

**Application of Mobility Models to Collaborative
Search and Rescue Robotics.**

by

Nida Fatima Bano

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 06,2011

Keywords: Collaborative robotics, Mobility models,
Random Way Point model, Manhattan model, Search and Rescue.

Copyright 2011 by Nida Fatima Bano

Approved by

Thaddeus Roppel, Chair, Associate Professor, Electrical and Computer Engineering
Prathima Agrawal, Samuel Ginn Distinguished Professor, Electrical and Computer Engineering
Shiwen Mao, Assistant Professor, Electrical and Computer Engineering

Abstract

This thesis addresses the challenges of using small, low-cost robotic nodes with limited processing capabilities in collaborative search and rescue. Mobility models tailored specifically to the mobility patterns of such nodes are developed and tested. Mobility models have traditionally been used to represent movement patterns in Ad-hoc network devices like cell phones, laptops, etc. The study of these models provides invaluable information on network performance and traffic analysis. As more and more applications of robotics employ the use of wireless protocols, there is a need for mobility models better suited to the use of communicating mobile robotic nodes. This thesis gives an overview of traditional models and proposes new models befitting the mobility pattern of a number of robotic nodes collaborating with each other to perform search and rescue. It is shown that slight variation in the conventional models can lead to the development of realistic movement patterns by incorporating obstacles and communication between the nodes.

Acknowledgments

I am deeply indebted to Dr. Thaddeus Roppel for giving me confidence to explore my research interests and the guidance to avoid getting lost in my exploration. Dr. Roppel was a great advisor. A special thanks to Dr. Prathima Agrawal for her constant support and invaluable advice. Her financial support gave me the peace of mind to focus entirely on my thesis. I would also like to thank Dr. Shiwen Mao for his guidance and acceptance to serve on my committee.

My colleagues at Broun 405 made sure I was constantly engaged and entertained along the way. Yogesh, Nirmal, Santosh, Gopal and Pratap, I thank all of them for their encouragement and friendship. Outside of the lab, many people kept me happy and sane at Auburn. A special mention and thanks to my roommate, Madhumidha for making life away from home so much more comfortable. Jennifer Wang, for stopping by just to get me out of the lab. To all my friends at Auburn; thank you. You made my stay memorable.

A big thank you to my friends: Ahmed- for his sound technical advice and encouragement, Namrata- for always lending a patient ear to all my complaints, Karishma- for having confidence in me when I had almost lost hope and Hafsa- for her unwavering support. Despite the geographical distance my family was always near. I am forever indebted and grateful to my father, mother, Api and Kulsum for their unwavering faith, love and support. Most importantly, I thank GOD for making me capable of this and blessing me with the best family and friends.

This work was supported in part by the National Science Foundation under grant IIP-0738088.

Table of Contents

Abstract.....	ii
Acknowledgments	iii
List of Figures.....	vi
Chapter 1 Introduction	1
Chapter 2 Literature Review	5
2.1 Inception of Collaborative Robotics.....	5
2.2 Simulated Environments	6
2.3 Mobility Models	7
2.4 Collaborative Robotics	8
2.5 Collaborative Robotics for Search and Rescue	10
2.6 Cooperative Robotics Research at Auburn University.....	10
2.6.1 Laboratory setup.....	10
2.6.2 Robot Architecture	11
2.6.3 Cooperative Search.....	12
2.6.4 Communication Network	12
Chapter 3 Mobility Models and Robot Swarm Algorithms	13
3.1 Mobility Models	13
3.1.1 Random Walk Mobility Model	15

3.1.2 Random Way Point Mobility Model.....	17
3.1.3 Random Direction Mobility Model.....	18
3.1.4 City Section Mobility Model.....	19
3.1.5 Manhattan Mobility Model.....	20
3.2 Swarm Robotics.....	21
3.2.1 Biological Inspiration.....	22
3.2.2 Ant Colony Optimization.....	23
3.3 Virtual Pheromones.....	24
3.3.1 Attraction and Repulsion forces between Robots.....	24
3.3.2 Payton’s Guided Growth Models.....	25
3.3.2.1 Gas Expansion Model.....	25
3.3.2.2 Biologically Inspired Guided Growth Model.....	25
Chapter 4 Robot Architecture and Simulation Models Designed.....	27
4.1 Laboratory Setup.....	27
4.2 Robot Architecture.....	28
4.2.1 Chassis.....	28
4.2.2 Drive System.....	28
4.2.3 Sensors.....	29
4.2.4 Control Systems.....	30
4.2.5 Batteries.....	30
4.3 Simulation Model Fundamentals.....	31
4.3.1 Boundary Conditions.....	31
4.3.2 Obstacle Avoidance.....	31

4.3.3 Communication	31
4.3.4 Repulsive Forces between Robots.....	32
4.4 Simulation Mobility Models Developed	32
4.4.1 Random Mobility without Communication and no Obstacle Avoidance.....	33
4.4.2 Random Mobility with Communication and Obstacle Avoidance.....	35
4.4.3 Manhattan Mobility with Obstacle Avoidance and no Communication.....	35
Chapter 5 Results	38
5.1 Simulation Setup	38
5.2 Simulation Parameters.....	39
5.3 Results and Analysis.....	39
5.3.1 Target Mobility.....	40
5.3.2 Varying Communication Intervals	42
5.3.3 Obstacle Avoidance.....	44
5.3.4 Varying Number of Obstacles	45
5.3.5 Varying size of Robots	46
5.3.6 Increasing Maximum Speed.....	47
Chapter 6 Conclusions and Suggestions for Future Work	49
6.1 Conclusions	49
6.2 Suggestions for Future Work.....	50
References	51

