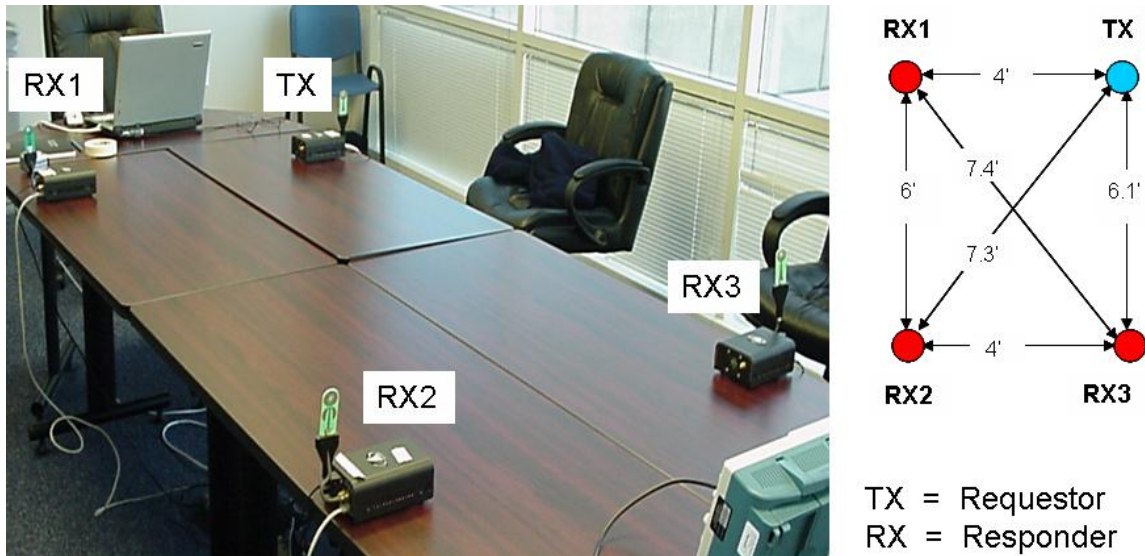


Figure 6: UWB Radio Setup

With the requestor radio assumed to be at $[X, Y] = [0, 0]$, and with a-priori information regarding placement of one more radio, we can easily compute the (X, Y) locations of the radios by triangulation techniques. With two known radio centers fixed, we accurately determine the (X, Y) locations of the remaining radios. The received UWB signals are averaged over a short period of time without any target present to determine ground noise and the system threshold is set based on the background noise. When an intruder crosses the path between the requestor and the responder along any link of the sensor web, an alarm was triggered when the received signal exceeded the noise threshold an alarm is triggered to indicate an intruder.

Sensor Unmanned Ground Vehicle Mobile Sensor Network

A novel method to locate and track entities in the absence of conventional navigation systems like Global Positioning Systems (GPS) that was developed and presented here [29]. The goal of this design was to develop low cost mobile platforms for army tactical engagement simulation (ATES) training exercises that can demonstrate alternate and innovative tracking methods in

urban combat areas [30]. The work also helped us to develop a tracking system that relied on mobile sensors to track a target.

For this network, Unmanned Ground Vehicles (UGVs) with various sensors such as Sonar, GPS, wide-angle cameras and optical encoders were developed and tested. These indoor/outdoor platforms were used for surveillance and reconnaissance of unknown terrain. Using GPS in outdoor conditions and inertial measurement unit for indoors the UGVs we were able to successfully navigate and send back real-time sensor information to the user control station. For processing the sensor data Kalman filtering was used [31].

Figure 13 shows the hardware architecture of the mobile platform and the mobile platform that was used for the experimental work. Figure 14 shows miniature robotic platforms that were also used for the sensor network.

The robotic platform developed was unique as it incorporated the following features.

The platform was designed with Commercial-of-the-Shelf (COTS) hardware and used free source software architectures for command and control;

The system was successfully deployed for unmanned applications and the technique for 3-Dimensional scanned point cloud data collection was demonstrated for indoor environment.

The robotic platform was equipped with LADAR, video camera and sensors for both indoor and outdoor mapping capability. Additional sensors can be easily mounted and integrated on the platform to improve the accuracy of the system.

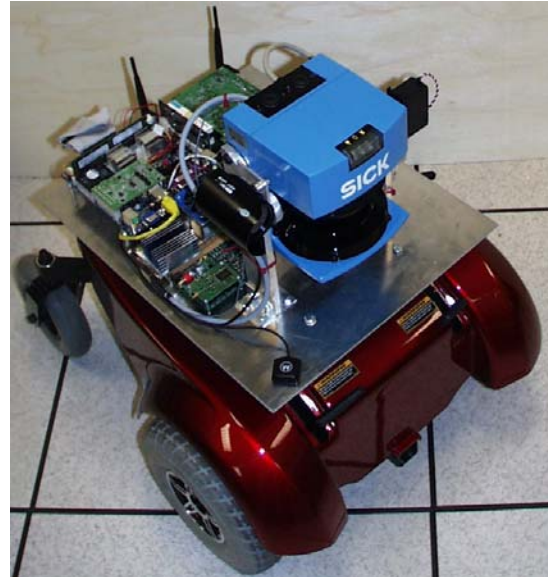
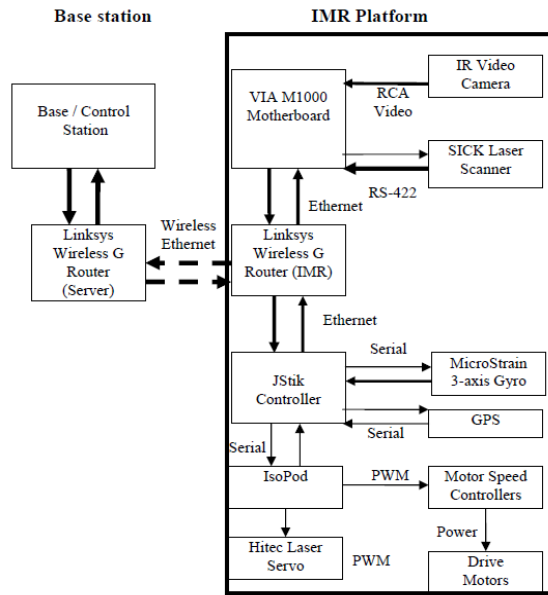


Figure 13: Block diagram of Unmanned Ground Vehicle for reconnaissance and tracking

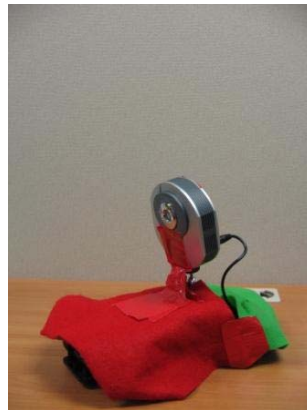


Figure 14: Unmanned ground vehicle used for the sensor network

To ensure constant communication with the mobile platform a mobile ad-hoc routing protocol (MAR) was used to “hop” the sensor and control data between the platforms and the central control station through other UGVs on the field. A scaled model of robotic platforms was used to mount on-board hardware components such as small form factor computer, microprocessors and

sensors. Software routines in Java are used to control and collect data from the sensors through the microprocessor. The on-board computer was used to compress the MPEG2 video signal feed from the camera and transmit it to the central station through a wireless 802.11b wireless router in ad-hoc mode.

When the platform moves from outdoors to indoors the GPS data no longer registers in the system. The software will continuously monitor the GPS data and when it no longer updates it will switch over to using the data from the onboard inertial sensors. The onboard sensor data is then used to extrapolate from the last known GPS position that was recorded by the platform. The onboard inertial measurement unit (IMU) measures the roll, pitch and yaw of the platform. However, due to their error propagation properties an external process, in this case, a Kalman filter was utilized to reduce the IMU errors. In the loosely coupled system used, the GPS receiver had its own Kalman filter algorithm which is used to calculate positions and velocities. GPS-derived positions and velocities are combined with IMU positions and velocities to form the error residuals which are sent to the navigation Kalman filter. This filter corrects the IMU in a feedback manner reducing the effects of biases and drifts. Figure 15 shows the feedback system in operation. The data gathered and the results obtained are presented in the next chapter.

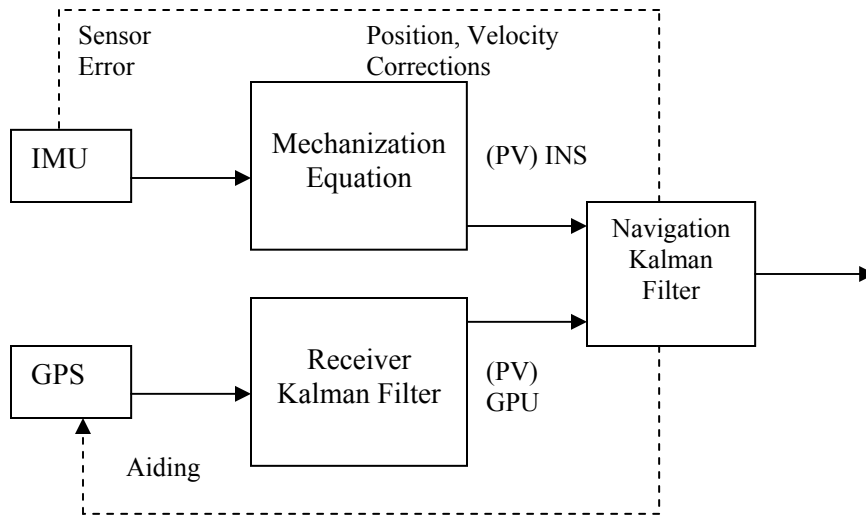


Figure 15: GPU and INS feedback system for a mobile platform

The three sensor system described above were used for location and tracking of a target through the sensor grid and the accuracy of each of the system was tabulated the results of which are presented below.

Accuracy of the Three Tracking Systems

As mentioned before, these static and mobile sensor networks for intruder detection were studied as independent homogenous suite of sensor systems and yielded accurate tracking results for various types of intruders with error margins in the order of a few feet. Table 4 below shows the average distance between sensors on each system and the associated errors measured.

Table 4: Type of sensor network and accuracy of intruder location

Sensor Network Type	Average Distance between sensor nodes	Accuracy of Intruder location
Mica mote sensor	50 feet	+ 4 feet
UWB sensor web	100 feet	+ 0.16645 feet
UGV mobile sensor	100 feet	+ 3.7 feet

A literature review showed no evidence of a heterogeneous sensor network system with static and mobile sensors designed for use in target tracking.

Hybrid Sensor Network

This section describes the design of the hybrid sensor network that combines some of the characteristics of the previous three sensor network designs. One of the main drawbacks of the previous three designs was that they were homogenous in nature, i.e., either static or mobile. Our goal is to develop a hybrid sensor network that combines static and mobile sensor nodes to efficiently track a target.

Self Organizing and Localization by the Hybrid Sensors

The unique feature of this hybrid sensor network is the ability of the nodes to self-organize and automatically form a sensor network. Self-organizing is the process of locating the position or coordinates of the sensor nodes on the sensor network. This is done by each of the sensor nodes ranging to its nearest neighboring nodes to measure the received signal strength value and then estimating its location. This self-localizing process is described in detail in the next chapter.

The other main feature of the hybrid sensor network is the use of mobile nodes to extend the range and coverage of the static network. For a given coverage area when the central node does not get feedback from the network nodes indicating that at least three nodes are tracking the target, the central node will direct an appropriate node to move to a specified geographic location such that at least three nodes are available to track the target [37]. The repositioning of the sensor nodes is needed not only to continuously track a target but also in case of node failure in order to maintain the coverage area of the sensor network. In our design we have assumed that each of the sensor nodes can act as either static or mobile sensor node as needed.

For example, consider a coverage area of 100 ft X 100 ft that needs to be populated with a certain number of sensor nodes to detect and track the presence of an intruder. The central node is programmed to have a precise map of this sensor field in terms of local coordinates X,Y extending from (0,0) on the X-axis and (0,100) on the Y-axis. When the static nodes are distributed in the sensor field in an ad-hoc fashion and report back their local coordinate position based on X and Y the central node forms a virtual map of the nodes. The program verifies if the coordinates of the static nodes are such as to fully cover the prescribed sensor field. For example if a static sensor node at an edge reporting back to the central node is at (80, 80) the central node directs a static node to become mobile and move to the location of (90, 90) and another to (100,100) such that the entire sensor field has coverage.

This type of a self organizing sensor network is very useful to maintain full coverage of a sensor field to handle node failure. The central node will constantly update the virtual map of the sensor field and when it detects pockets of uncovered areas due to failure of nodes or absence of an appropriate static node it can direct a node to become mobile and move to that location and help

it integrate into the local grid and thus maintain full coverage. This process is illustrated in Figure 16, Figure 17 and Figure 18 below.

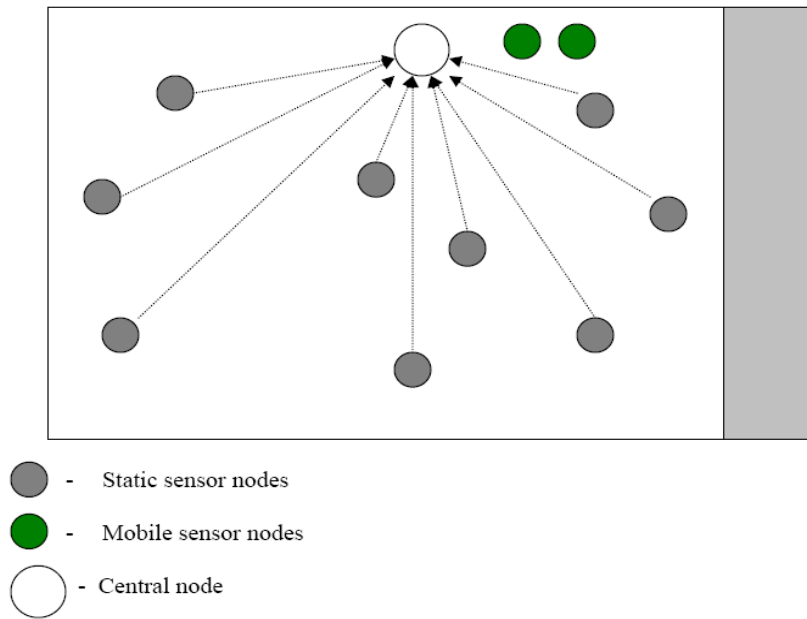
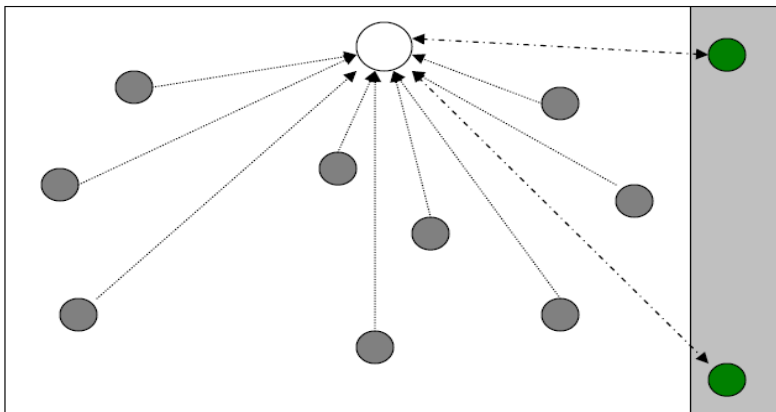
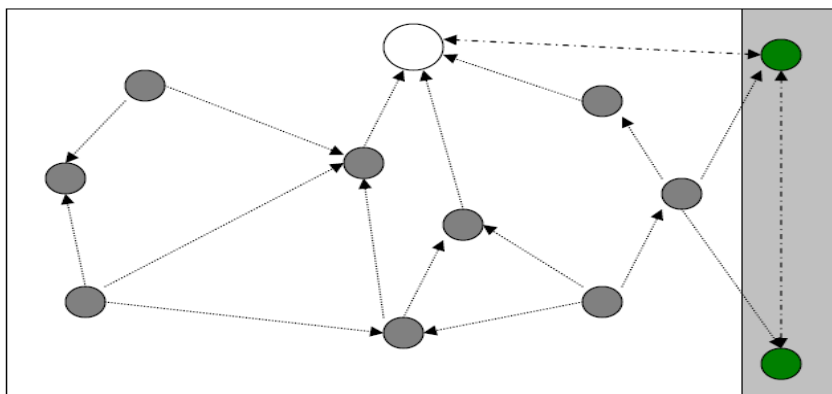


Figure 16: Stage 1: Static nodes report position to central node



- - Static sensor nodes
- - Mobile sensor nodes
- - Central node

Figure 17: Stage 2: Central node dispatches mobile nodes



- - Static sensor nodes
- - Mobile sensor nodes
- - Central node

Figure 18: Stage 3: Central node organizes the static and mobile nodes into a grid

Target Tracking in the Hybrid Sensor Network

The entire network is divided into cells and one cell will comprise of a few nodes one of which will act as the main node. This main node in the cell runs a signal strength detection algorithm to its nearest neighbor nodes. This algorithm samples and averages the signal strength over a short period of time at a fixed rate. In our design the received signal strength indicator (RSSI) measurement for the Mica2 mote sensors will be used for the static network. For the mobile network, the sensors on the mobile platform will use a signal strength algorithm to locate and track the entity. When a target enters the cell the main cell node will detect a change in the RSSI due to interference and signal reflection from the target. Based on comparison to a background noise threshold the main cell node will indicate the presence of the intruder inside the cell. When the target moves to the next cell, this procedure is repeated in the next cell. Using this method the sensor network can track a single target as it traverses inside the sensor network. Figure 19 shows the layout of a hybrid sensor network that combines static sensor nodes consisting of Mica2 motes and mobile sensor nodes consisting of UGVs. Figure 20 illustrates the detection and tracking of a single target through the sensor network. If multiple targets are sufficiently separated in space and time it can essentially use the above procedure to track multiple targets. Sufficient separation in space and time implies that each target is indicated by a distinct RSSI change in the sensor readings.