List of Figures

Figure 1.1 Underwater Robots attempting to activate a blowout preventer	3
Figure 2.1 Connex 400xm-bt motherboard and wifistix used for communication.....	11
Figure 3.1 Classification of Mobility Models.....	14
Figure 3.2 Travelling Pattern of a node using Random Walk mobility model.....	16
Figure 3.3 Travelling Pattern of a node using Random Way Point mobility model.....	17
Figure 3.4 Travelling Pattern of a node using Random Direction mobility model.....	18
Figure 3.5 City Section Mobility Model, Nodes take multiple steps to travel from A to B.....	19
Figure 3.6 Topography showing the movement of nodes for Manhattan mobility model.....	20
Figure 3.7 Travelling Pattern of a node using Manhattan mobility model.....	21
Figure 3.8 Zones surrounding a robot implementing pheromone robotics.....	24
Figure 3.2 A “bud” acts as a single point until it hits a dead-end, then a new one takes over.....	26
Figure 4.1 The hardware test-bed.....	28
Figure 4.2 IR sensor turret assembly.....	29
Figure 4.3 Batteries and mounted Optical mice.....	30
Figure 4.4 Travelling Pattern of a node using Random Way Point mobility model.....	33
Figure 4.5 Simulation setup for the Random mobility model with no obstacle avoidance and no inter-node communication.....	35
Figure 4.6 Simulation setup for the Random mobility model with obstacle avoidance and internode communication.....	36

Figure 4.7 Travelling Pattern of a node using Manhattan Mobility model.....	37
Figure 4.8 Simulation setup for Manhattan Mobility Model with obstacle avoidance and no internode communication.....	37
Figure 5.1: Mobility follows the RWP model. The frequency of communication is varied from none to continuous for a mobile target.. ..	41
Figure 5.2: Mobility follows the RWP model. The frequency of communication is varied from none to continuous for a stationary target.....	41
Figure 5.3: Both RWP mobility and Manhattan mobility models represented. No communication between nodes. Target motion is varied between stationary and mobile.....	42
Figure 5.4: Mobility follows the RWP model. The communication interval is varied from none to occasional to continuous.....	43
Figure 5.5: Mobility follows the RWP model. Obstacles are included in the simulation at random locations.....	44
Figure 5.6: Number of obstacles is increased. Nodes follow RWP mobility model.....	46
Figure 5.7: The radius of the robot is varied while the repulsion radius is kept constant.....	47
Figure 5.8: The maximum speed is increased and the effect on average speed is studied.....	48

CHAPTER 1

INTRODUCTION

“...the robots helped identify five victims, and transmitted many detailed videos and photos that could have proven invaluable had there been survivors.” [1]

“...BP used robots in its latest attempt to curtail the flow of crude from the largest spill in U.S. history, which spread to barrier islands off Alabama and Mississippi on Tuesday.” [2]

Mobile robotics generally involves un-tethered robots that run on batteries and have an on-board computer. Mobile robots have a plethora of applications ranging from carpet cleaning [3] to inter-planetary exploration (Mars Rover). When wireless technology is added to mobile robotics, it opens the door to the possibility of collaborative behavior between multiple mobile robots. The research described in this thesis is focused on novel mobility models for communicating robots. These nodes are ultimately intended for search and rescue operations in disaster scenarios.

Collaborative robotics is a term used to define a group of robots interacting with each other to accomplish a prescribed task. Using multiple robots to perform a given task can have advantages compared with single robot systems [4,5]. Collaborating robots may be able to carry out a task in much less time than a single robot. For instance, a system of collaborative robots outperformed several single robot systems designed to carry out the same task [6]. A multiple robot system might also be able to perform better localization if the robots exchange location

each other [7]. Such multi-agent systems also reduce the possibility of single point failures. This topic has been under research for almost three decades. Most of the initial research was done to assist in collecting environmental data with the help of sensor nodes. The collected data were relayed to a base station using wireless protocols. Numerous advances have been made to this area of research since then and many ‘collaborating robots’ are now in practical use.

Collaborating robots can prove to be invaluable for the development of a search and rescue team. Many algorithms have been developed in the past using multiple robotic nodes. A team of autonomous robots searches for a stationary target in [8]. They use wireless communication to build and share collective maps of the environment. Since the robots were low-cost and had limited processing capabilities, a large amount of path planning and navigation was off-loaded to a centralized server. The processed data (maps) were then broadcast to all the robots. This increased the processing capabilities and helped in quicker and better results to find the target sooner, but was not practical for an implementation outside the laboratory..

The first use of mobile robots in an actual human-search-and-rescue mission was in the aftermath of the attacks on the World Trade Center [9]. Since then collaborating robots have been used in a number of search and rescue missions in both man-made and natural calamities. For instance: on May 1 2010, a robot was employed to recover an undetonated bomb in Times Square, New York. Bomb-disposal robots are tasked with not only removing the bombs, but getting rid of them through disablement or detonation, while humans keep a safe distance. Another recent example of the application of robots in the event of catastrophe is the use of remote-operated vehicles (ROV) in the face of the BP oil spill. ROVs were put to work

examining and attempting to activate the malfunctioning blowout preventer, a device that sits atop the wellhead and is supposed to seal off the well in the event of a blowout. The ROVs successfully inserted a riser insertion tube tool, which is now capturing about 2,000 barrels of oil per day that would otherwise have been leaked into the sea.



Figure 1.1: Underwater robots attempting to activate a blowout preventer.[10]

Exploring an area effectively is one of the key factors in collaborative search-and-rescue. One of the major constraints in the design of search and rescue algorithms is time. It is necessary to ensure the robots search as much area in as little time as possible, i.e. achieve maximum exploration in minimum time. Care must be taken to ensure that the same area is not explored twice. Eliminating the presence of a central processing unit and enabling communication between nodes will considerably reduce the time to completion of the search task. Having pre planned or predictable motion is another factor that would help reduce the time required to search an area for a target.

A number of reinforcement and learning techniques are available in literature and all of these mainly employ autonomous robots with advanced processing capabilities. A real-life rescue mission was carried out by 14 researchers and a group of search and rescue robots in Lebanon, Indiana. Center for Robot-Assisted Search and Rescue, CARSAR and Indiana Task Force 1, one of 28 urban search-and-rescue task forces established by the Federal Emergency

Management Agency (FEMA), organized the training exercise with support from the National Science Foundation (NSF). This exercise was conducted by simulating an earthquake and was designed to elicit research issues and practical matters that robot developers might overlook or take for granted [11]. One of the findings of this exercise was that robots didn't need to be autonomous. Researchers learnt that the robots just need to be able to respond to one or two simple commands and to give a signal. There is thus a need to design effective and efficient search and rescue algorithms for robotic nodes with limited processing capability.

The work described in this thesis proposes the use of specially-tailored mobility models in a search and rescue algorithm for use by low-cost and simple collaborating robotic nodes.

Mobility models have been very widely simulated to analyze routing protocols and network analysis of mobile ad-hoc networks (MANETs). To study the performance of MANET protocols, it is essential that we employ mobility models that closely imitate the nodes' movement patterns. Analysis in this work shows that slight variation like incorporating obstacles and communication between the mobile nodes in conventional mobility models can lead to realistic models better suited to mobile robotic nodes.

The remainder of this thesis is organized as follows. Chapter 2 discusses a literature review of work related to the discussion in this thesis. Chapter 3 helps gain insight into a few conventional mobility models available in simulation and swarm algorithms in use with reference to collaborative robotics. Chapter 4 discusses the robot architecture and the mobility models designed and simulated. The simulation setup, an analysis of the simulated results and discussions on the same are presented in Chapter 5. Conclusions and suggestions for future work are in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

There is a significant increase in the interest for collaborative robotics as a viable alternative to the more centralized classic approach as energy consumption and reliability (link failure) are becoming important constraint factors. Therefore, this ever-growing field has become a diverse research area that often seems to go in several different directions at once. In the past twenty years numerous topics of interest have emerged, each making a significant amount of contribution to this field.

This chapter introduces the primal areas of research being carried out in the field of collaborative robotics that pertain to the work presented in this thesis.

2.1 Inception of Collaborative Robotics:

Collaborative robotics is a relatively new field of study emerging in the late 1980's when researchers started studying the possibilities of using multiple mobile robot systems. Most of the initial research was done to assist in collecting environmental data with the help of sensor nodes and which would relay the information gathered to a base station using wireless protocols.

In 1987, Fakuda and Nakagawa [12] laid the foundation for a dynamically reconfigurable robotic system that could autonomously reconfigure itself based on the task it was to perform. These systems consist of robotic "cells" that communicate with each other and configure themselves based on the objective of their task. These cells though simple and basic on their own, exhibited complex and new mannerisms when combined in groups.

This field of study gained a lot of popularity due to its application in hazardous situations and the prospect of efficient performance of tasks. Numerous advances have been made to this area of research and many ‘collaborating robots’ are now in practical use.

2.2 Simulation Environments:

Simulation is an important tool in wireless networking research. Simulations are easier to replicate, modify and are very portable compared to physical experiments. Current simulators however, are unable to model many essential characteristics of the real-world [13], [14], [15], [16]. When deploying mobile nodes, real-world experiments are rare. To obtain a more realistic approach, we looked into the use of testbeds available for mobile wireless research. The need was for a testbed that contained actual mobile nodes, provided means for programming the devices, ensure accurate motion and ease collection of the experimental data. Such a testbed should also be available online and be accessible to remote users. Many of these testbeds made use of the Network Simulator-Ns. Ns is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks [17].

A number of such options are available to researchers. MiNT[18] is a miniature 802.11 testbed. Its focus is on reducing the area required for a multihop 802.11 testbed, and on integrating network simulator-ns simulation with emulation. It achieves mobility through the use of antennas mounted on Lego Mindstorm robots, tethered to a PC running the applications. The robot mobility is limited. The ORBIT [19] testbed provides a remotely accessible 64-node indoor grid with 1m spacing, that can emulate mobility. User code is run on PCs, which are dynamically bound to radios. By changing the binding of a PC to a radio, the PC appears to move in discrete hops. Mobility is constrained in this testbed as well.

The Mirage testbed [20] is complementary to all of the other testbeds in that it uses resource allocation in a sensor net testbed as a target to explore new models of resource allocation, e.g., market-based models.

2.3 Mobility models

Due to the emergence of new and advanced wireless and cellular technology, a lot of attention has been focused on the development and evaluation of protocols for ad hoc networks with mobile devices [21]. Various mobility models have been proposed for these networks. This section gives an overview of these movement models designed specifically for ad hoc networks. There are two types of mobility models used in the simulation of networks currently: Traces and Synthetic Models [22]. Traces are mobility patterns observed in real-life systems.

They involve a large number of participants and an appropriately long observation period. Ad hoc networks however, are not easily modeled if traces have not yet been created. Synthetic models, on the other hand, attempt to realistically represent the behaviors of mobile nodes without the use of traces. A mobility model should be able to closely mimic the movement of nodes in real scenarios. Changes in parameters such as speed and direction must occur realistically and in reasonable time slots.

A variety of mobility models have been proposed for ad hoc networks [23, 24, 25, 26, 27] and a survey of many is presented in [28, 29]. These models vary widely in their movement characteristics. In the Random Walk mobility model described in [30], nodes select a direction in which to move, between 0 and 2π , a speed from a given distribution, and then move in that direction at that speed for either a specified number of steps or for a time period. At the end of this period, the nodes repeat this process. The Random Direction model [30, 31] operates

similarly to the Random Walk, except that nodes continue to walk until they are within some ϵ of the simulation boundary. Once they reach this area, they select a new direction in which to walk.