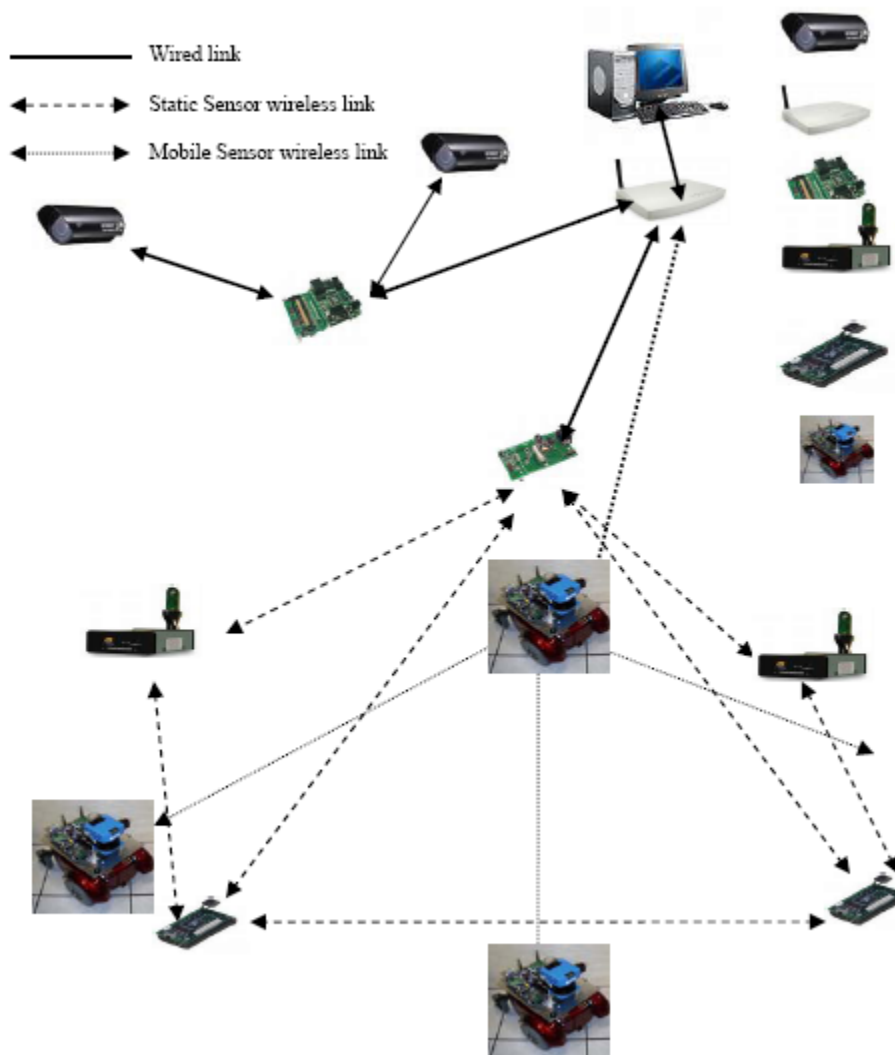


Figure 19 A hybrid static mobile intruder detection sensor network

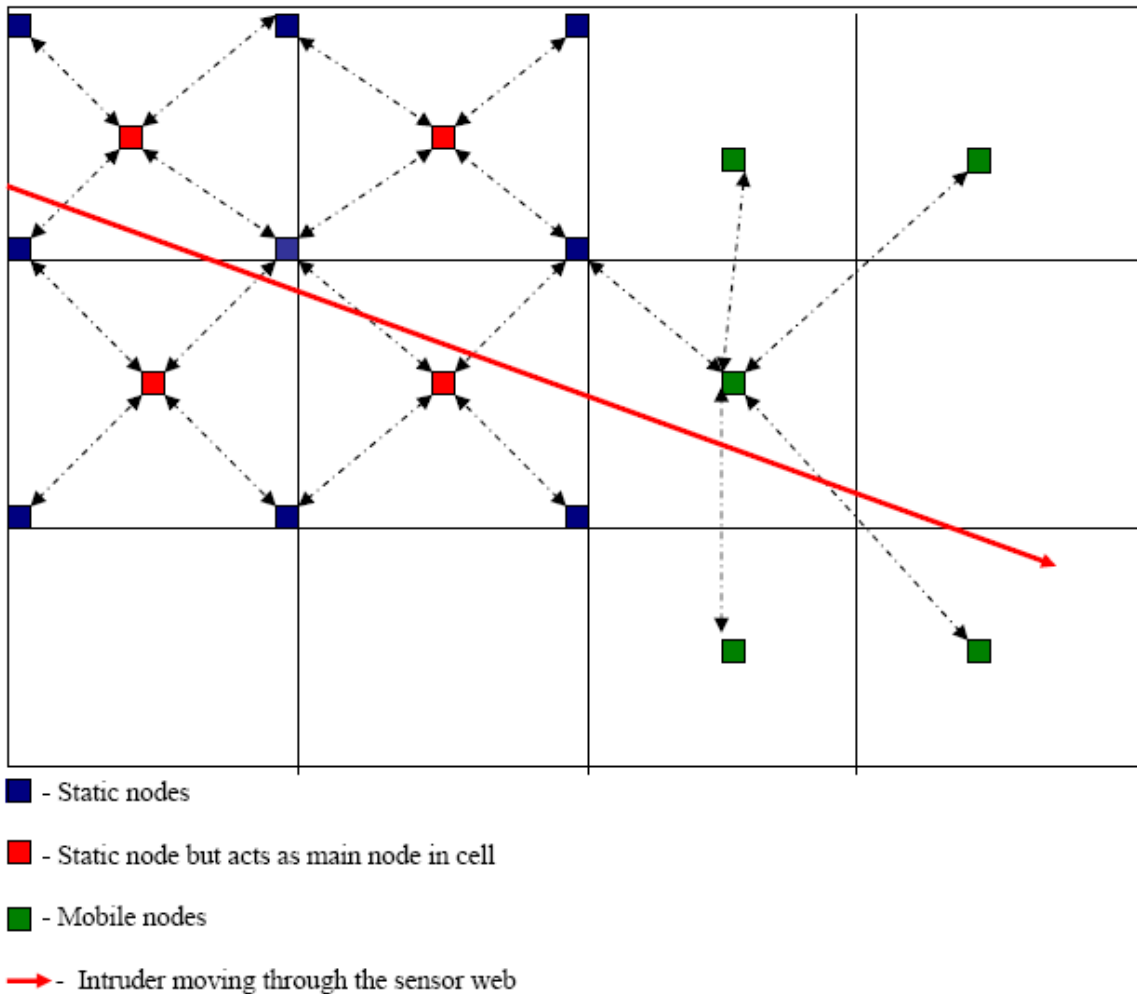


Figure 20: Target tracking through the sensor web

Constant False Alarm Rate

Setting a constant threshold for the main cell node for target detection is not be feasible as there can changes in the background noise conditions. Therefore, we modeled the noise as a Gaussian random variable and the threshold was dynamically adjusted according to the noise variance of detector output so that the detector maintains a constant false alarm rate (CFAR). This technique of using a varying threshold to maintain a constant false alarm rate is called CFAR processing.

There are many types of CFAR techniques and we chose to use the Cell averaged CFAR processing technique. Cell averaged CFAR (CA-CFAR) is an adaptive process by which the threshold is estimated by measuring the mean level of the signal in a set of neighboring sensor nodes. The CA-CFAR processor is the optimum CFAR processor (maximizes detection probability) in a homogeneous background when the sensor nodes contain independent and identically distributed (IID) observations governed by Gaussian or exponential distribution. The basic idea is to average the signal to noise ratio in a set of neighboring sensor nodes and use the averaged value to set the threshold for the given sensor node detector. The CA-CFAR process will be explained in detail in the next chapter.

Data Fusion using Kalman Filtering

The sensor data gathered will be corrupted by noise due to different factors such as electronic noise and environmental conditions. If this noise is not filtered from the data it will have a cumulative effect and degrade the accuracy of the measurements. Thus a hybrid Kalman filter based filtering approach was used to process the data obtained from different sensor nodes in the network. Figure 21 shows the data collected from the static and mobile sensors being processed through the Kalman filter.

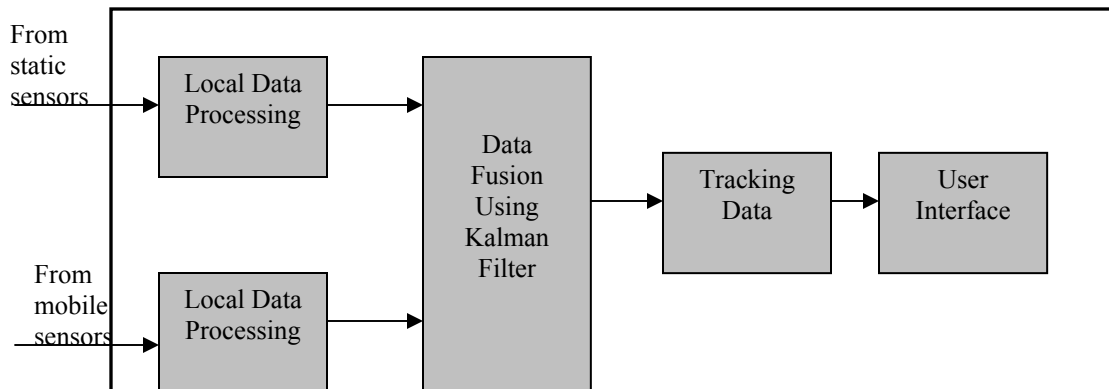


Figure 21 Sensor data fusion using Kalman Filter

The Kalman filter is a system that predicts or estimates a future state. It is basically a set of algorithmic equations that predict a future state then corrects the actual state based on the error between the predicted state and the actual state [33]. This filter was used to correct the sensor readings that are used to track the dynamic target. The Kalman filter will work in 2 states: time update (predict) and the measurement update (correct). The term update is used because the filter is a feedback control, thus the filter is constantly updating information to reduce the error between future estimations and present values. The time update equations are considered the predictor equations because it is responsible for estimating future states so as to anticipate estimates of the next step called priori estimates. The measurement update equations are considered the corrector equations because they are responsible for obtaining the current states, and respond to errors between the estimated future state and the measured actual state. They return new values to the priori estimates to obtain a new and improved current state or posteriori estimate of the position of the target. The Kalman filter also applies a correction factor to its estimate in order to account for the drift error of the inertial system over a period of time.

Thus the Kalman filter stage as shown in Figure 21 will combine the sensor data from the static and mobile sensors to remove the background noise from the incoming sensor data and accurately estimate the location of the moving target.

Simulation and Testing

Simulation studies of this self organizing hybrid sensor network were performed using Matlab to prove its efficacy to accurately track a dynamic target. The sensor nodes were simulated on a 2 dimensional grid and a target moving on the horizontal axis was tracked. The simulation also demonstrated the sensor nodes losing track of the target and the reacquiring the target once a mobile node is appropriately relocated to continue the tracking. Field experiments were done on the hybrid network to verify the simulation results. Mica mote sensors with a range of 50-100 feet were used to collect signal strength data up to a distance of 100 feet in outdoor conditions. Remote controlled UGVs were used to carry mote sensors and take user commands from the central node and move to a specific location on the local coordinate map. To quantify the accuracy and real-time capability of this hybrid sensor network we measure parameters such as signal-to-noise ratio as a function of average distance between the sensors, percentage area of sensor coverage and accuracy of target were measured. The results of the simulation and measurements are presented in the next chapter.

CHAPTER FOUR: RESULTS

This Chapter describes the experimental and simulation results to demonstrate that by using a hybrid sensor network the coverage of the sensor grid can be extended over a large area and reduce many of the typical problems associated with static sensor networks discussed in previous chapters.

Our work focuses on three areas of interest in sensor network systems

1. Developing a localization algorithm starting with anchor nodes at known locations such that the location of all the other sensors in the network can be determined
2. Target tracking with static nodes
3. Using hybrid sensor nodes i.e. mobilizing nodes as needed to extend the coverage of the sensor grid

Our simulation and experimental work uses the Mica2 Mote sensors as described in the previous sections. The sensor operates at the 916 MHz frequency band with a range depending on the operating environment, typically 50 feet in indoor conditions.

Localization of Reference Nodes

Our algorithm uses the inherent capability of sensor nodes to measure simple connectivity metrics such as received signal strength and range estimation to its nearest neighbor. Initially we introduce three sensor nodes on the sensor grid at known locations. These are termed as anchor nodes. The anchor nodes are introduced at known locations in order to aid other nodes to

determine their position in the grid. In order to determine the 2D position we need three known points on a grid. Our test-bed initially consists of 3 sensors with known locations that will serve as anchor nodes. The location of the anchor nodes can be given by $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3)$. A node is then introduced into the sensor grid and will use the anchor nodes to localize itself by estimating its centroid location in reference to the anchor nodes. This technique is very effective when the all the anchor nodes are visible to the sensor node. This new fourth node is now added to the list of known node locations and all of these can be used for localization of all other nodes in the grid using any three at a time. Once the fourth node is introduced into the sensor grid we refer to all the nodes as reference nodes since any 3 can be used for localization. Our experimental and simulation set-up consists of 6 reference nodes that will track a target through the sensor grid. We define a few terms that will be used to demonstrate our localization algorithm technique

r – Separation between the reference nodes in Euclidean distance metric

R – Transmission range of each anchor and reference node

$N_{sent}(I, t)$ = Number of beacon signals sent by each node in time t

$N_{recv}(I, t)$ = Number of beacon signals received by each node in time t .

(X_{est}, Y_{est}) = Estimated location of the reference node

(X_{act}, Y_{act}) = Actual location of the reference node

The equation below shows the formula used by the node to localize its position based on the reference nodes

$$(X_{est}, Y_{est}) = \left(\frac{X_i + X_k}{k}, \frac{Y_i + Y_k}{k} \right) \quad (5)$$

The estimated location of the sensor node is the average of the k sensor reference nodes in the X and Y axes. This formula is a simple and robust method for determining the estimated location of the sensor nodes.

Connectivity Metric

Each reference node transmits 100 byte data packets every 125 msec. The newly introduced nodes collect these signals from the reference nodes and estimate their location by calculating the distance to the three reference nodes in range. We define a term called connectivity metric (CM) to characterize the signal level received by the nodes in the sensor grid. The connectivity metric (CM) is defined in terms of the ratio of the signal received by the nodes to the signals sent.

$$CM = \frac{N_{recv}(i, t)}{N_{sent}(i, t)} \quad (6)$$

Where $N_{recv}(i, t)$ is the number of packets received at the node, $N_{sent}(i, t)$ the number of packets sent and t is the time period. The connectivity metric is a user set value in the simulation such that the reference nodes will only utilize the signals from the anchor nodes for values that exceed the connectivity metric (example, 90%). For our experiments we chose the connectivity metric to be at 90% which implies that 90% of the packets sent by the transmitter should be received by the receiver to be considered as a successful connection. We decided on this measure of

connectivity metric based on the bit error rate values for a BPSK modulated system in a fading channel environment. Figure 22 shows the BER as a function of various signal-to-noise ratio values.

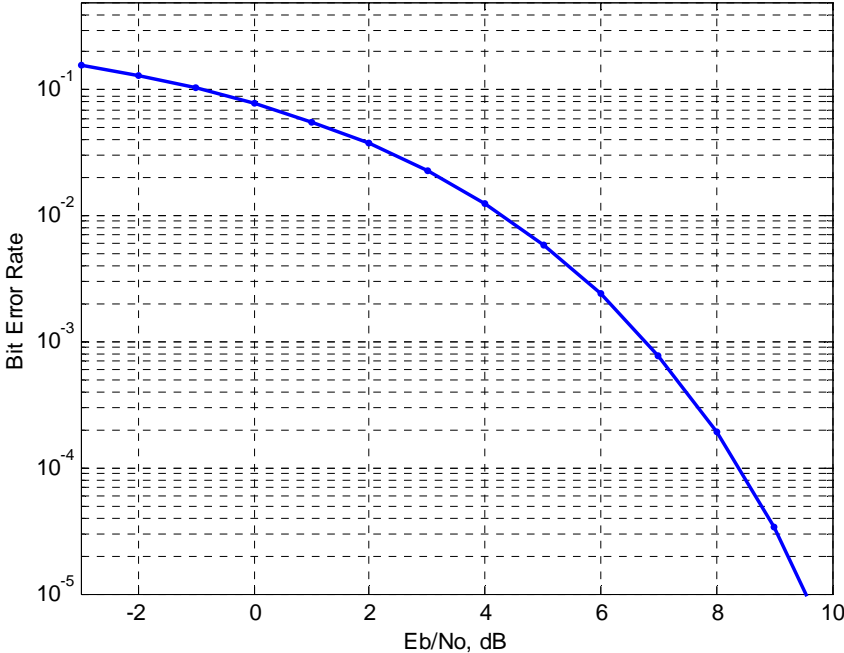


Figure 22: BER for BPSK Channel with Rayleigh and Rician fading

Received Signal Strength Model

From each of the data packets received by the node it can calculate the Received Signal Strength Indicator (RSSI) to the nearest three reference nodes. In the Mica 2 mote model that are used in our experiments and simulation the RSSI is provided at the analog to digital convertor output as a 10 bit number. The following equation gives the conversion of the ADC counter value to RSSI value. The values are defined as

V_{RSSI} = RSSI Voltage measured from the RSSI pin of the CC1000 chip of the Mica 2 sensor

$V_{Battery}$ = Mote voltage battery

ADC_COUNTS = 10 bit ADC count at the micro-controller

K = Signal strength constant

r = transmission distance

$$V_{RSSI} = V_{battery} \times ADC_COUNTS / 1024 \quad (7)$$

$$RSSI(dBm) \equiv -51.3 \times V_{RSSI} - 49.2 \quad (8)$$

$$Signal\ strength(dBm) \equiv x \equiv 10 \times \log_{10} \left(\frac{Power(mW)}{1(mW)} \right) \quad (9)$$

Assuming free space wave propagation and an omni-directional antenna radiation pattern we can conclude that the total power radiated by the antenna remains a constant although the surface area increases by a factor of r^2 . Therefore for signal strength of K constant, the absolute power decreases in the ratio of K/r^2 . Therefore the signal strength x can be defined as

$$x \equiv 10 \times \log_{10} \left(\frac{K}{r^2} \right) \quad (10)$$

$$x \equiv 10 \times \log_{10}(K) - 20 \log_{10}(r) \quad (11)$$

$$x \equiv (-51.3 \times V_{RSSI}) - 49.2 \quad (12)$$

Using the definition above $r \equiv 10^p$ where

$$p \equiv \frac{(10 \log_{10} K + 51.3(V_{battery} \times ADC_COUNTS / 1024) + 49.2)}{20} \quad (13)$$

To localize the position of the nodes in the sensor grid the anchor nodes send periodic broadcasts of HELLO messages every 125 millisecond and each message has a packet size of 100 bytes. The message payload consists of four major fields, including the source mote id, the sample counter of 2 bytes, the ADC channel number and the message fields are of 2 bytes each. This 100

byte message length can be varied by the user using the software interface. The data is sent using the little-endian format. Table 5 below shows the message format in the Mica2 mote sensor used in our simulation and experiments [38] .

Table 5: The message format of the Mote sensor

Destination Address	Source Address	Counter	Channel #	Message
2 bytes	2 bytes	2 bytes	2 bytes	92 bytes

The three anchor nodes are placed at (1,5), (10,5) and (25,5) on a 35 x 35 meter sensor grid area for our simulation setup as shown in Figure 23. We limit our grid area to a 35 x 35 meter setup in order to make the measurements easier and also due to the limited number of sensors available. These anchor nodes will serve as initial references for subsequent nodes for localization as discussed. Figure 24 shows three nodes added at (1, 20), (10,30) and (25,25) that form the sensor grid. These nodes used the information of the initial three anchor nodes to fix their position on the grid. All these nodes now serve as reference nodes and any of one of the can be made mobile to track a target. The various parameters used in the simulation model are shown in Table 6.