One of the most popular mobility models for studying ad hoc networking protocols is the Random Waypoint model. In this model, a node selects a random destination within the network, a speed from a distribution, and then moves to the selected destination at the selected speed. Once the node reaches the destination, the node rests for a pause time, and then repeats the process by selecting a new destination and speed and resuming movement.

Some attempts have been made to design mobility models to better suit robot movements. As mentioned in [32] robot motion is driven by the robot's task and usually little attention is given to the impact of the robot's behavior on the communication network. In [33], the mobility patterns exhibited by autonomous robots roaming in a confined region are compared to random mobility models traditionally used for representing mobile device user behavior. The authors incorporate obstacles and repulsive forces to avoid collision. Although this approach helps realize a more realistic approach to robot mobility, it is not designed to help develop a search and rescue algorithm. However, results obtained in [33] help establish a better understanding of how to develop mobility models representative of robot movements.

2.4 Collaborative Robotics:

Reinforcement learning has been extensively studied [34, 35] to enable cooperative behavior among robots. Reinforcement learning algorithms are used to achieve cooperation and enable concurrent learning in robots. Machine learning techniques have been proposed and studied for multi robot systems. A reinforcement learning controller and behavior based control networks to help in multi robot concurrent learning by using a reward system is implemented by the authors of [36]. The control network is behavior based and is created according to human

knowledge of robot, its environment and its mission. To enable learning of cooperative behaviors, four kinds of rewards have been used: two positive to encourage ‘good behavior’ and two negative to discourage a bad decision. This algorithm has however not been implemented on actual machines.

Vision-based localization to enable accurate positioning is also another very popular field of study that can prove invaluable to collaborative search and rescue by robots. Sim and Dudek [37] combine the benefits of image and landmark-based learning methods and use an attention mechanism to detect potential landmarks. The authors use image landmarks to perform position estimation. These landmarks are however learned from an offline mapping phase. Localization is a two-step process consisting of an off-line preprocessing stage and an on-line estimation stage. A duplication of the surroundings in the form of database comprises the offline stage. This is later used for positioning. The on-line stage is used to accurately match observed landmarks to stored ones. These matches are then used to compute position estimates. Researchers have also attempted to make use of SLAM-Simultaneous Localization and Mapping, wherein the robot strives to determine its position in the environment and simultaneously attempts to build a map of its surroundings using a stereo-vision system [38]. The author uses SIFT – (Scale Invariant Feature Transforms) algorithm to process the images gathered from the environment and the Rao-Blackwellized particle filtering algorithm to estimate the ego-position of the robot from the collected images.

SARA-1employs six robotic nodes configured as an ad hoc network [39]. Robots search for a stationary target and use wireless communication to build and share collective maps of the environment. Since the robots were low-cost and had limited processing capabilities, a large

amount of path planning and navigation was off-loaded to a centralized server. The processed data (maps) was then broadcast to all the robots.

2.5 Collaborative Robotics for Search and Rescue

Various attempts have been made to use collaborating robots for Search-and-Rescue. Jennings et. al. present two large robots searching for and ‘rescuing’ warehouse-like boxes [40]. The search-phase however, occurs in a single, obstacle-free room and involves no intentional cooperation.

Vainio *et. al.* [41] present a control architecture for a group of underwater search-and-destroy robots. While this experiment makes great use of unintentional cooperation to provide speed-up over a single-robot, more general applications would benefit from direct communication and intentional cooperation. The first actual search and rescue by robots was performed in the aftermath of the attacks on the World Trade Center.

2.6 Cooperative Robotics Research at Auburn University

The work presented in this thesis is one of many collaborative projects of the Cooperative Robotics Research (CRR) Team at Auburn University. This section briefly describes several of the other projects that are currently underway. More information about these projects, including videos, can be found at [42].

2.6.1 Laboratory set-up:

A team of six robots has been implemented in the laboratory. The functionality includes on-board processing, mobility and wireless capability. The laboratory experiment is carried out in a grid of dimensions 16 ft × 8 ft ($\frac{1}{4}$ 5 m × 2.5 m), with boundaries of cardboard walls. The 5 m long aisle is flanked by three rooms on either side with openings to serve as doorways. A simple room scenario was designed for this project. The robots are deployed in predetermined positions at one end of the room. That is, the initial pose of the robot is assumed to be known.

Maps are exchanged thereafter to update each robot of the location of every other robot in the network. A stationary target cone is deployed at a random position when required. The robots perform a collaborative search of the environment for the stationary target. The robots use two 2800 mAh /7.2 V batteries for operation and are capable of motion at the rate of 0.5 inch per second ($\frac{1}{4}$ 0.0381 m/s).

2.6.2 Robot Architecture

On the hardware side, the robots are designed with the following capabilities. They consist of an on-board Linux OpenEmbedded platform built on a Gumstix Connex 400xm-bt motherboard based on the 400MHz Intel XScale PXA255 processor. The motherboard features a 64 MB SDRAM and 16 MB Flash Memory. Wireless connectivity is enabled by a wifistix expansion board that connects to the Connex motherboard via the 92-pin bus header. The wifistix FCC (with wifi antenna) uses standard 3.5V- 6V input. The on-board Marvell 88W8385 chipset provides IEEE 802.11a/b/g WLAN capability. (The 88W8385 enjoys embedded CPU and on-chip memory). Fig. 2.1 shows the equipment used in the laboratory. A more detailed description of the robot architecture is included in Chapter 4.

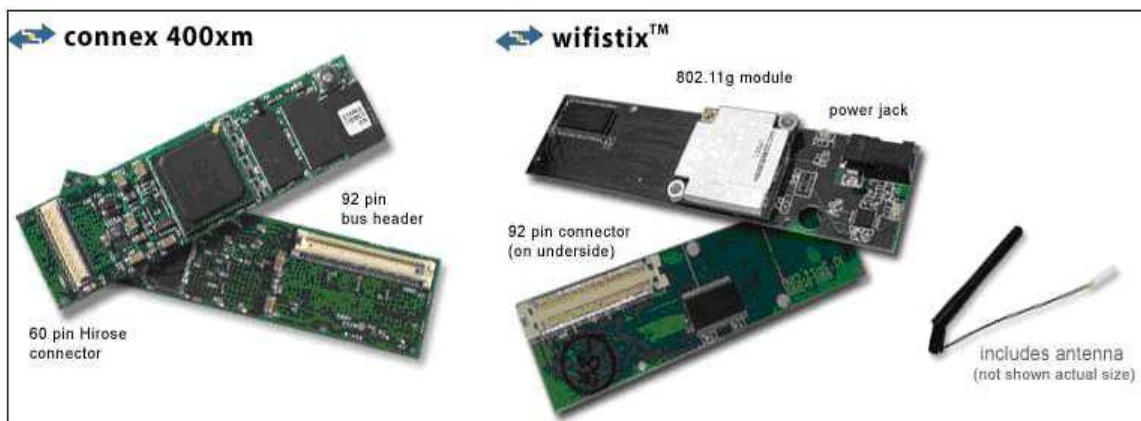


Figure 2.1: Connex 400xm-bt motherboard and wifistix used for communication

2.6.3 Cooperative Search :

The objective of the robot team is to search the area and locate the stationary target. Both exploration of the environment and location of the target require collaboration among the robots. The robots use SONAR to scan the area in order to identify and avoid obstacles (walls and fellow-robots). A search-and-rescue algorithm [39] was designed by the CRR team for this purpose.

2.6.4 Communication network

The six gumstix robots are part of a WLAN (Wireless Local Area Network) with the fixed server acting as Base Station or Access Point. Due to processing power constraints, the robots are currently designed to communicate with only the BS and not peer-to-peer. Collaborative robotics calls for regular communication and exchange of information between the robots. Since an ad-hoc design is more suitable for real-world collaborative search-and-rescue operation, an ad-hoc peer-to-peer network is used in the experimental simulations. When modeling a robotic sensor network, it is important to note the environmental aspects in terms of communication features required. Key characteristics of the above scenario can be summarized as follows:

- Limited number of robotic nodes.
- The nodes need to be fully mobile.

Keeping in mind that the eventual implementation would be in an outdoor environment, a more general rectangular area is considered for our experimental purposes.

CHAPTER 3

MOBILITY MODELS AND ROBOT SWARM ALGORITHMS

This research aims at examining practical movement patterns for a group of robots searching for a target. The first part of this chapter discusses the various mobility models available in literature and the latter analyzes a few algorithms used to disperse a large group of autonomous mobile robots efficiently throughout a bounded environment using local inter-robot sensing or communication.

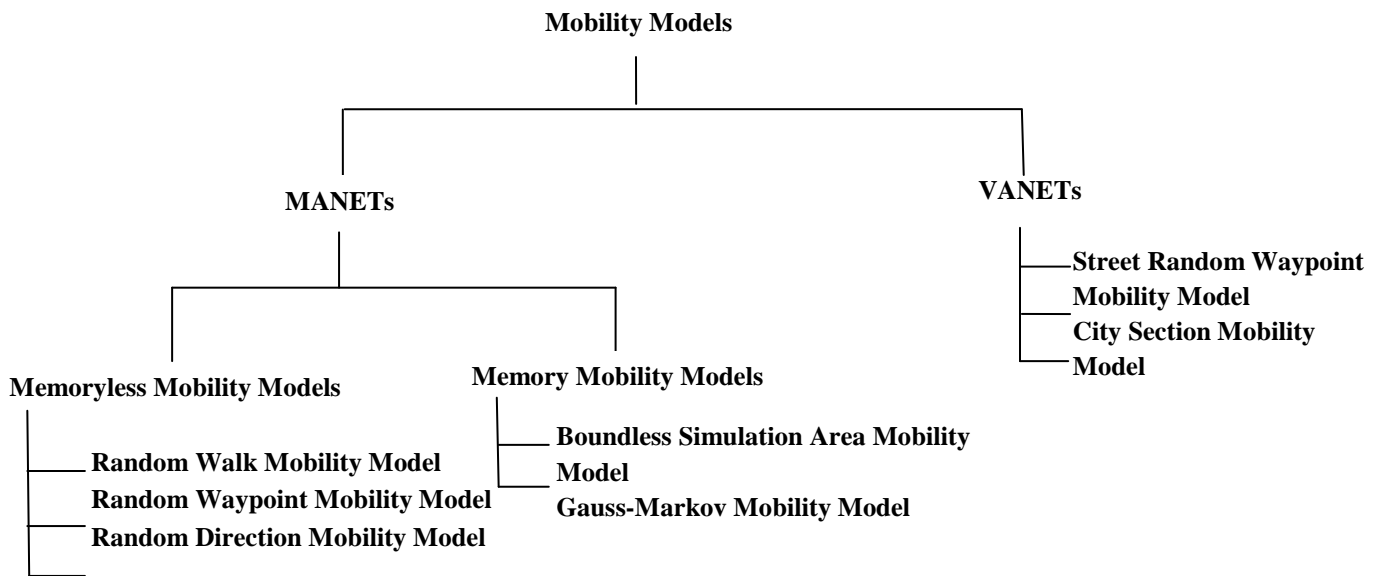
3.1 Mobility Models

There are mobility models available in simulations that incorporate movement in nodes adhering to certain mathematical rules. The initial deployment of nodes in the region may follow various mathematical distributions such as uniform, Gaussian, etc. Once the nodes have been distributed, their movement is governed by these mobility models.