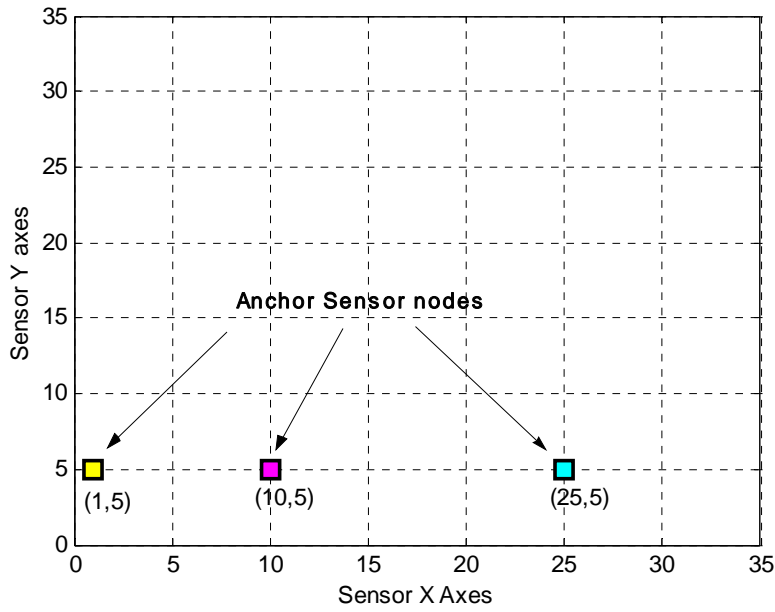


Figure 23: Location of initial anchor nodes on the sensor grid

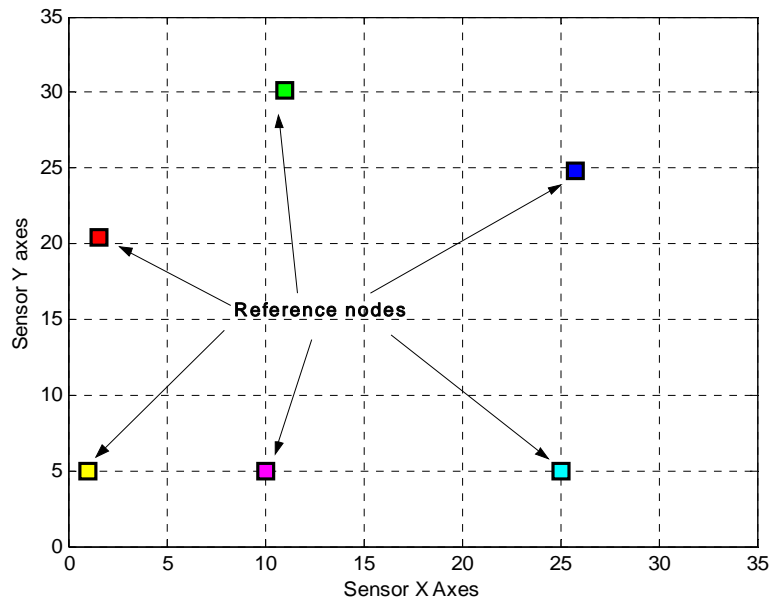


Figure 24: Reference nodes that form the sensor grid

Table 6: Various Parameters used in Simulation Model

Parameter	Value
Path loss coefficient	$\beta = 2.7$
Variance	$\sigma = 16$ dB
Frequency	916 MHz
Antenna	Omni-directional
Packet size	100 bytes
Range	15 meters
Power	0 dBm

The localization algorithm can thus be summarized as:

Step 1: Sensor nodes are randomly spread in the sensing environment with three known sensor node locations that act as anchor nodes

Step 2: Any node in the range of the anchor nodes will use the centroid location of the anchor nodes to localize its position. We know have references nodes in the sensor grid.

Step 3: All reference nodes can be used for further localization and eventually form the sensor grid.

Simulation experiments using Matlab and associated wireless libraries were conducted to verify and validate the localization algorithm. Using the motes we measured the RSSI signal strength and calculated the actual location of the motes on the experimental grid. Consider three reference nodes transmitting signals from position $x = 0, y = 0$, $x = X_a, y = 0$ and $x = 0, y = Y_a$ on a sensor grid area. The received power at the nodes is give by $P_{r_j} = kd_j^{-\alpha}$ where $k=2$ is a constant that refers to the carrier frequency of the signal, d is the distance between the sensors and $\alpha=2$ is the attenuation exponent for an indoor channel. A literature review of indoor communication channels shows that for the frequency range used in our simulation and experimental studies the RMS delay spread is measured at 2.5. To simply our experimentation we placed a Mica2 mote in

another corner of the room at $x = X_a, y = Y_a$ and measured the connectivity metric and RSSI at 1 meter intervals over a $X_a = 35 \times Y_a = 35$ meter grid of area. Figure 25 shows the location of the anchor nodes and the reference node. The localization error (LE) for the location of the nodes is given by using the equation

$$LE = \sqrt{(X_{est} - X_a)^2 + (Y_{est} - Y_a)^2} \quad (14)$$

Where (X_{est}, Y_{est}) and (X_{act}, Y_{act}) are the estimated and actual location of the reference nodes.

The localization error values are then incorporated into the simulation model to correct for errors of the location algorithm.

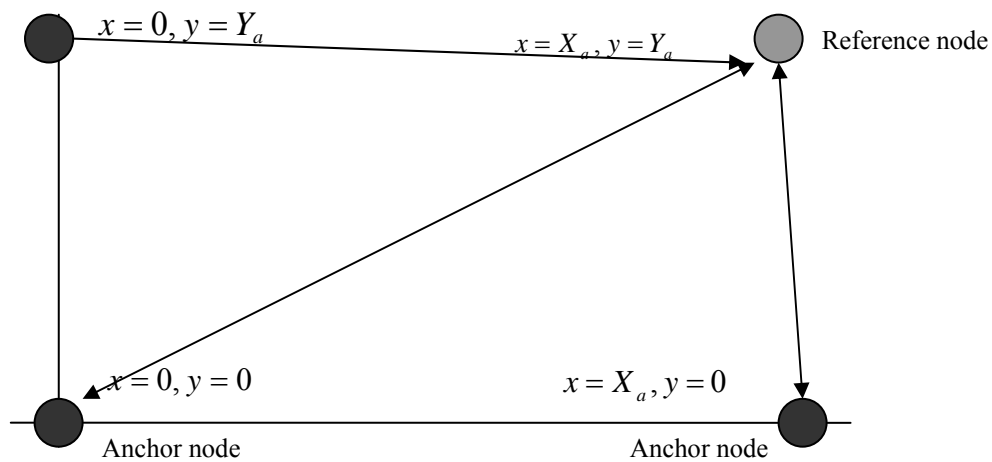


Figure 25: Sensor location estimation

Our focus is on the two issues that make localization difficult to achieve in an indoor environment. The first is the effect of Rayleigh fading channels which is the variation of the received power as a function of time due to multi-path interference. The second is the time taken for the mobile reference nodes to relocate when they are attempting to expand the coverage of

the sensor region. The received power is given by the relation: $P_{r_j} = kd_j^{-\alpha}$ but we average the power received by the sensors over a specified period of time to increase the accuracy of estimation. The average received power from the j -th reference node for the i -th sample is given by

$$\bar{P} = \frac{1}{w} \sum_{h=0}^{w-1} P_{r_j}(i-h). \quad (15)$$

If k and α are known to each sensor the distance metric can be calculated from the received power by using

$$d_j(i) = \left(\frac{\bar{P}_{r_j}(i)}{k} \right)^{-1/\alpha} \quad (16)$$

Consider three anchor nodes B1 at $(x=0, y=0)$, B2 at $(x=X_a, y=0)$ and B3 at $(x=X_a, y=Y_a)$ with radius of signals: d_1, d_2 and d_3 . The three anchor nodes can be detected by the fourth sensor at $x(i), y(i)$ by using the following relation.

$$\begin{cases} x^2(i) + y^2(i) = d_1^2(i) \\ (x(i) - X_a)^2 + y^2(i) = d_2^2(i) \\ 0 \leq x(i) \leq X_a \\ 0 \leq y(i) \leq Y_a \end{cases} \quad (17)$$

$$x(i) = \frac{d_1^2(i) - d_2^2 + X_a^2}{2X_a} \quad (18)$$

$$y(i) = \sqrt{d_1^2(i) - x^2(i)}$$

The errors in the location estimation due to fading and motion can then be calculated by

$$\varepsilon(i) = \sqrt{[x(i) - \tilde{x}(i)]^2 + [y(i) - \tilde{y}(i)]^2} \quad (19)$$

where $x(i), y(i)$ and $\tilde{x}(i), \tilde{y}(i)$ are the estimated and actual sensor location respectively.

Target Tracking with Static Nodes

Once the reference nodes have been located on the sensor grid as shown in Figure 24 the grid can be used to track a target moving through it in real-time. When a target intrudes into the sensor grid three or more of the reference nodes will be used to track its location in the grid. The target tracking algorithm is described in a series of steps that will highlight the essential process in the algorithm.

Step 1: The target node to be tracked is preloaded with the location information of all the reference sensor nodes and the localization error values from a central base station as a database file

Step 2: The target node begins movement in the sensor node and receives connectivity from the reference nodes. It calculates the connectivity metric (CM) to determine the closet set of reference nodes and compares the values to the stored database of reference node locations using Euclidean distance metric.

Step 3: The target node calculates the centroid of the Euclidean values to locate its position in the sensor grid. It then corrects for the location from the localization error database to be able to locate itself in the grid.

There are many methods to track a target in a sensor grid such as Time of Arrival, Angle of Arrival and Time Difference of Arrival. Our method is uniquely different from existing tracking methods in that it uses a simple received signal strength indicator in order to improve the probability of tracking without increasing the latency of the tracking systems. Before the target node is introduced into the sensor grid it is pre-loaded with the location information of the reference nodes and the localization error information as determined from previous simulation.

The next step is for the tracked target node to gather signals from the reference nodes. In order to be tracked the target node should be in the range of three or more reference nodes. We use the same RSSI measure that was used for localizing the reference nodes for tracking the target. To compute the distance from the measured RSSI to each of the reference RSSI that is in range of the target node we use the Euclidean distance metric

$$E(r, s) = |meanRSSI(t)_r - meanRSSI(t)_s| \quad (20)$$

where $E(r, s)$ is the set of Euclidean distances from the measured RSSI to the reference RSSI, $meanRSSI(t)_r$ is the mean value of the measured *RSSI* from each of the reference nodes and $meanRSSI(t)_s$ is the mean *RSSI* of the reference nodes stored in the database of the target node.

The target node then compares the values to the user-defined Connectivity metric ratio. In our case we have set the CM ratio to 90% as was specified earlier, which implies that the target node needs 90% of the packets to be received in order to use that reference node for position location estimation. Figure 26 shows the essential steps in the target localization algorithm.

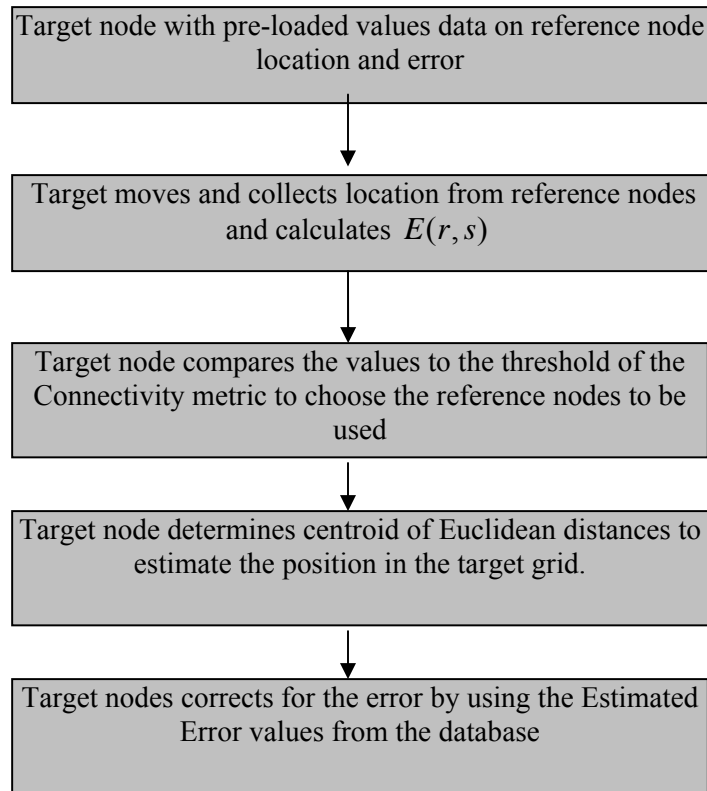


Figure 26: Steps involved in tracking of target node

Detection in the Presence of Noise

In any detection system there is noise that is present along with the received signal. Our goal is to reduce the effect of the noise and therefore improve the probability of detection. This section describes some of the noise characteristics that are taken into consideration in our simulation and experimental set-up.

An additive zero mean white Gaussian noise $\mu(t)$ with variance ψ is added to the input signal to the nodes is composed of the signal $n(t)$. The noise is assumed to be spatially incoherent and uncorrelated with the signal. The effect of fading effects as the signal propagates from the transmitter to the receiver is also considered.

The different noise functions that we have used in our simulation model are described in the following sections.

Fading Channels

When signal propagation takes place close to the ground surface, free space signal propagation models are not enough to describe and predict the channel characteristics. The signal from the transmitter to the receiver will follow multiple reflective paths that will cause destructive interference with the received signal's amplitude, phase and arrival time. These phenomena are primarily modeled by the Rayleigh or Rician fading channel models. All our analysis are done in indoor conditions and the time varying indoor radio propagation channel can be modeled using linear time-varying filter with impulse response given by

$$h(t, \tau) = \sum_{k=0}^{N(\tau)-1} a_k(t) \delta[\tau - \tau_k(t)] e^{j\theta_k(t)} \quad (21)$$

where t and τ are the observation and application time of the impulse respectively $N(\tau)$ is the number of multi-path components, $a_k(t)$, $\tau_k(t)$, $\theta_k(t)$ are the amplitude, arrival-time and phase sequences respectively. A channel model can be used to obtain the response of the channel to any transmitted signal $s(t)$ by convoluting it with $h(t)$ and adding noise. The output of the $y(t)$ of the channel to the transmitted signal can therefore be given by

$$y(t) = \int_{-\infty}^{\infty} s(\tau) h(t - \tau) d\tau + n(t) \quad (22)$$

where $n(t)$ is the low-pass complex value added Gaussian noise [39].

Rayleigh Fading Effect

When there are multiple received signals at the receiver without a line-of-light component the multiple reflective paths can be described by the Rayleigh fading effect [41]. The probability density function of the Rayleigh distribution is given by

$$P(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{\sigma^2}\right) \quad r \geq 0 \quad (23)$$

where σ is the Rayleigh parameter. The mean and variance of the distribution is given by $\sqrt{\pi/2}\sigma$ and $\{2 - \pi/2\}\sigma^2$ respectively. Figure 27 shows the received signal over time in a Rayleigh fading channel at 915 MHz using the Rayleigh probability density function equation. The simulation assumes that there are 100 scatterers in the environment with the multi-paths arriving at the receiver at the same time without delays (no delay spread).

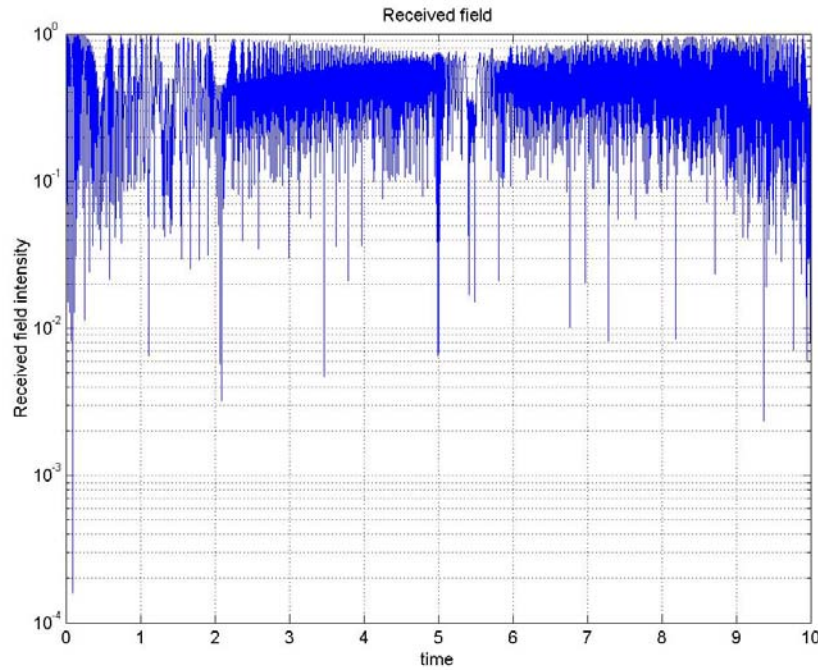


Figure 27: Simulated Rayleigh fading channel at 915 MHz

Rician Fading Effect

In an indoor propagation case there is always a dominant line of sight path between the transmitter and the receiver and several multi-path signals that tend to add or cancel this dominant path signal. When there is line of sight the noise model is given by the Rician fading model probability density function equation that is given by

$$f(r) = \frac{r}{\psi^2} I_0\left(\frac{rA}{\psi^2}\right) \exp\left(-\frac{r^2 + A^2}{2\psi^2}\right) \quad (24)$$

where

$f(r)$ = Rician fading model

r = envelope of the signal

ψ^2 = variance of noise

I_0 = Inphase component of signal

Figure 28 shows the cumulative probability density function of a Rician fading channel model as a function of x with mean $\mu = 0$ and standard deviation $\psi = 1.5$. These values were chosen by studying various indoor propagation models.

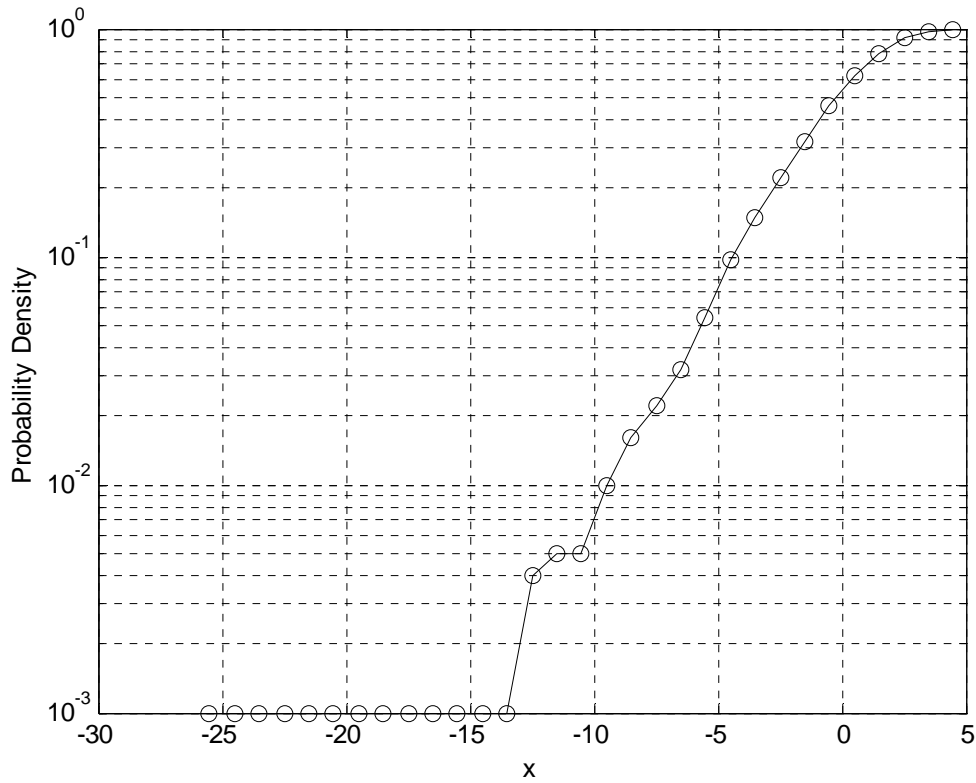


Figure 28: Simulated Rician fading channel probability function

Gaussian Probability Density Function

Noise is usually defined in terms of the Gaussian probability density function given by the equation below

$$f(r) = \frac{1}{\sqrt{2\pi}\psi} \exp\left(-\frac{(r - A)^2}{2\psi^2}\right) \quad (25)$$

where A and ψ are the same as for the Rician fading model. Figure 29 shows the Gaussian probability density function values for $\mu = 0$ and standard deviation $\psi = 1.5$

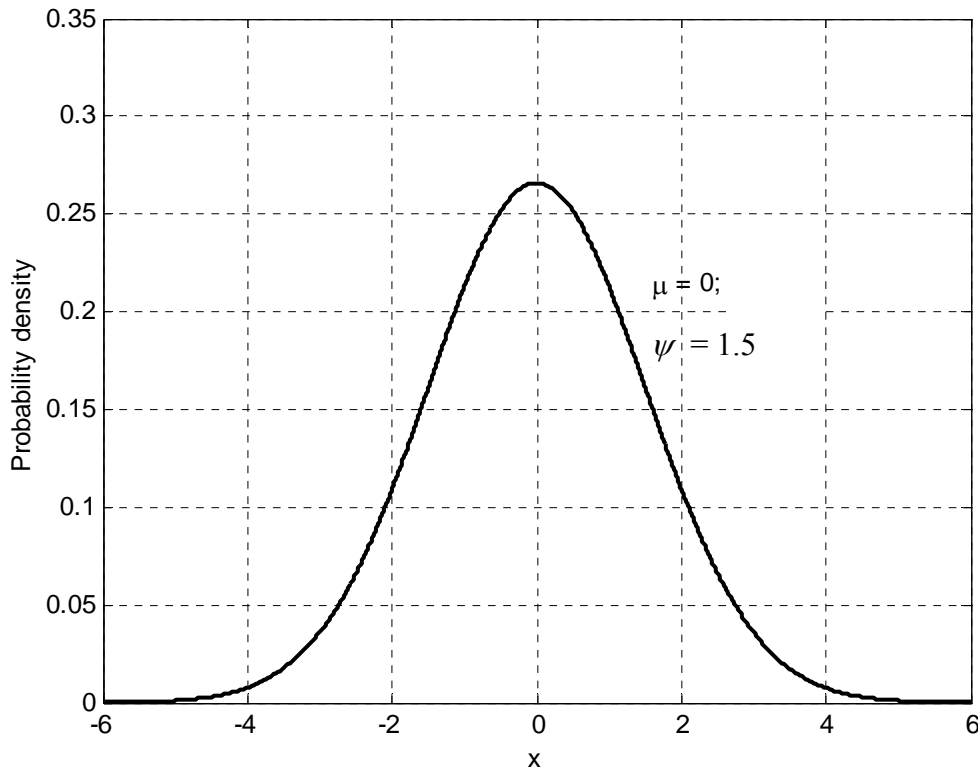


Figure 29: Gaussian probability density function

Other Channel Related Issues

There will be temporal variations of the channel in indoor environments due to the non-stationary nature of the system. This is a very difficult phenomenon to measure and quantify as the temporal variations can be due to many factors that are not yet clearly understood by researchers. Therefore most propagation measurement models assume stationary data while collecting impulse response measurements to predict the channel characteristics. Another channel characteristic that is defined for indoor channel is the Mean Excess Delay and RMS Delay spread. The RMS delay spread of a channel model is given by the equation

$$\tau_{rms} = \left\{ \left[\sum_k (t_k - \bar{m} - t_A)^2 a_k^2 \right] / \sum_k a_k^2 \right\}^{1/2} \quad (26)$$

Where t_A is the arrival time for the first path and τ_m is the mean excess delay defined by

$$\tau_m = \left[\sum_k (t_k - t_A) a_k^2 \right] / \left[\sum_k a_k^2 \right] \quad (27)$$

and t_k is the arrival time for the k signal. The mean excess delay and RMS spread have been measured for number of indoor propagation models. For the small and medium offices similar to our experimental and simulation area used values range from 20 to 50 nanoseconds. Since accurately measuring or quantifying the RMS delay spread is out of scope of this research effort we have assumed the value of the RMS delay spread to be 50 nsec.

Cramer Rao Lower Bound on estimation of sensor node location

The estimates of our sensor node locations are error-prone due to interference and noise from the indoor channel. We have defined our accuracy in terms of the Cramer-Rao lower bound (CRLB).

In a general case if \hat{x}_t (estimated location) is an unbiased estimator for the state variable x_t (actual location) based on $z(1:t)$ (measurements), then the covariance matrix of x_t denoted

by P_t has a lower bound defined by [40]

$$P_t \equiv E\{(\hat{x}_t - x_t)(\hat{x}_t - x_t)^T\} \geq J_t^{-1} \quad (28)$$

where J_t^{-1} is the CRLB for an unbiased estimator. In our simulation and experimental analysis the CRLB is used to determine the accuracy in our system. The CRLB is dependant on the distance and geometry of the sensor grid.

$$CRLB_{RSSI} = \frac{1}{b} \frac{\sum_{i=2}^N d_{1,j}^{-2}}{\sum_{i=2}^{N-1} \sum_{j=i+1}^N \left(\frac{d_1 d_{i,j}}{d_{1,i}^2 d_{1,j}^2} \right)} \text{ Where } b = \frac{10n_p}{\sigma_{RSSI} \ln(10)} \quad (29)$$

n_p = Path loss coefficient

σ_{RSSI} = standard deviation of RSSI measurement

d = distance between nodes

The effect of the CRLB in our tracking system is summarized in the Figure 30 shown below.

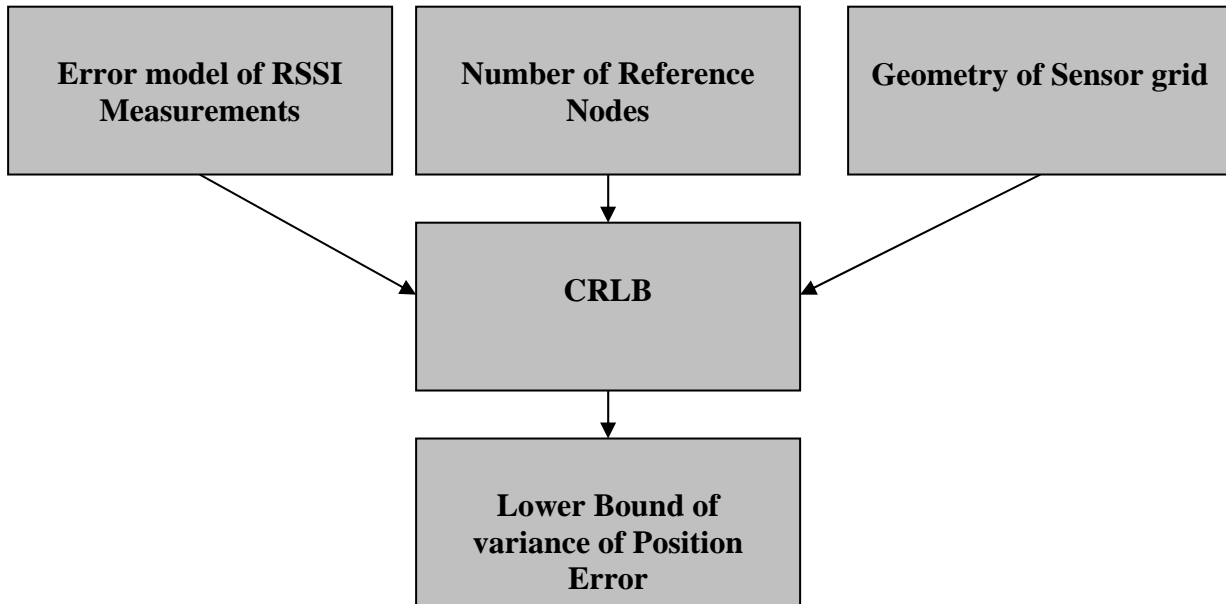


Figure 30: Cramer-Rao Lower Bound on variance of position error

Probability of False Alarm

When a signal is received by the sensor it not only contains the signal but also a noise component. The receiving sensor must decode the signal from the noise and detect the presence of the target. The Probability of False alarm P_{fa} is defined as the probability that a sample R of

the signal $n(t)$ will exceed the threshold voltage V_T when noise alone is present in the radar. It is given by

$$P_{fa} = \int_{V_T}^{\infty} \frac{r}{\psi^2} \exp\left(-\frac{r^2}{2\psi^2}\right) dr = \exp\left(\frac{-V_T^2}{2\psi^2}\right) \quad (30)$$

$$V_T = \sqrt{2\psi^2 \ln\left(\frac{1}{P_{fa}}\right)} \quad (31)$$

The receiver sensor has to set a threshold value above which the signal detects the presence of a target. Figure 31 shows the inverse of the Probability of False alarm is plotted against the normalized threshold values from the above equation. It can be seen that as the threshold increases the probability of false alarm decreases.

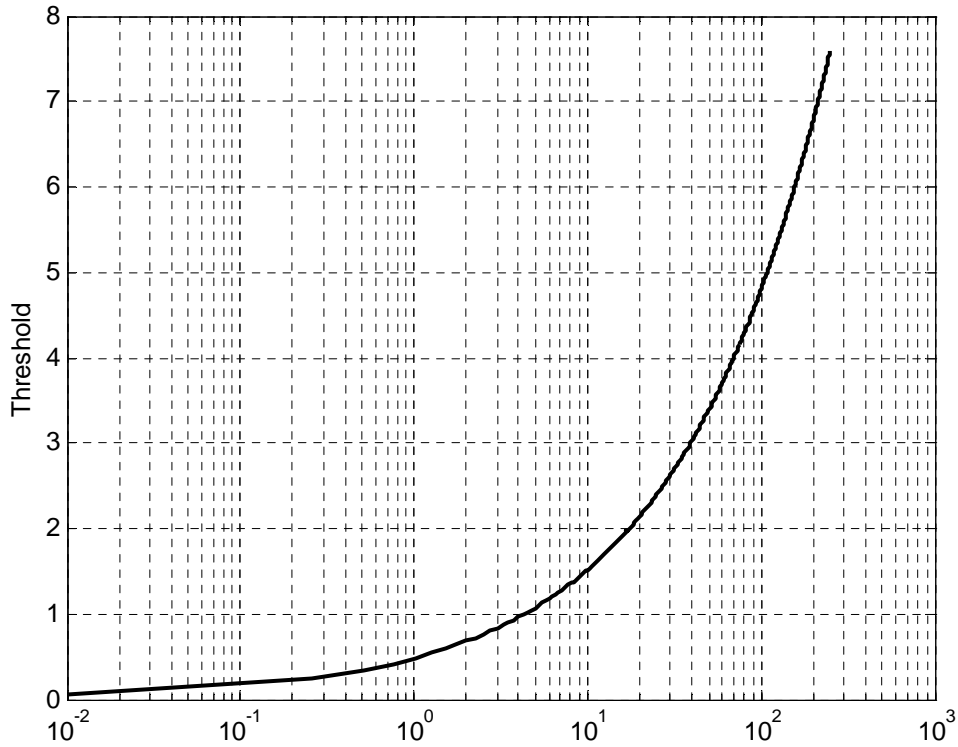


Figure 31: Normalized detection threshold versus probability of false alarm

Simulation Results

The next three sections describe the different simulations that were carried out to determine the characteristics of the hybrid sensor network. Some of the most important factors of the sensor network include the probability of detection, probability of false alarm, detection range and the constant false alarm rate for a given configuration of sensor nodes.

Probability of Detection Vs. Number of Nodes in the Network

We simulated 9 reference nodes in the sensor network and measured the probability of detection of the target in relation to the signal-to-noise ratio for varying number of reference sensors at probability of false alarm $P_{fa} = 10^{-8}$. Though this measure is a factor of the sensor grid configuration it gives us an idea of how many sensors can be added to the system before the additional sensors do not make a significant contribution to the improvement of the probability of detection. Figure 32 shows the signal-to-noise ratio as a function of probability of detection for varying number of reference sensors. It can be observed that after certain the number of reference nodes is reached the gain in probability is no longer significant. This is due to the fact that for the given sensor area adding more reference nodes does not provide any significant advantage because the sensor ranges overlap with each other and offer no improvement in the probability of detection.

The probability of detection was calculated by using the following relation [42].

$$P_D \cong \frac{\operatorname{erfc}\left(\frac{V}{\sqrt{2}}\right)}{2} - \frac{e^{-V^2/2}}{\sqrt{2\pi}} \left[C_3(V-1) + C_4V(3-V^2) - C_6V(V^4-10V^2+15) \right] \quad (32)$$

where

$P_D =$ Probability of Detection

$V =$ Voltage $= \frac{V_T - n(1 + \text{SNR})}{\omega}$

$V_T =$ Threshold voltage

$C_3, C_4, C_6 =$ Gram-Charlier series

$V_T =$ Threshold Voltage

$\text{SNR} =$ Signal-to-Noise ratio

$\omega =$ Frequency

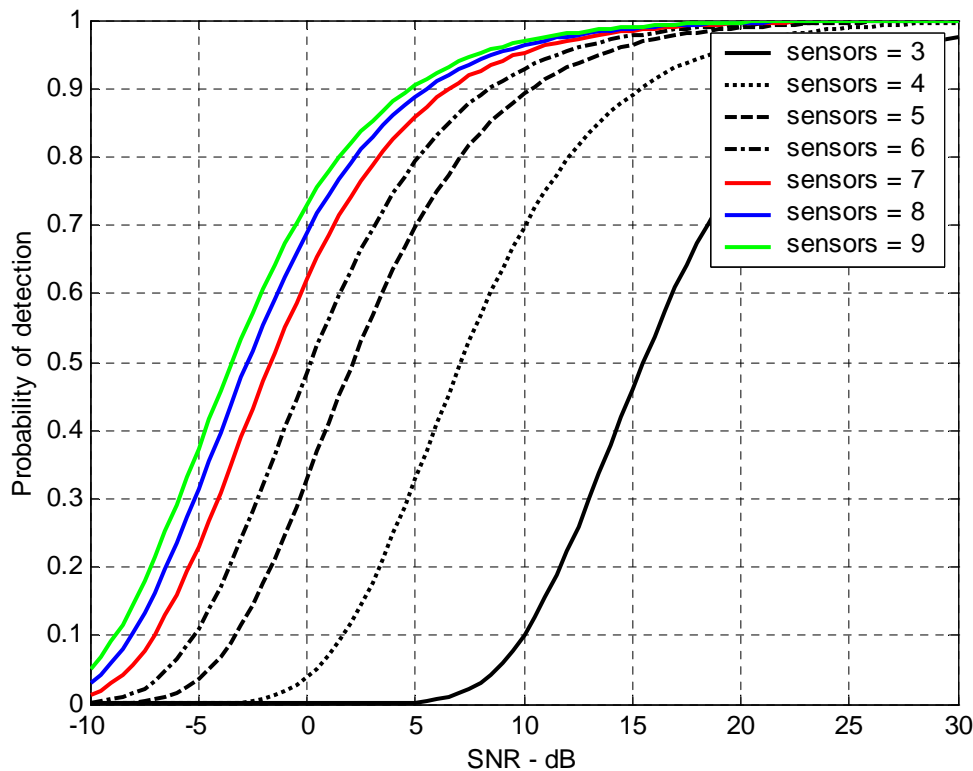


Figure 32: Signal-to-Noise ratio versus Probability of Detection

Detection Range and Signal-to-Noise Ratio

The reference nodes are randomly placed on the sensor grid area and their range of operation is determined the by the Signal-to-Noise ratio of each reference node. Using large number of reference nodes will obviously improve the detection range of the sensor grid. Figure 33 shows the detection range of the sensors as a function of the signal-to-noise ratio for a varying the number of reference nodes. This graph is derived from the equation given by

$$(SNR) = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 k T_e B F L R^4} \quad (33)$$

Where

$P_t =$ Transmit Power = 0 dBm

$G =$ Antenna Gain = 0 dBm

$\lambda =$ Wavelength = .3275 meter

$\sigma = 2$ (indoor)

$T_e =$ Noise Temperature = 290K

$B =$ Bandwidth = 24 MHz

$F =$ Frequency = 916 MHz

$L =$ Antenna Loss = 0 dBm

$R =$ Range = 15 meters

Although it can be seen that using more number of sensors will increase the detection range of the sensor web for lower signal-to-noise ratios this trend will not be sustainable as adding more sensors to the system will increase the noise level and reduce the detection range. Hence there is a limit on the number of sensors one needs to introduce in the network.

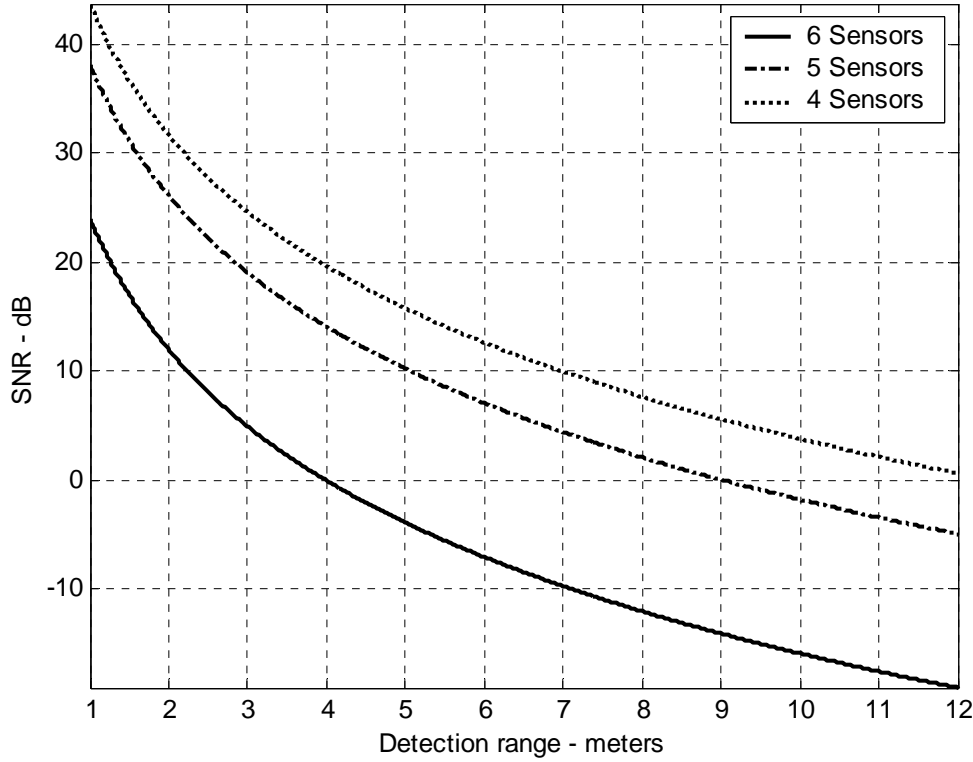


Figure 33: Detection range versus Signal-to-Noise Ratio

Constant False Alarm Rate

The detection threshold is computed so that the receiver can maintain a constant pre-determined Probability of False Alarm. The relation between the threshold, noise power and probability of false alarm is given by

$$V_T = \sqrt{2\psi^2 \ln\left(\frac{1}{P_{fa}}\right)} \quad (34)$$

where P_{fa} is the Probability of False alarm, ψ^2 the noise power and V_T is the threshold. If the noise power is constant then a fixed threshold can satisfy our requirements but unfortunately in

most cases the noise power is a varying phenomenon. Therefore the threshold is varied in order to maintain the constant False alarm rate. This process of continuously changing the threshold in order to maintain the False alarm rate is called Constant False Alarm Rate (CFAR) technique. There are many techniques of CFAR such as adaptive threshold, non-parametric CFAR, Cell-averaging, Greatest of (GO) or Smallest of (SO) CFAR techniques. In our system the Cell Averaging technique is used in which the receiver will average a set of received signal values over a range of receiving nodes and a period of time to set the threshold of detection. Figure 34 shows the cell-averaging process in the receiver of the sensor node. The output of the reference cells to the left and right of the main cell is averaged.