Mobility models have been very widely simulated to analyze routing protocols and network analysis of Mobile Adhoc Networks; MANETs. In recent times much research has gone into using mobility models for Vehicular Adhoc Network; VANET simulation as well. To study the performance of MANET and VANET protocols, it is essential that we employ mobility models that closely imitate the nodes' movement patterns.

As mentioned in Chapter 1, there are two types of mobility models currently used in the simulation of networks: Traces and Synthetic models. Synthetic models can further be divided into memory and memoryless mobility models. Memory mobility models take into account

values gained by the different simulation parameters during a particular iteration and use them to deduce new values for those parameters during the next iteration. On the contrary, memoryless mobility models use random values to populate the parameters during every iteration in the simulation. The Gauss Markov Mobility Model is a prime example of a memory mobility model while the Random Walk Mobility Model, Random Waypoint Mobility Model and the Random Direction Mobility Model are all examples of the latter category. Figure 3.1 gives the



classification of the various mobility models available in simulation.

Figure 3.1: Classification of Mobility Models.

Node mobility is one of the inherent characteristic of a MANET. The memoryless mobility models are most commonly used. In most simulation experiments, node movement is modeled as an independent random walk. As seen from Figure 3.1, the most important random models are:

Random Walk Mobility Model (including its many derivatives): A simple mobility model based on random directions and speeds.

Random Waypoint Mobility Model: A model that includes pause times between changes in destination and speed.

Random Direction Mobility Model: A model that forces mobile nodes to travel to the edge of the simulation area before changing direction and speed.

Another set of commonly used mobility models is one used for Vehicular Ad-hoc Networks (VANETS). This includes:

Street Random Waypoint Mobility Model (STRAW): The STRAW (STreet RANdom Waypoint) mobility model addresses the issues of fading signal strength at the receiving nodes caused by street layouts and different obstructions [43]. This problem is handled by constraining node movement to streets defined by map data for real US cities and limits their mobility according to vehicular congestion and simplified traffic control mechanisms.

City Section Mobility Model: This model assumes the network to be divided into grids: square blocks comprised of horizontal and vertical streets. The direction of nodes is decided randomly.

Manhattan Mobility Model: The direction of nodes is decided probabilistically at every street intersection. In the Manhattan model, the mobile node is allowed to move along the horizontal or vertical streets on the urban map.

All of these models have been extensively modeled and simulated for study of network protocol analysis.

3.1.1 Random Walk Mobility Model:

Each node chooses a random speed and direction to move within a bounded region.

Given the minimum speed V_{min} and the maximum speed V_{max} , the random speed ($v(t)$) and the random direction ($\theta(t)$) are chosen based on the Uniform distribution U_{rand} .

$$v(t) = U_{rand}(V_{min}, V_{max}) \quad (1)$$

$$\theta(t) = U_{\text{rand}}(0, 2\pi) \quad (2)$$

Every node chooses a new set of values if one of the following two conditions is met: a pre-selected timer has timed out or a node reaches the boundary of the area of simulation.

In the event of a time out, a new set of values is selected and assigned. In the second case, the node bounces back with the same speed but varied angle. This is referred to as the border effect.

Figure 3.2 shows the movement pattern of a node using the Random Walk mobility model.

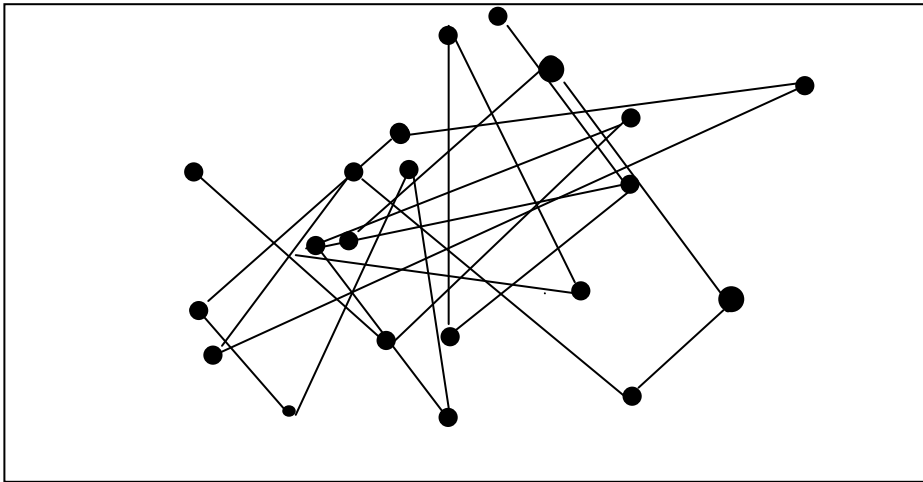


Figure 3.2 Travelling pattern of a mobile node using Random Walk mobility model

A node is initially given a random position on the grid. At each point, a node randomly chooses a direction between 0 and 2π and a speed between 0 and V_{max} . The node is then allowed to travel for T seconds before changing direction and speed.

A very popular modification of this model is the *Random Way Point Mobility Model*. This model introduces pauses between change in destinations and speed.

In our research, a group of 5-6 robotic nodes equipped with WiFi transceivers are configured as a mobile ad-hoc network (MANET). As these robots have to be employed to carry out search and rescue, it would be an advantage to have a predictable and pre planned mobility pattern. This would greatly help reduce the time and processing capabilities of the robots.

Mobility models have been used to a great extent in mobility planning for robots. The work described in [33] incorporates obstacles and repulsion forces to avoid collision. Although this approach helps realize a more realistic approach to robot mobility, it is not designed to help develop a search and rescue algorithm. In this work we try to include communication between the nodes and make use of obstacle avoidance to develop mobility models better suited to collaborative search and rescue robots.