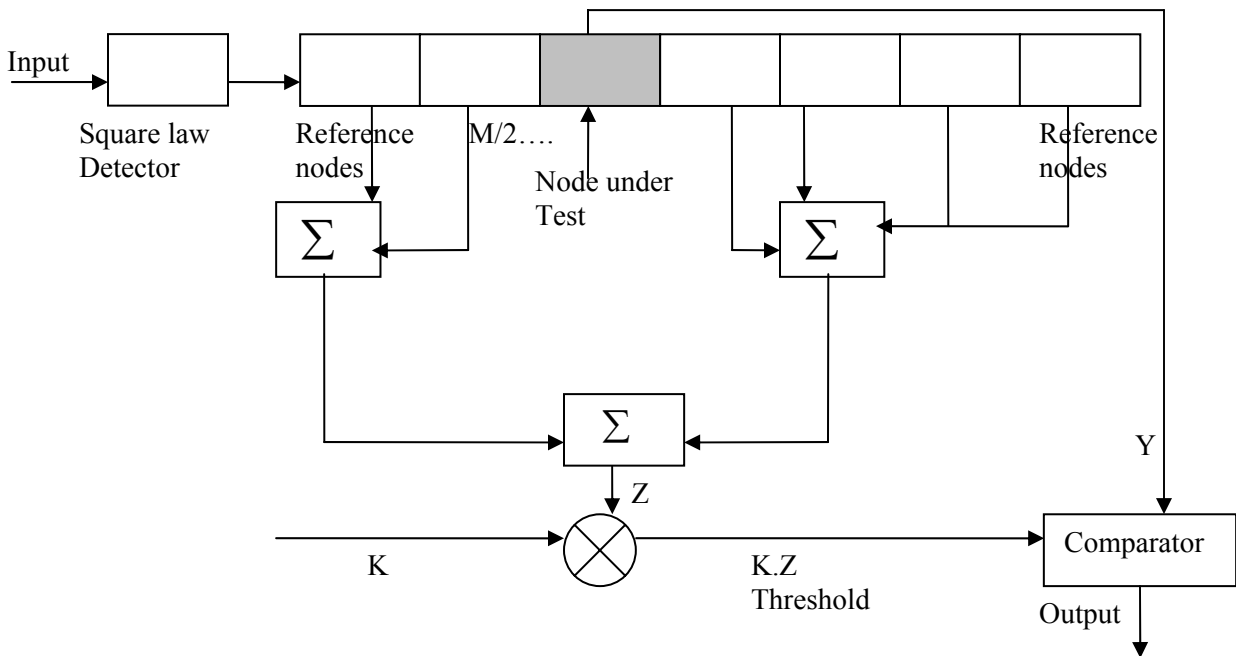


Figure 34: Constant false alarm rate detection

The threshold value is obtained by estimating the average of the reference cells given by Z , multiplying the value by a constant K (for scaling) and comparing it to the output of the Node

under Test. A detection is declared if $Y \geq KZ$. Cell-averaging assumes that the target of interest is in the Node under Test and the reference cells contain zero mean Gaussian noise of variance ψ^2 . The conditional probability of False alarm $y = V_T$ can be written as

$$P_{fa}(V_T = y) = e^{-y/2\psi^2} \quad (35)$$

The unconditional probability of false alarm is

$$P_{fa} = \int_0^{\infty} P_{fa}(V_T = y) f(y) dy \quad (36)$$

where $f(y)$ is the *pdf* of the threshold and given by

$$f(y) = \frac{y^{M-1} e^{-y/2K\psi^2}}{(2K\psi^2)\Gamma(M)} \quad (37)$$

Integrating the P_{fa} in the above equation yields $P_{fa} = \frac{1}{(1+K)^M}$ which is independent of the noise power [44].

Hardware

The hardware used for this project is the Mica2 sensor developed by CrossBow Technologies and University of California, Berkeley. Our objective was to understand the underlying hardware architecture to access the required parameters for our application. This section will explain in brief the chipset used for the Mica2 sensor and our method to access the relevant parameters. The Mica2 sensor uses a chipset called the CC1000 manufactured by ChipCon Inc. The features of this chipset are

Frequency selectable transmission from 300 – 100 MHz

Data transmission rate of 76.8 Kb/sec

Selectable transmission of Power levels

Manchester encoding

CSMA/CD coding scheme for data transmission

Figure 35 shows the CC1000 chipset with the various internal devices. The Logger Flasher processes the data uploaded into the chipset through the 51-pin connector and sends the data to the Analog and Digital data convertor. The tunable radio frequency module transmits the data through the external antenna.

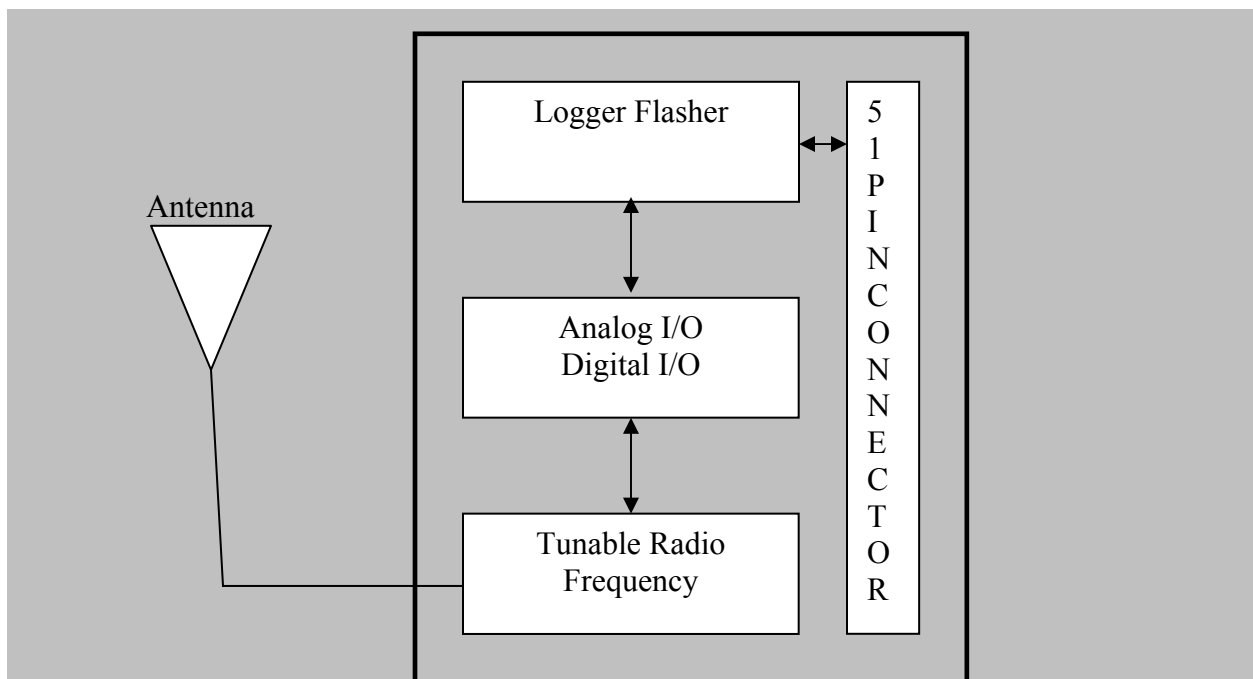


Figure 35: CC1000 Chipset components

We used NesC to program the chipset to access the RSSI values for measuring the distance between the different sensors. The transmitting Mica2 sensor is coded with the following commands shown in Figure 36 to transmit a message at periodic user-defined intervals.

```

typedef nx_struct RssiMsg{
    nx_uint16_t rssi;
} RssiMsg;

```

Figure 36: Message from Transmitting Sensor

The signal transmitted is received by another Mica2 sensor at a calibrated distance from the transmitter. The receiving sensor is coded with NesC code shown in Figure 37. The receiving sensor node transmits the data to the computer through the serial interface where the measured RSSI is converted to dBm units.

```

event bool RssiMsgIntercept.forward(message_t *msg, void *payload, uint8_t
    len) {
    RssiMsg *rssiMsg = (RssiMsg*) payload;
    rssiMsg->rssi = getRssi(msg);

    return TRUE;
}
#ifdef __CC2420_H__
uint16_t getRssi(message_t *msg){
return (uint16_t) call CC2420Packet.getRssi(msg);
}
#elif defined(CC1K_RADIO_MSG_H)
uint16_t getRssi(message_t *msg){
cc1000_metadata_t *md =(cc1000_metadata_t*) msg->metadata;
return md->strength_or_preamble;
}
#else
#error Radio chip not supported! This demo currently works only \
for motes with CC1000 or CC2420 radios.
#endif //__CC2420_H__

```

Figure 37: Receiving Code for measuring RSSI

Programming Aspects

This section will describe in detail the TinyOS Operating system and how to use NesC for programming the sensors. TinyOS is a free source component based Operating System used for

embedded sensors and wireless sensor networks. TinyOS is designed to be hosted on small scale devices with limited memory and processing power. TinyOS can be hosted on any device that is limited by processing and memory capacity such as miniature sensors and other devices. TinyOS is written in NesC language.

TinyOS Components

A TinyOS application consists of one or more components connected to form an interface. A component provides and uses interface. The components can only be accessed through the interface. For the component to execute the interface it should execute all the commands in the interface. There are two types of components: modules and configurations. Modules are application code written in NesC language. Configurations join different components together and also wire the interfaces together. Figure 38 shows the structure of programming using TinyOS and nesC [45].

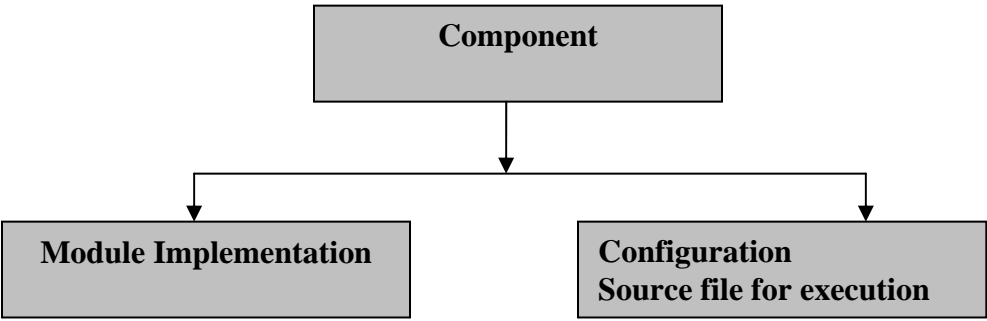


Figure 38: TinyOS Structure Implementation

The Mica2 mote sensors have in-built functions in order to successfully control the frequency and transmit power of the mote sensors. In this section we have detailed the programming aspects and the procedure to pre-set the frequency and transmit powers of the mote sensors. In the mote sensor application that is installed using the TinyOS operating system there are different folders and directories under the Cygwin directories that contain different control and executable files that will be used to manage the operations of the Mica2 mote sensors.

Searching for the configuration file that controls the frequency and transmit power of the Mica2 mote sensor we found that the file “MakeXbowLocal.nc” contains the different parameters of frequency and power settings. This file is found under the directory “C:\tinyos\cygwin\opt\tinyos-1.x\contrib\xbow\apps” in any normal user installed system. Figure 39 below shows the frequency range of the sensors. We can thus choose our frequency in a given band of operation of the sensors by automatically reading the needed channel from this configuration file.

```

# 916 MHz Band
#
# CHANNEL_00 - 903 MHz      CHANNEL_02 - 904 MHz      CHANNEL_04 - 905 MHz
# CHANNEL_06 - 906 MHz      CHANNEL_08 - 907 MHz      CHANNEL_10 - 908 MHz
# CHANNEL_12 - 909 MHz      CHANNEL_14 - 910 MHz      CHANNEL_16 - 911 MHz
# CHANNEL_18 - 912 MHz      CHANNEL_20 - 913 MHz      CHANNEL_22 - 914 MHz
# CHANNEL_24 - 915 MHz      CHANNEL_26 - 916 MHz      CHANNEL_28 - 917 MHz
# CHANNEL_30 - 918 MHz      CHANNEL_32 - 919 MHz      CHANNEL_34 - 920 MHz
# CHANNEL_36 - 921 MHz      CHANNEL_38 - 922 MHz      CHANNEL_40 - 923 MHz
# CHANNEL_42 - 924 MHz      CHANNEL_44 - 925 MHz      CHANNEL_46 - 926 MHz
# CHANNEL_48 - 927 MHz
#CFLAGS = -DCC1K_DEFAULT_FREQ=CC1K_914_077_MHZ
#CFLAGS = -DCC1K_DEFAULT_FREQ=CC1K_915_998_MHZ
CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_00
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_02
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_04
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_06
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_08
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_10
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_12
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_14
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_16
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_18
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_20
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_22
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_24
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_26
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_28
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_30
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_32
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_34
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_36
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_38
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_40
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_42
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_44
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_46
#CFLAGS = -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_48
#-----
#

```

Figure 39: Varying the frequency on the Mote sensor

Figure 40 shows the different power levels that can be used for transmission of the signals from the Mica sensors. This is also contained in the MakeXbowlocal configuration file.


```

#####
#
# MICA2 RF Power Levels
#TXPOWER_MAX      TXPOWER_0DBM
#TXPOWER_0DBM     0x1f    //-0dBm
#TXPOWER_M3DBM    0x23    //-3dBm
#TXPOWER_M5DBM    0x19    //-5dBm
#TXPOWER_M10DBM   0x0B    //-10dBm
#TXPOWER_M15DBM   0x07    //-15dBm
#TXPOWER_M25DBM   0x03    //-25dBm
#TXPOWER_MIN      TXPOWER_M25DBM
#####

```

Figure 40: Varying the frequency on the mote sensor

There are advantages and disadvantages to varying the frequency and power of the Mica2 mote sensors during the tracking of the target node. The primary advantage is that the nodes will be able to receive the signal even in presence of multipath interference from various materials in the indoor conditions and signal attenuation. The disadvantages are that larger memory is needed to store the collected data from the reference nodes, the latency of the system increases and there are more message transmissions which will use up bandwidth or cause interference with other co-existing wireless systems.

Table 7 shows the various advantages and the disadvantages of varying frequency and transmit power in the sensor system

Table 7: Frequency and power advantage and disadvantage

Characteristic	Advantage	Disadvantage
Frequency	Increases accuracy	More memory needed on sensors; more message transmission
Power	Increases accuracy	Increase latency; lower power sensors might not receive signal due to interference

In our setup each node cycles through three frequency and at each frequency three power levels are used. At initialization each reference node starts to transmit at a random chosen frequency (to

minimize collision) and then cycles through the set of pre-set frequencies. It then switches to the next power level and cycles through the set of frequencies again. Figure 41 shows this cycling through the frequency and power level done by the reference nodes.

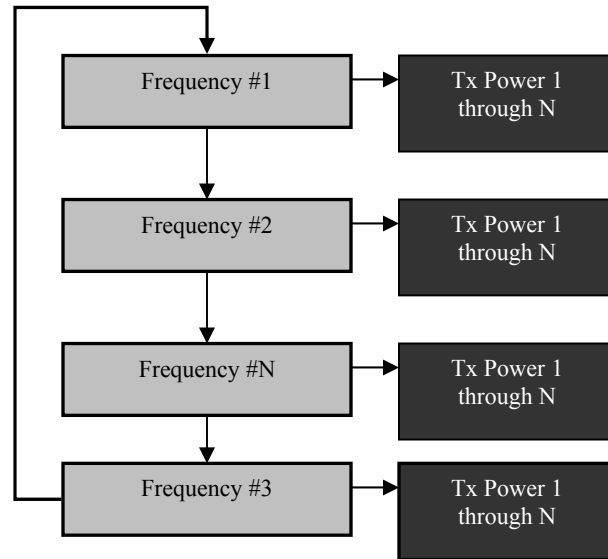


Figure 41: Varying frequency and transmit power on the reference nodes

Since the target node will not know which frequency the reference node is transmitting at any given moment, the easiest method is to assume that the target node estimates its location only after a time interval greater than the maximum time required for reference node to cycle through all the pre-set frequencies. Although this will increase the accuracy of the system it also adds to the latency. Latency results have been discussed in the other sections of this chapter.

Compiling and Installing with Mica2 Sensor

The Mica2 sensor is placed on a serial interface board called the M510 in order to upload the code through the computer to the sensor. Figure 42 shows a MIB510 serial interface board with a Mica2 mounted on it. The MIB510 board is connected through a serial port to the computer for uploading the code into the Mica2 sensor. There are two methods to compile and upload the code into the Mica2 sensor. We can either use a Linux based computer or a Windows based system with Cygwin. Cygwin is a Linux emulator type application on a Windows machine. The command issued to compile the code is “*make -f Makefile mica2 install.0 mib510,/dev/ttyS3*”. This command compiles the code on the computer and transmits the executable code through the serial port # 2 (ttyS3 = port #2) of the computer to the Mica2 sensor.



Figure 42: MIB510 Serial Interface board with Mica2 Sensor

If the compile is successful the messages shown in Figure 43 will scroll through the computer screen indicating that the code was successfully compiled and Mica2 sensor will contain the executable code to make the application work. If there are errors during compile operation or the

computer port is no available there will be appropriate error messages on the screen to indicate this fault.

```
compiling Blink to a mica binary
  ncc -board=micasb -o build/mica/main.exe -Os -target=mica -Wall
-Wshadow -DDEF_TOS_AM_GROUP=0x7d -finline-limit=200 -fnesc-
cfile=build/mica/app.c Blink.nc -lm
  avr-objcopy --output-target=srec build/mica/main.exe
  build/mica/main.srec
    compiled Blink to build/mica/main.srec
    installing mica binary
  uisp -dprog=dapa --erase
  pulse
  Atmel AVR ATmega128 is found.
  Erasing device ...
  pulse
  Reinitializing device
  Atmel AVR ATmega128 is found.
  sleep 1
  uisp -dprog=dapa --upload if=build/mica/main.srec
  pulse
  Atmel AVR ATmega128 is found.
  Uploading: flash
  sleep 1
  uisp -dprog=dapa --verify if=build/mica/main.srec
  pulse
  Atmel AVR ATmega128 is found.
  Verifying: flash
```

Figure 43: Compile messages on Computer indicating successful compile and install

The data from the Mica2 sensor is sent to the computer through the serial board interface MIB510 and is processed in an application called the SerialForwarder that runs in the Cygwin environment of the computer. Figure 44 shows the SerialForwarder Java interface application that collects data from the Mica2 sensor and processes the data.

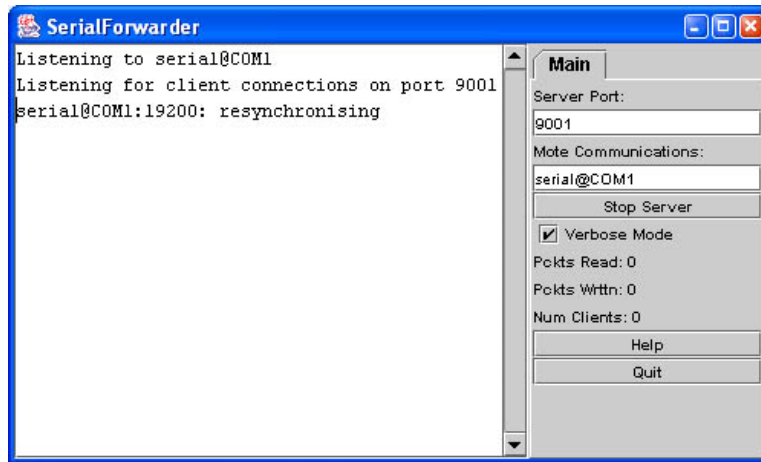


Figure 44: Serial forwarder interface to computer

Connecting Multiple Nodes to form Sensor grid

Once we are able to receive a simple RSSI signal to measure the distance with one pair of transmitter and receiver we can now form a sensor grid. We set up experiments in the Institute for Simulation & Training lab with six sensors to form a sensor grid. We program each of the sensors using the serial MIB510 board and place them at the pre-determined locations. We restricted our sensor web to the two dimensional X-Y plane and also to a 35 x 35 meter in order to be able to accurately take measurements and verify the results through simulation. Although a large area for the sensor web is feasible it would be difficult to accurately place the sensors and map their position to the sensor grid. The data collection is done using a USB interface that connects the central station to the nodes and the computer. The computer's Graphical User Interface displays the location of the sensor nodes and the tracked target sensor as it moves along in the sensor grid. Figure 45 shows the sensor network grid as displayed on the GUI of the computer with the locations of the reference nodes and also the target node.

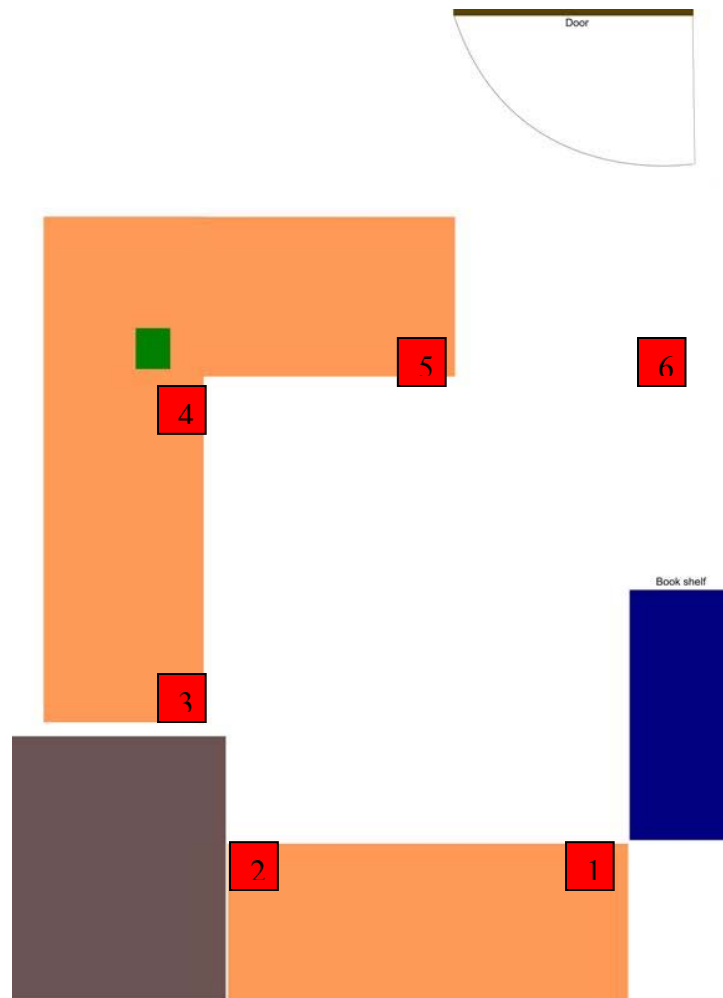


Figure 45: Sensor web grid with reference nodes and the target node

It can be seen that the reference nodes are in line-of-sight of the target node although this system is capable of detecting the target even in non line-of-sight conditions. The data collected from the nodes is displayed through this Java based GUI capable of updating and refreshing the position location of the target at 30 Hz.

Example of Target Tracking in Static Sensor Network

Figure 46 shows the tracking of a target node on the static sensor grid by the six reference nodes that form the grid. The target is moving along a straight line in our simulation. The Blue circle dots indicate the target node location as it moves through the sensor grid. The target is being tracked by 3 reference nodes located at (1, 20), (1, 5) and (10, 5). Although the actual movement of the target is along the x-axes from (0, 10) to (25, 10) the simulation accounts for the various errors in the estimation of the location of the target.

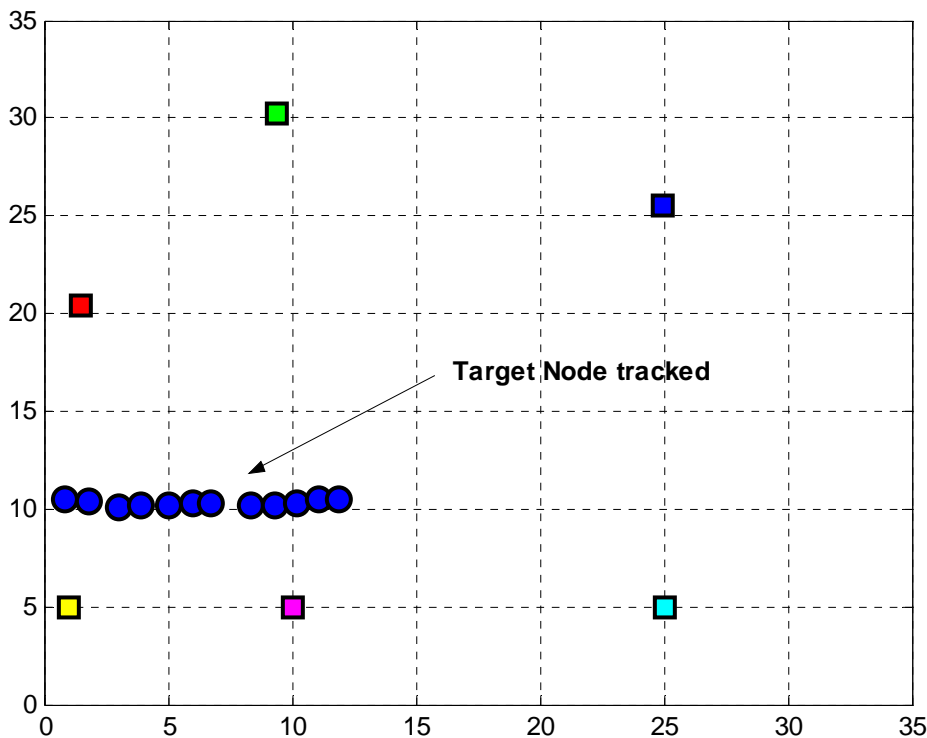


Figure 46: Target node tracked in the Sensor grid

As the target moves along the straight line a minimum of reference nodes track the target. But at a certain location on the sensor grid there are only two reference nodes in range of the target and the reference nodes lose track of the target. This is indicated by the red dots in Figure 47 that shows the reference nodes losing track of the target as it moves through the grid.

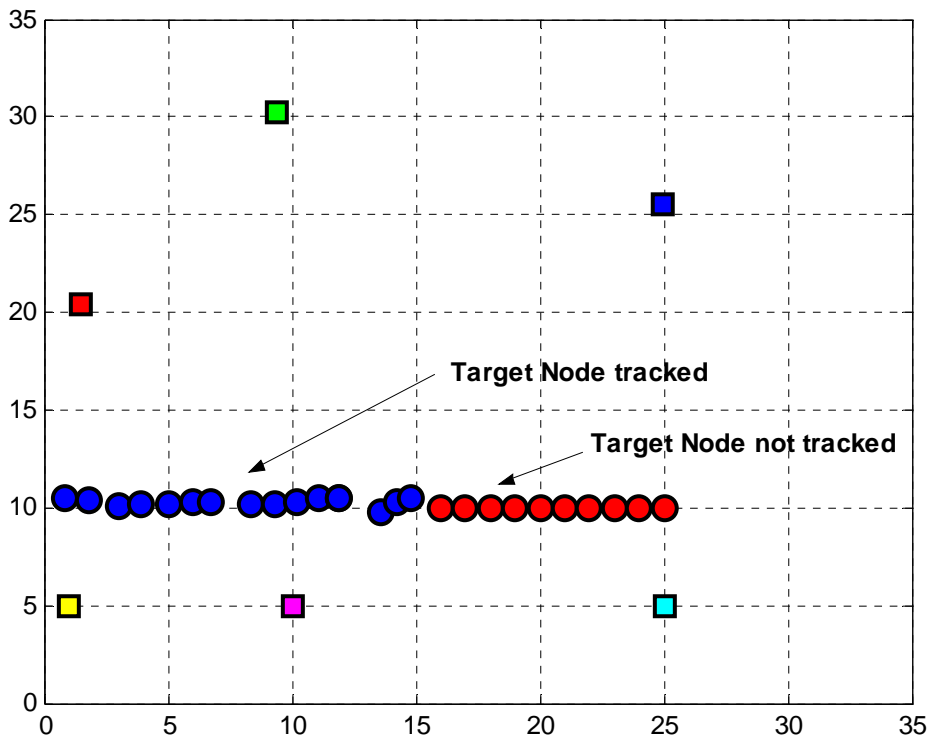


Figure 47: Target out of range for detection the static reference node grid

The Table 8 shows the output of the tracking software as it displays the location of the tracked sensor through the sensor grid which enables the user to determine the reference nodes that are tracking the target.

Table 8: Output from tracking Sensor software

Time Interval	1	2	3	4	5
Data from sensor grid	loc1 = 20.0805 10.4118 D1 in range D2 in range D3 in range	loc1 = 22.0589 8.5390 D1 in range D2 in range D3 in range	loc1 = 22.0799 10.1339 D1 in range D2 in range D3 in range	loc1 = 22.4749 11.4267 D1 in range D3 in range Required Sensors not in range	D1 in range D3 in range Required Sensors not in range

As it can be from the previous experiment with static reference nodes that the target node is not continuously tracked in the sensor grid due to the limitations of the static reference nodes in not being in range of the continuously moving target. Although by incorporating additional sensors in the grid the coverage can be improved this leads to increased costs of adding more sensors and deploying them. A cost-benefit analysis of adding more sensors to a static network compared to introducing using a hybrid network will be provided in later sections of this chapter.

Target Tracking with Mobile Reference Nodes

The previous set of simulation was conducted with the reference nodes located in static positions with the target moving through the sensor grid. The next set of simulations will involve the reference sensor nodes with the capability to move and relocate their position based on the direction of the target motion. Figure 48 shows the target sensor node being tracked for a period of time but and then not being tracked due to the lack of the minimum three reference nodes needed for continual tracking. This phenomenon is the same as was seen for the static network.

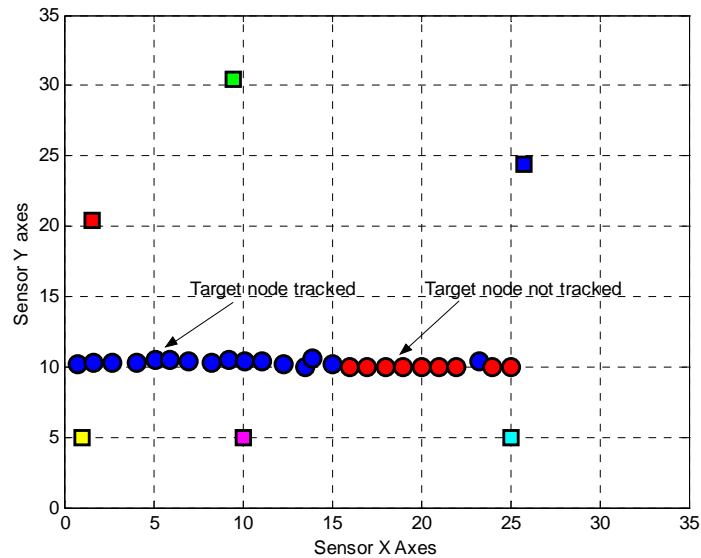


Figure 48: Target node traversing the sensor grid but unable to be tracked

To continuously track the target through the sensor grid we introduce the hybrid network. In our design one of the reference nodes closet to the target moves in the direction of the last known position of the target node for a specific time, and then localize its position on the sensor grid. This way, once again three reference nodes are available for tracking. In Figure 49 the reference node located at (1,5) moves to the new location at (5,5). We select a reference node that can be made mobile and moved to the right location to continue tracking the target. Since the target node was moving in the X direction the reference node also moves in the X direction.

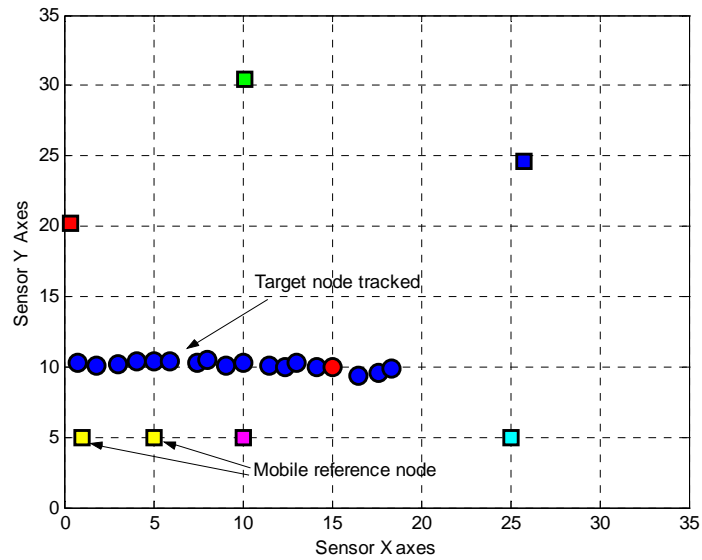


Figure 49: Mobile reference node used to continuously track the target

If the target node moves out of range again, then another reference node will move and continue to track the target. This is shown in Figure 50

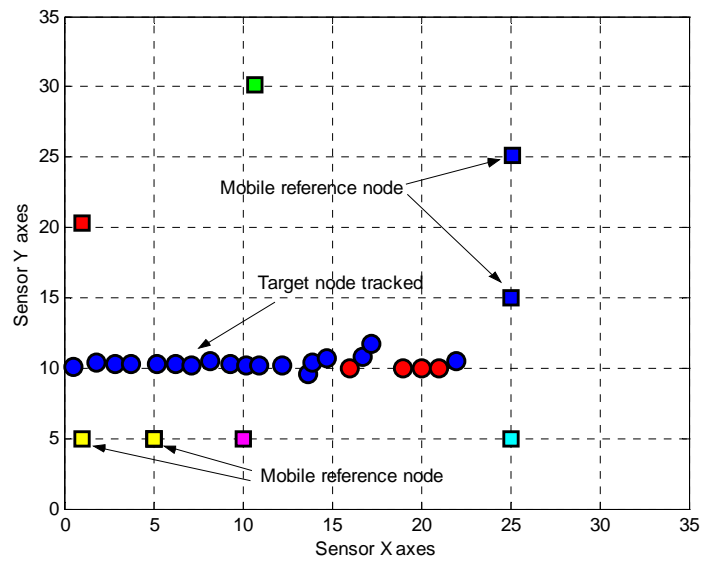


Figure 50: Second mobile sensor node movement to track target

Figure 51 shows how the target node is being tracked in the sensor grid by three mobile reference nodes. Allowing the nodes to move only as needed and moving the appropriate node (i.e. to form 3 reference nodes for localization), the number of sensors needed and the overall cost of the network is reduced as compared to a network of mobile sensors.

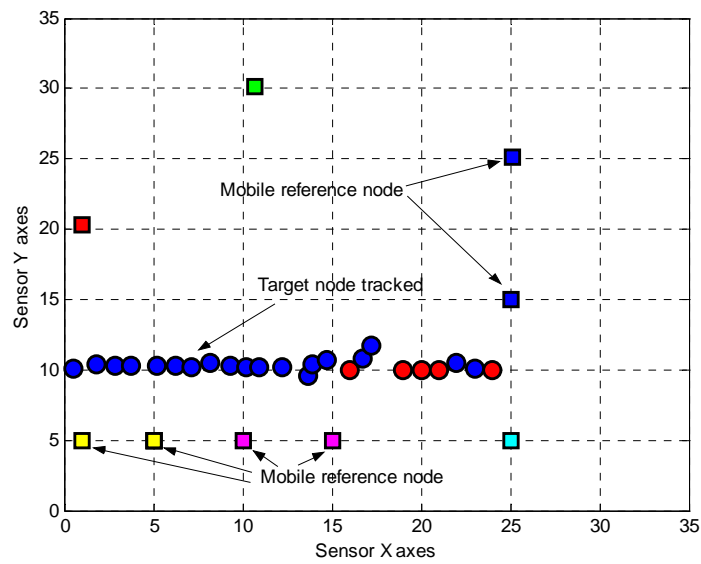


Figure 51: Three mobile reference nodes for target tracking

When the reference node moves to a new location it will need to relocalize its position information again within the sensor grid. This introduces latency in the tracking system. The more the number of mobile sensor nodes the larger the latency in the system.

Error Correction using Kalman Filtering

As discussed earlier Kalman filtering technique is used to correct for the localization errors from the target node. The Kalman filter is given by the following relationship defined by [46]

$$\begin{aligned}
 S_k &= P_k + R \\
 K_k &= AP_k S_k^{-1} \\
 P_{k+1} &= AP_k A^T + Q - AP_k S_k^{-1} P_k A^T \\
 x_{k+1} &= A\hat{x}_k + K_k(z_{k+1} - A\hat{x}_k)
 \end{aligned} \tag{38}$$

The process noise is given by w_k is white and has a co-variance of Q. Also the noise measurement is given by v_k with a co-variance of R and is unrelated from the process noise. In the above equation the matrix transpose is indicated by the superscript T, S is the co-variance of innovation, K is the gain and P is the prediction error. A, B and H are matrix constants

Figure 52 shows the entire predictor-corrector update of the Kalman filter process.

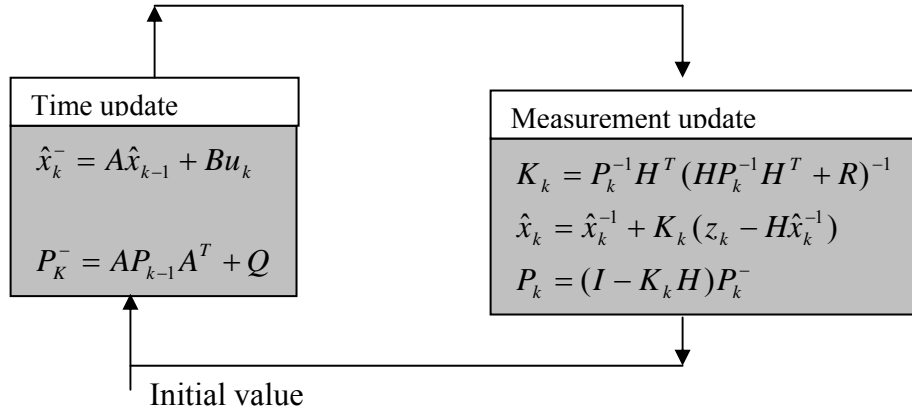


Figure 52: Kalman filter process

The Kalman filter reads in the estimated location of the target node, uses the measured values and then corrects for the errors by predicting the location of the target node in the future. Figure 53 shows the working of the Kalman filter in our target node tracking system. The Kalman filter receives the estimated values from the sensors and the corrected values from the error corrector. These two inputs act as the measured and estimated values for the Kalman filter and as explained earlier the Kalman filter outputs the

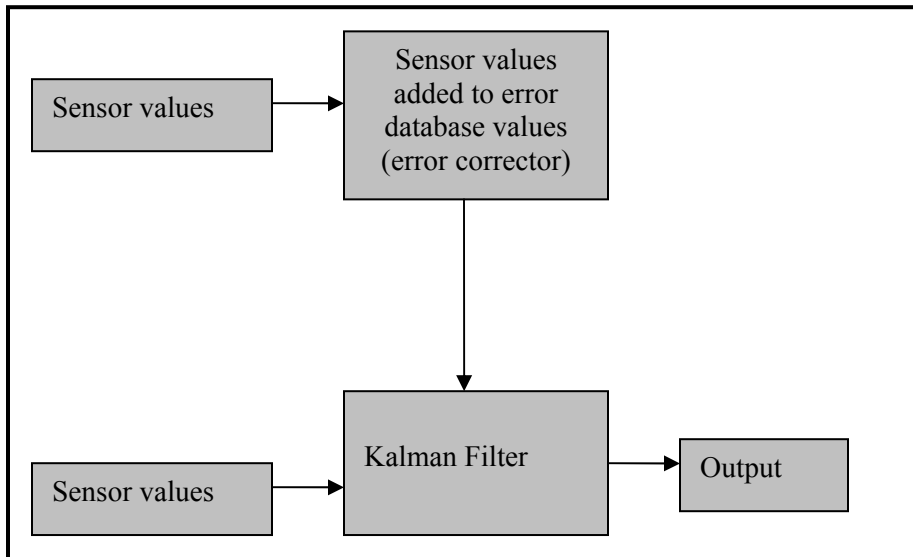


Figure 53: Kalman filter operation in target node estimation

To demonstrate the working of the Kalman filter and its efficient use in our sensor network system we measured the random motion of a target with bursts of acceleration with a standard deviation of 0.5 feet/sec². The position was measured with an error of 10 feet. Figure 54 shows the results of the Kalman filter used to locate the position of the target in presence of noise. The

parameters used for this simulation are: Acceleration 0.5 feet/sec^2 , Mean noise 5 feet/sec, measurement error co-variance = $(\text{Mean noise})^2$ [47]