Of the random mobility models mentioned before, the RWP mobility model is most commonly used in the study of MANETs. It is easy to implement, easy to derive analytical results and can be described using very few parameters.

3.1.2 Random Way Point Mobility Model

One of the most popular mobility models for studying ad hoc networking protocols is the RandomWaypoint model. It is the most elementary synthetic model. The analytical results obtained from the RWP model also help in choosing realistic values for the model parameters before the actual process simulation.

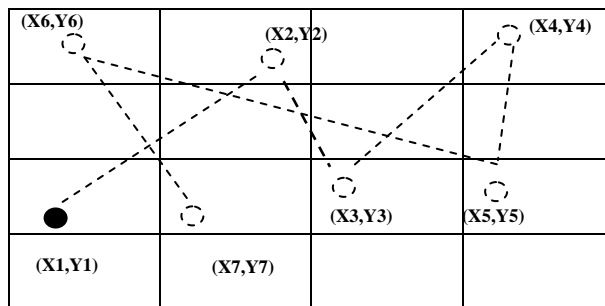


Figure 3.3 Travelling pattern of a mobile node using Random Way Point mobility model.

In this model, a node selects a random destination within the network, a speed from a distribution, and then moves to the selected destination at the selected speed. Once the node

reaches the destination, the node rests for a pause time, and then repeats the process by selecting a new destination and speed and resuming movement.

3.1.3. Random Direction Mobility Model:

The Random Direction Mobility Model [30,31] is similar to the Random Waypoint mobility model in that the robots calculate the speed and direction using the same equations mentioned in the earlier models. But instead of changing the node direction and speed after a constant time interval T , the nodes are forced to continue traveling with the same speed and direction until they encounter a boundary. Figure 3.3 shows the travelling pattern of a node following the Random Direction mobility model.

All of these random models are based on a mobility phenomenon called the ‘Brownian Movement’, a seemingly random movement of particles suspended in a fluid (i.e. a liquid such as water or air). It is not a very realistic movement pattern but is very easy to implement.

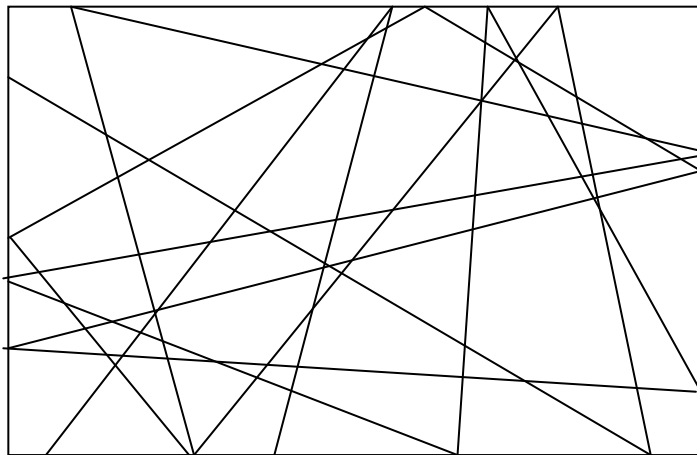


Figure 3.4 Travelling pattern of a mobile node using Random Direction mobility model

Mobility models for Vehicular ad-hoc Networks (VANETs):

A lot of research is being carried out in developing network protocols and services for vehicular ad-hoc networks (VANETs). Due in part to the prohibitive cost of deploying and implementing such systems in the real world, most research in this area relies on simulation for

evaluation. A key component of these simulations is a realistic vehicular mobility model that ensures conclusions drawn from such experiments will carry through to real deployments.

Unlike many other mobile ad-hoc environments where node movement occurs in an open field (such as conference rooms and cafés), vehicular nodes are constrained to streets often separated by buildings, trees or other objects. Street layouts and different obstructions increase the average distance between nodes and, in most cases, reduce the overall signal strength received at each node. We argue that a more realistic mobility model with the appropriate level of detail for vehicular networks is critical for accurate network simulation results. This section discusses two mobility models based on VANETs.

3.1.4 City section Mobility Model:

The *City Section* mobility model constrains nodes movement on a grid road topology, where all edges are considered bi-directional, single-lane roads. Vehicles randomly select one of the intersections as their destination over the grid and move towards it with constant speed, with (at most) one horizontal and one vertical movement [44]. The speed depends on the road the vehicle is traveling on: two road classes, high speed and low speed, are considered, and each node sets its speed to a high or low value accordingly. Car-to-car interactions are ignored, since all adjacent vehicles travel at the same speed, and vehicles are allowed to overlap at road junctions.

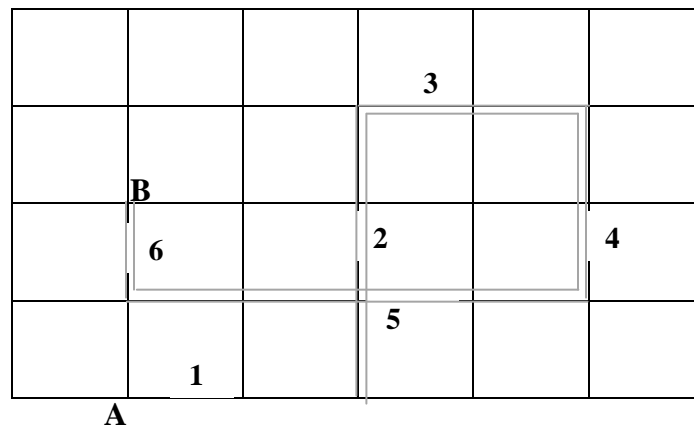




Figure 3.5 City section Mobility model. Nodes take multiple steps to travel from A to B.

No pauses at intersections or at the end of trips are specified. In Figure 3.5 an example of a vehicle (node) trip with the City Section model is provided.