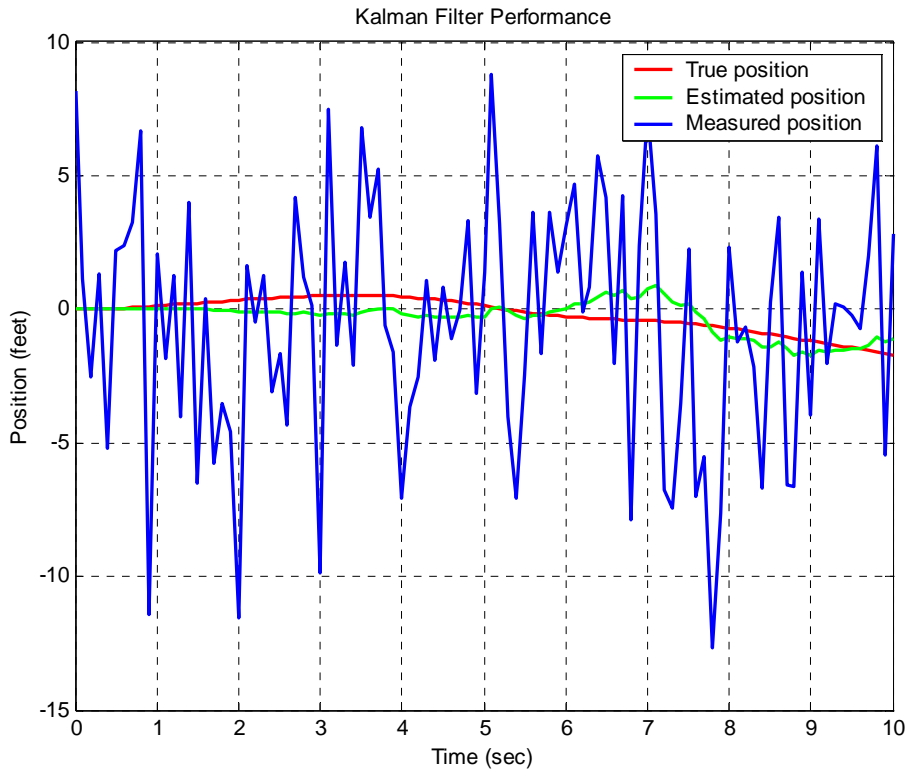


Figure 54: Kalman filter used for target tracking in presence of noise

Figure 55 shows the filtered error for our Kalman filter as a function of distance (m) travelled in meters by the target in the sensor grid. The initial spike in the filtered error is due to the estimated value that is input into the filter which is subsequently corrected by the filter. The filter then uses the predictor corrector algorithm to reduce the error in the measured location of the target. Figure 56 shows the measured error without Kalman filter as a function of the distance travelled by the target. It can be seen that as the target travels a greater distance the location

estimation algorithm determines the location of the target but the errors accumulate over a period of time .

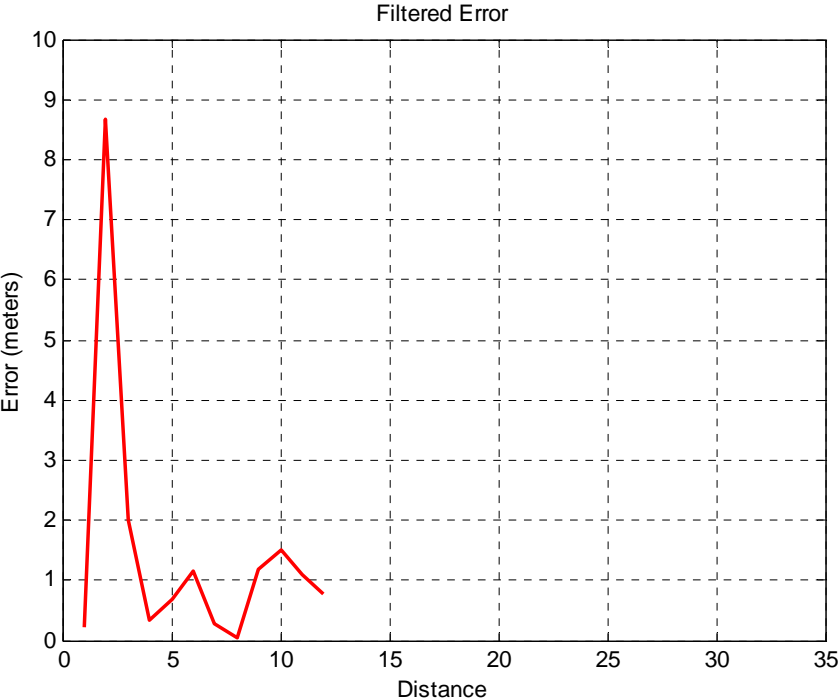


Figure 55: Filtered error from sensor

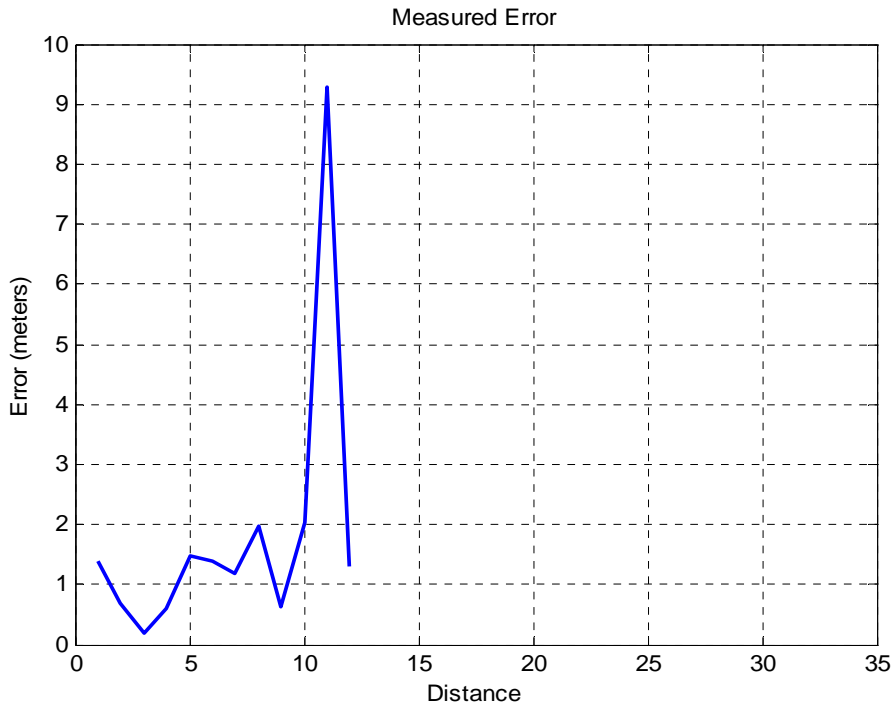


Figure 56: Measured error on the sensor

Evaluation of the Sensor System

In order to determine the accuracy of our hybrid system we examined the characteristics of the hybrid sensor system in terms of transmission powers used by the reference nodes; the frequencies used by the reference nodes; the number of reference node that are used for the estimation location algorithm and the number of operational reference nodes. We also measured the latency of the system in terms of number of samples of data that are collected for the estimation.

Number of Reference Nodes

Using five reference nodes we studied the effect on the accuracy of the hybrid sensor system as the number of reference nodes was varied. Figure 57 shows the error distance in feet as a function of number of reference nodes used in our experiment for varying power levels of transmission emitted by the reference nodes. The error distance is defined as the error between the actual and measured location of the target. Due to limitations on the number of hardware nodes that were available for the experimentation we restricted our experiments to five reference nodes using three power levels (0 dBm, 5 dBm and 10 dBm) for transmission. It can be clearly seen that as the number of reference nodes are increased the estimated location error decreases for increasing power levels of transmission.

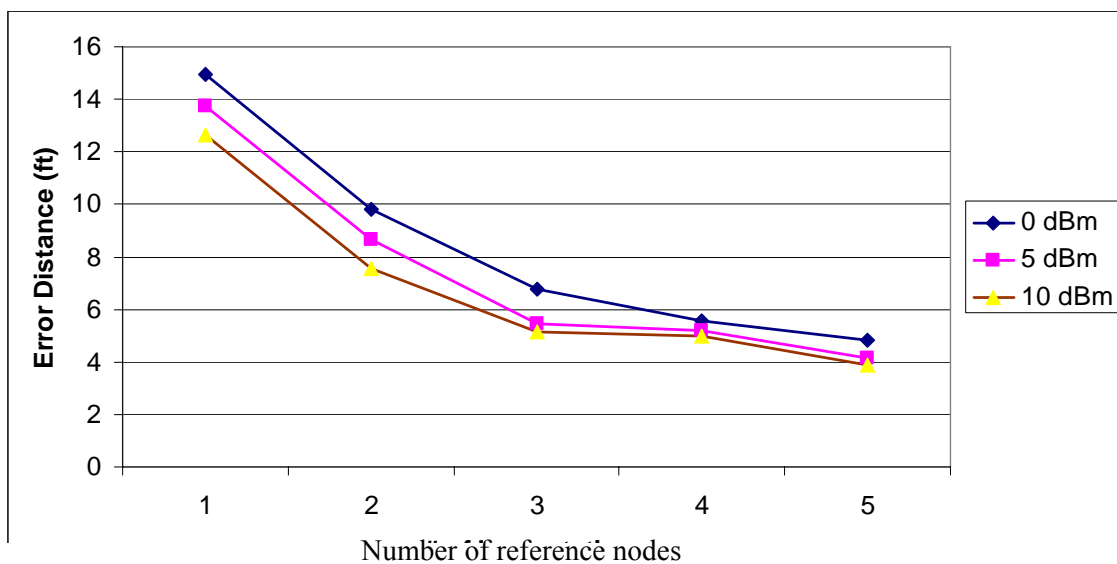


Figure 57: Error distance (in feet) as the number reference nodes are increased

Number of Frequencies

Another parameter that was considered in the tracking algorithm is the number of frequencies. The reference nodes cycle through a set of frequencies in order to improve the accuracy of the system in various environmental conditions. The tracking system will be robust if the number of frequencies is increased to accommodate various in-door conditions. However the latency of the system will increase as the mobile node will need to wait for the reference nodes to cycle through all the frequencies before arriving at location estimation. Figure 58 below shows the estimated errors as a function of number of frequencies of the reference used for 0 dBm power level transmission. It is noted that as the number of frequencies is increased the error decreased. Although we have considered four frequencies it is possible to increase the number of frequencies used based on available sensors.

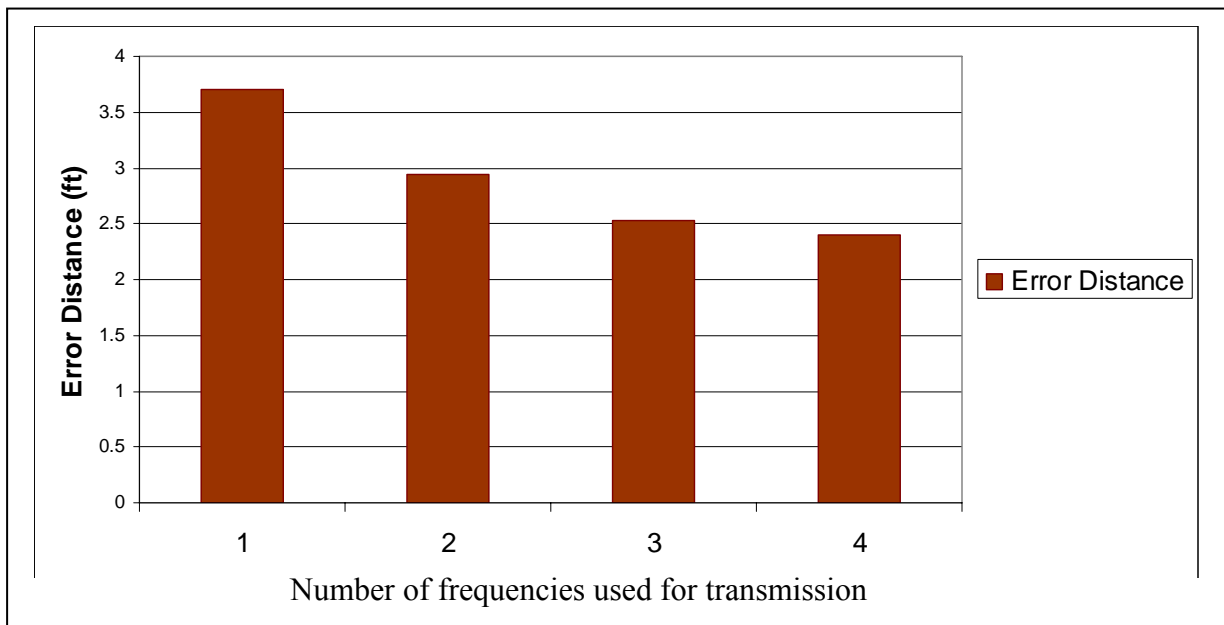


Figure 58: Target error distance as a function of frequency

Number of Transmit Power Levels

Figure 59 below shows the measurement error in terms of Transmit power from the reference nodes at one frequency. It can be seen that as the transmit powers in increased the measured error decreases. Generally there is a limit on the transmit power due to hardware considerations and also higher transmit powers increase battery consumption thereby reducing the lifetime of the sensors. Thus there has to be a trade off between accuracy and lifetime of the sensors.

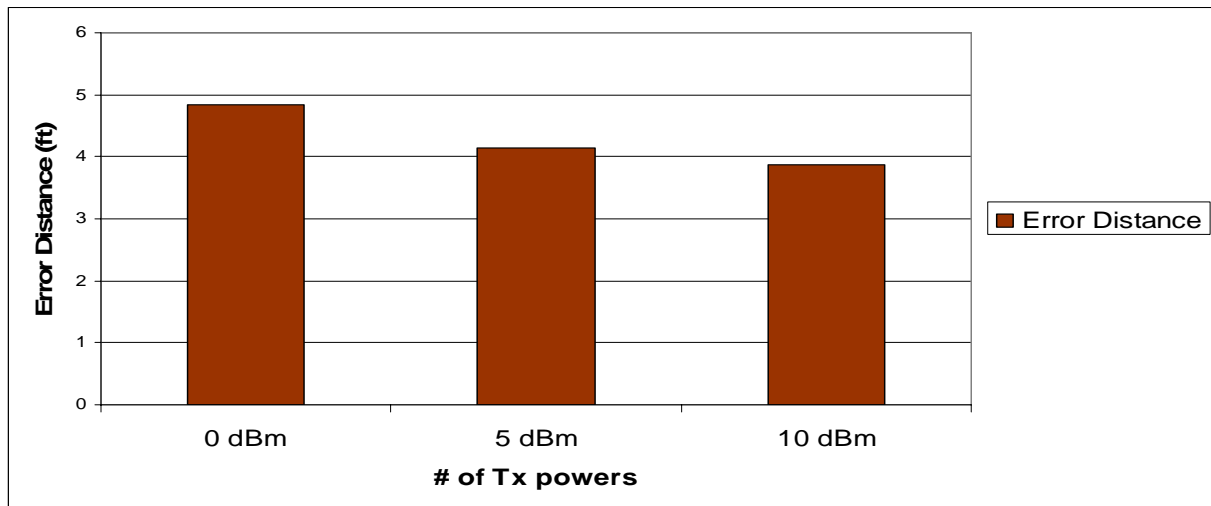


Figure 59: Error distance of target as a function of transmit power

Number of Reference Signatures

Figure 60 below shows the error distance as a function of the number of reference signatures that are stored in the mobile node used for estimation location. When the mobile node stores a large number of reference signatures the measured error reduces as the mobile node will have adequate number of reference data to compare the collected values for position estimation. But the trade-off again in this case is memory availability on the mobile node and the time required for the mobile node to gather and update all the reference signatures periodically.

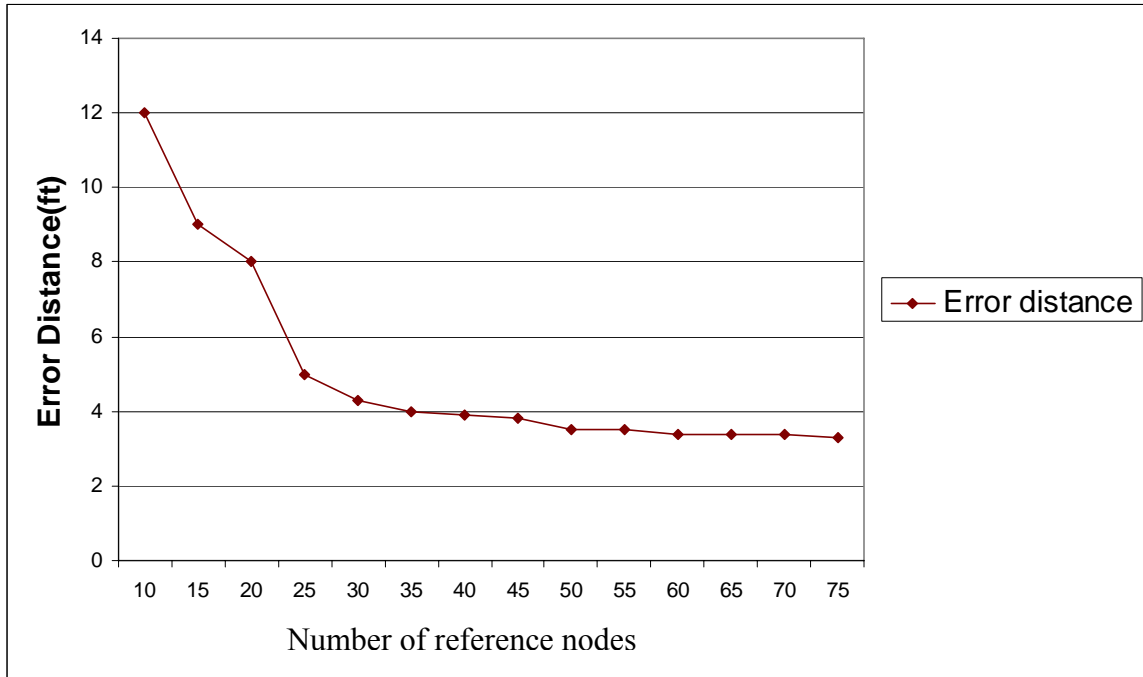


Figure 60: Target Error as a function of number of reference signatures

Number of Data Samples

We studied the effect of varying the number of data samples from the reference nodes on the latency of the system. The latency is defined as the time difference between the target at a certain location to the time taken by the algorithm to estimate its location. Figure 61 below highlights the latency of the system as the number of data samples that are collected by the mobile target node is increased. It can be argued that a large number of samples can improve the accuracy of the system as the mobile target will calculate the average of the samples and therefore reduce errors. But the latency of the system will increase as the target will take greater time to acquire the samples.

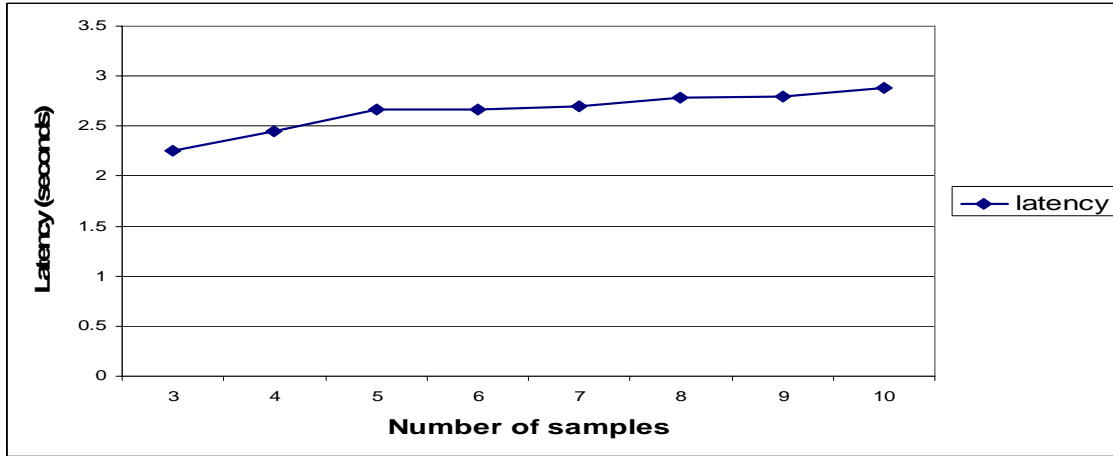


Figure 61: Latency as a function of Number of Data Samples

Cost Analysis for Static Sensor Deployment

The hybrid sensor field consists of static and mobile sensor nodes that work synchronously to detect and track a target in the sensor web. This section will analyze the cost of deploying the individual sensors in the sensor field and determine an optimum cost function for a given area to maximize the coverage. A naïve analysis of a sensor web system would indicate that to maximize coverage deploying a large number of sensors is ideal. But there are many factors to consider before randomly deploying a large number of sensors to cover a given area. Although each sensor node is very inexpensive there are associated costs in deploying them either manually or through air-borne methods. We have identified a set of factors that would influence the cost function of sensor deployment and associated a percentage of influence or weights to each factor. Researchers planning to deploy a sensor web system can vary weights of each factor to determine the optimal cost function for a sensor web. We calculate the cost of deployment as:

$$\text{Cost of deployment } Cd = Co(n) + statx * statn - Ns - Ps \quad (39)$$

where $Co(n)$ the coverage obtained by using n static sensors; $statx$ the cost of each sensor; $statn$ is the number of sensors; Ns Signal-to-Noise ratio and Ps Probability of detection. Figure 62 shows the cost function for using a total of 9 sensors in our 35 x 35 meter sensor grid area. As the number of sensors is increased the cost function also exponentially increases in the graph. There is a very steep increase in the cost from sensor 7 to 8 as seen in the plot. This could be due to lower probability of detection that we measure since the sensor grid is fully covered and hence there is no appreciable increase the detection capability is gained by adding more sensors.

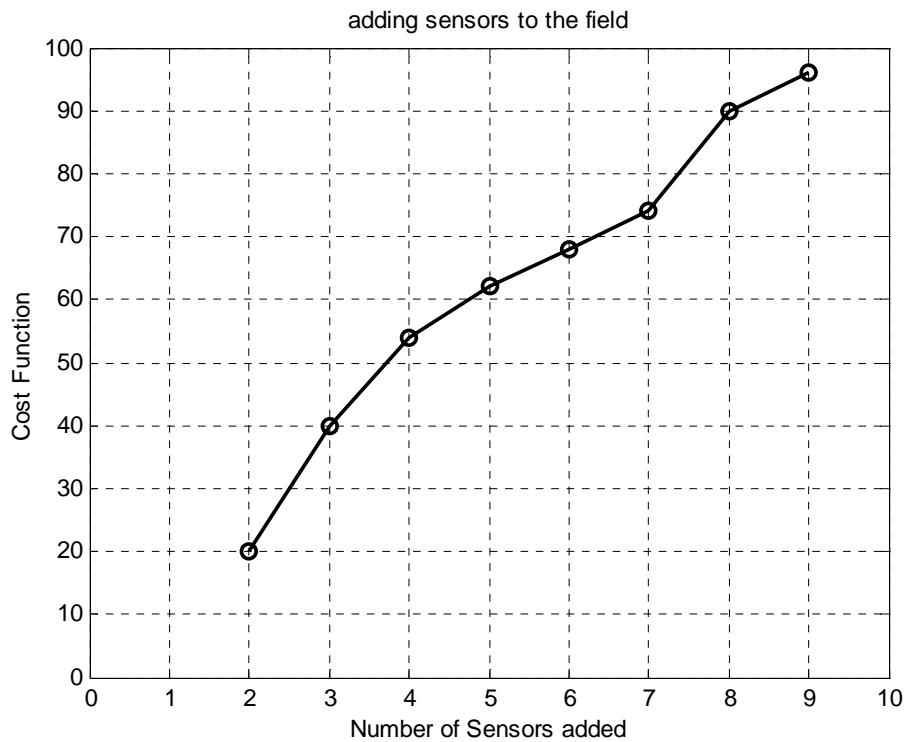


Figure 62: Cost function for increasing the number of sensors

Cost Analysis for Mobile Sensor Deployment

The cost of the hybrid sensor nodes can also be measured in terms of cost function by analyzing the varying factors that affect the deployment of the sensor nodes. For the 35 x 35 meter area of the sensor grid we arrive at the cost function using the following relation.

$$C_m(d) = C_m(n) + movn * movx - N_m - P_m \quad (40)$$

where $C_m(n)$ the coverage from n mobile sensors, $movn$ cost of moving n mobile sensors; $movx$ of each mobile sensor; N_m the signal-to-noise ratio for the n mobile sensors, P_m probability of detection of n mobile sensors. Figure 63 shows the cost of deployment for mobile sensor nodes for 9 sensors. It can be seen that the cost function is a minimum for 5 mobile sensors which implies that if we decide to make to develop a hybrid sensor grid with static and mobile sensors for our area of 35 x 35 meter we can make a maximum of 5 sensor nodes mobile out of 9 in order to maximize coverage and minimize the cost of deployment. This is a very significant hypothesis as it helps researchers determine the cost of adding mobile sensors to a specific sensor grid area to optimize the cost of deployment and coverage.

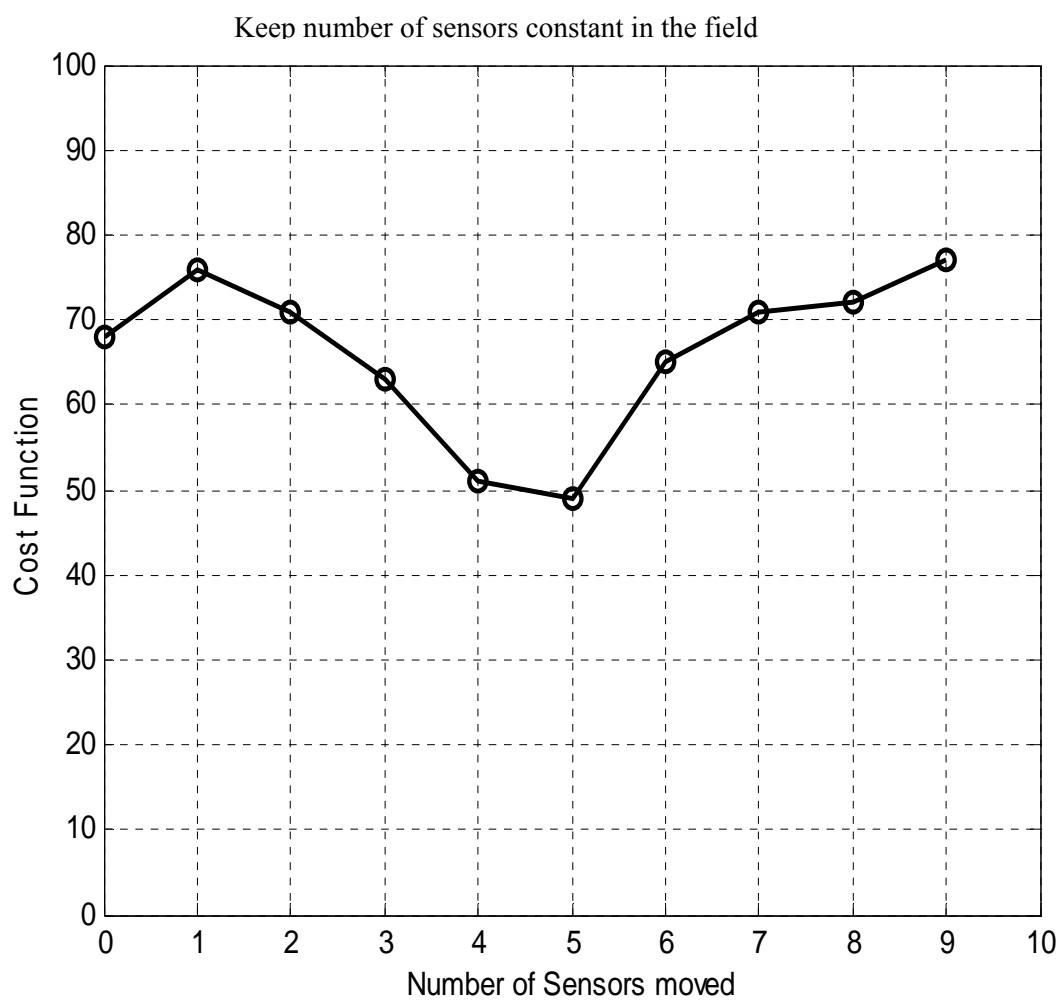


Figure 63: Cost of Deployment for Mobile sensor node

CHAPTER FIVE: CONCLUSION

A hybrid sensor network consisting of static and mobile sensor nodes was analyzed for effectively tracking a target in real-time with minimum latency. Various techniques to improve the target tracking within the sensor grid were addressed. These include using multiple frequencies, transmit power levels and filtering. It was shown that using larger number of reference nodes increases the accuracy of the system but resulted in increased latency. We also quantified the cost of deploying multiple sensors to cover a specific area of interest. Based on using this analysis other users can perform a cost budget analysis to determine the number of sensors, static and mobile needed to cover a given sensing area. The results presented in this dissertation can be used for multiple civilian and military applications such as tracking fire-fighters in buildings, setting up a virtual fence to protect vital installations and also track cargo including shipping containers. Future work will include increasing the number of sensor nodes and the sensing area such that large scale systems can be tested before deployment. One of the immediate issues in these simulations is the time taken to process the data through all the nodes. One possible solution will be to use parallel algorithms on High Performance Computing (HPC) systems for processing the data in parallel.

REFERENCES

- [1] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. 2002. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international Workshop on Wireless Sensor Networks and Applications* (Atlanta, Georgia, USA, September 28 - 28, 2002)
- [2] Aleksandar M, Chris O, Emil J, Wireless sensor networks for personal health monitoring: Issues and an implementation, *Computer Communications*, Volume 29, Issues 13-14, Wireless Sensor Networks and Wired/Wireless Internet Communications, 21 August 2006, Pages 2521-2533.
- [3] Akyildiz I F., Su W, Sankarasubramaniam Y, Cayirci E, Wireless sensor networks: a survey, *Computer Networks*, Volume 38, Issue 4, 15 March 2002, Pages 393-422, ISSN 1389-1286
- [4] Shnayder, V., Hempstead, M., Chen, B., Allen, G. W., and Welsh, M. 2004. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international Conference on Embedded Networked Sensor Systems* (Baltimore, MD, USA, November 03 - 05, 2004).
- [5] Landsiedel, O.; Wehrle, K.; Gotz, S.; , "Accurate Prediction of Power Consumption in Sensor Networks," *Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on* , vol., no., pp. 37- 44, 30-31 May 2005
- [6] Joseph A. Paradiso, Thad Starner, "Energy Scavenging for Mobile and Wireless Electronics," *IEEE Pervasive Computing*, pp. 18-27, January-March, 2005
- [7] Dissanayake, M.W.M.G.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M.; , "A solution to the simultaneous localization and map building (SLAM) problem," *Robotics and Automation, IEEE Transactions on* , vol.17, no.3, pp.229-241, Jun 2001
- [8] R. Peterson and D. Rus, "Interacting with sensor networks," presented at Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, 2004.
- [9] Y. Liuqing and G. B. Giannakis, "Ultra-wideband communications: an idea whose time has come," *Signal Processing Magazine, IEEE*, vol. 21, pp. 26-54, 2004.
- [10] V. P. Thomas Clouqueur, Parameswaran Ramanathan, Kewal K. Saluja, "Sensor Deployment Strategy for Target Detection," *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002.

- [11] S. B. Andrew Jamieson, Paddy Nixon Duncan Smeed, "MiPOS - the Mote Indoor Positioning System," presented at International Workshop on Wearable and Implantable Body Sensor Networks, Imperial College, United Kingdom, 2004.
- [12] G. V. Cesare Alippi, "A RSSI-based and calibrated centralized localization technique for Wireless Sensor Networks," presented at Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), 2006.
- [13] K. W. D. Li, Y. Hu, A. Sayeed,, "Detection, classification and tracking of targets in distributed sensor network," *IEEE Signal Processing Magazine*, vol. 19, pp. 17-29, 2002.
- [14] D. Xiaojiang and L. Fengjing, "Improving routing in sensor networks with heterogeneous sensor nodes," presented at Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st, 2005.
- [15] G. Song, W. Zhuang, Z. Wei, and A. Song, "Racemote: A Mobile Node for Wireless Sensor Networks," presented at Sensors, 2006. 5th IEEE Conference on, 2007.
- [16] Z. Yin, H. Zhao, K. Lin, P. Sun, Y. Gong, Y. Zhang, and Y. Xu, "Multi-sensor Data fusion in wireless sensor networks," presented at Computational Engineering in Systems Applications, IMACS Multiconference on, 2006.
- [17] D. A. Walquist, "Sensor data fusion in a simulated sensor environment," presented at Aerospace Conference, 2005 IEEE, 2005.
- [18] K. E. B. Andrew M. Ladd, Algis Rudys, Guillaume Marceau, "Robotics-based location sensing using wireless ethernet," presented at Proceedings of the 8th annual international conference on Mobile computing and networking, Atlanta, Georgia, USA 2002.
- [19] D. Min, A. Terzis, I. J. Wang, and D. Lucarelli, "Multi-Modal Calibration of Surveillance Sensor Networks," presented at Military Communications Conference, 2006. MILCOM 2006, 2006.
- [20] Konrad Lorincz and Matt Welsh, MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking, In *Proceedings of the International Workshop on Location and Context-Awareness (LoCA 2005)* at Pervasive 2005, May 2005.
- [21] Homayoun Hashemi, "The Indoor Radio Propagation Channel", Proceedings of the IEEE, Vol 81, No. 7, 1993.
- [22] J. Cort'es, S. Mart'inez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243-255, 2004.

- [23] Sonia Martinez, "On Optimal Sensor Placement and Motion Coordination for Target Tracking", Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona, Spain, April 2005
- [24] Tatiana Bokareva, Wen Hu, Salil Kanhere, Branko Ristic, Neil Gordon, Travis Bessell, Mark Rutten and Sanjay Jha, "Wireless Sensor Networks for Battlefield Surveillance", Land Warfare Conference 2006 Brisbane.
- [25] P. Enge and P. Misra. Special issue on GPS: The Global Positioning System. Proceedings of IEEE, 87(1):3-172, January 1999
- [26] N. Bulusu, J. Heidemann, D. Estrin, "GPS-less low cost outdoor localization for very small devices", in IEEE Personal Communications Magazine, Special Issue on Smart Spaces and Environments, 2000
- [27] G. Schiavone, P. Wahid, E. Van Doorn, R. Palaniappan, and J. Tracy, "Target detection and tracking using a UWB sensor web," presented at Antennas and Propagation Society International Symposium, 2004. IEEE, 2004.
- [28] R. P. Hong N. Nguyen, Nevin Aydinz, Shiyuan Jin, Damla Turgut, "Intrusion Detection in Wireless Sensor Networks," presented at World Congress in Computer Science, Computer Engineering, and Applied Computing, Las Vegas, 2007.
- [29] Phuong-Thao Pham, US Army RDECOM, R. Palaniappan, L. Mangold, J. Tracy, A. Wheeler , "Development of Low cost mobile platforms for indoor tracking navigation and geo-location", ITEA Modeling and Simulation Conference, 2005 Dec 12-15, Las Cruces, NM
- [30] M. F. Mysorewala, D. O. Popa, V. Giordano, and F. L. Lewis, "Deployment Algorithms and In-Door Experimental Vehicles for Studying Mobile Wireless Sensor Networks," presented at Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2006. SNPD 2006. Seventh ACIS International Conference on, 2006.
- [31] S. Gezici, T. Zhi, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, pp. 70-84, 2005.
- [32] G. Schiavone, P. Wahid, R. Palaniappan, J. Tracy, E. van Doorn, and B. Lonske, "Outdoor propagation analysis of ultra wide band signals," presented at Antennas and Propagation Society International Symposium, 2003. IEEE, 2003.
- [33] Lawrence, P.J., Jr.; Berarducci, M.P.; , "Navigation sensor, filter, and failure mode simulation results using the distributed Kalman filter simulator (DKFSIM)," *Position Location and Navigation Symposium, 1996., IEEE 1996* , vol., no., pp.697-710, 22-26 Apr 1996

- [34] V. Dua, T. N. Krishnan, and U. Mukherji, "Analysis and simulation of a multiaccess scheme over flat slow fading channels," presented at Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE, 2000.
- [35] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 42, pp. 1905-1915, 1994.
- [36] P. J. Escamilla-Ambrosio and N. Mort, "A hybrid Kalman filter-fuzzy logic architecture for multisensor data fusion," presented at Intelligent Control, 2001. (ISIC '01). Proceedings of the 2001 IEEE International Symposium on, 2001.
- [37] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2 ed: Springer, 1998.
- [38] Wadhwa, M.; Min Song; Rali, V.; Shetty, S.; , "The impact of antenna orientation on wireless sensor network performance," *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on* , vol., no., pp.143-147, 8-11 Aug. 2009
- [39] Reynaud, S.; Cocheril, Y.; Vauzelle, R.; Reineix, A.; Aveneau, L.; Guiffaut, C.; , "Influence of an accurate environment description for the indoor propagation channel modelling," *Wireless Technology, 2005. The European Conference on* , vol., no., pp.51-54, 3-4 Oct. 2005
- [40] Dominik Lieckfeldt, Dirk Timmermann, "Using Cramer-Rao-Lower-Bound to Reduce Complexity of Localization in Wireless Sensor Networks", Baltic Conference Advanced Topics in Telecommunication, ISBN: 978-3-86009-022-0, Riga, Lettland, 2007
- [41] Chengshan Xiao; Zheng, Y.R.; Beaulieu, N.C.; "Statistical simulation models for Rayleigh and Rician fading," *Communications, 2003. ICC '03. IEEE International Conference on* , vol.5, no., pp. 3524- 3529 vol.5, 11-15 May 2003
- [42] Urkowitz, Harry; , "On Detection and Estimation of Wave Fields for Surveillance," *Military Electronics, IEEE Transactions on* , vol.9, no.1, pp.44-56, Jan. 1965
- [43] Levis, P.A., "TinyOS: An Open Operating System for Wireless Sensor Networks (Invited Seminar)," *Mobile Data Management, 2006. MDM 2006. 7th International Conference on* , vol., no., pp. 63- 63, 10-12 May 2006
- [44] Bassem R. Mahafza and Atef Z. Elsherbeni, "MATLAB Simulations for Radar Systems Design", Chapman & Hall 2004
- [45] Liu, D.; Ning, P.; Du, W.K.; , "Attack-resistant location estimation in sensor networks," *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on* , vol., no., pp. 99- 106, 15 April 2005

[46] Welch, G. and Bishop, G. 1995 *An Introduction to the Kalman Filter*. Technical Report. UMI Order Number: TR95-041., University of North Carolina at Chapel Hill.

[47] D. Simon, "Using Nonlinear Kalman Filtering to Estimate Signals," *Embedded Systems Design*, vol. 19, no. 7, pp. 38-53, July 2006