3.1.5 Manhattan Mobility Model:

In this model, each node chooses a random direction along the grid when it reaches a junction and keeps moving in the chosen direction until it reaches the next junction. Figure 3.6 is example topography showing the movement of nodes for Manhattan Mobility Model with nine nodes. The map defines the roads along the nodes can move. Figure 3.7 represents Manhattan model topology, each line representing a single-lane road. Node movement occurs on the direction shown by the arrows

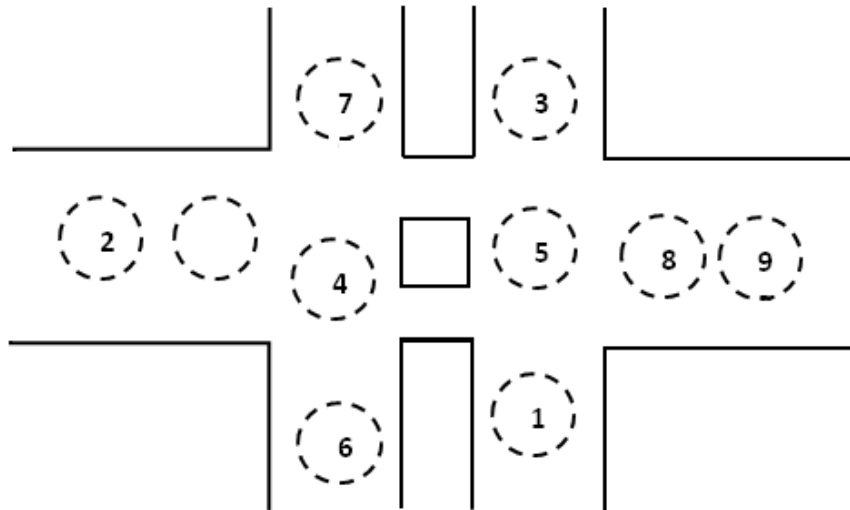


Figure 3.6 Topography showing the movement of nodes for Manhattan Mobility Model.

The mobile node is allowed to move along the grid of horizontal and vertical streets. At an intersection of a horizontal and a vertical street, the mobile node can turn left, right or go straight with certain probability.

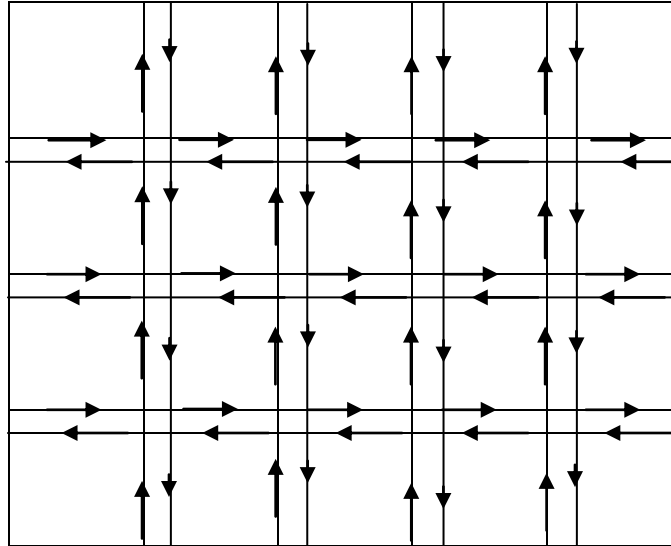


Figure 3.7 Travelling pattern of a Mobile Node using Mnahattan mobility model.

The nodes are initially assumed to be randomly placed at street intersections and the movement of a node is decided one street at a time.

A few variations in these conventional models have been made to suit the requirements of this research and to realize practical mobility patterns for robotic search and rescue.

3.2 Swarm Robotics

Swarm robotics is an emerging branch of cooperative robotics. Moving toward targets or regions will create movement patterns dissimilar to those created by any of the afore-mentioned random mobility models. Swarm robotics is focused on creating a system of simple modules from which complex behaviors emerge. This chapter examines a few algorithms available in literature used to efficiently disperse a large group of mobile robots through a bounded region using inter-robot sensing or communication.

3.2.1 Biological Inspiration:

Animals can move in environments inaccessible to humans (pipes, collapsed building, etc). Their locomotion includes crawling, walking, wall climbing, swimming, jumping, flying, etc. This has inspired attempts at designing robots imitating biological systems. These robots are expected to exhibit much greater robustness in performance in unstructured environments than robots available today.

SWARM intelligence is the most widely applied area of research while studying the applications of groups of communicating biologically inspired robots. It is a novel approach to the collaboration of large numbers of robots. It is inspired from the observation of social insects - --ants, termites, wasps and bees- where individual creatures interact to create collectively intelligent systems. Social insects are known to coordinate their actions to accomplish tasks that are beyond the capabilities of a single individual: termites build large and complex mounds, army ants organize impressive foraging raids, and ants can collectively carry large preys. Such coordination capabilities are still beyond the reach of current multi-robot systems, but a number of prototypes have been made. A robot fly has wings controlled by an actuator that drives wing movement when a voltage is applied [45]. A swarm of such ‘flies’ promises valuable aid during rescue missions in the event of disasters like earthquakes, mine explosions and volcanic eruptions. “The tiny machines would detect signs of life, perhaps by sniffing the carbon dioxide of survivors' breath or detecting the warmth of their bodies [45].”

Snake Robots that can slither in and out of small crevices, climb up pipes and small vents are a new breed of life-saving machines that can be invaluable not only for search and rescue, but also find great use in human surgery[46]. These robots have been used by search and rescue teams, the Indiana and New Jersey task forces, when they go on training scenarios. Currently these snake bots are battery powered. This adds limitations to how long they can

operate and also the distance that can be covered by them. Advances in fuel cell technology could give them the range and power source that the designers of these robots have been looking for.

3.2.2 Ant Colony Optimization:

The initial biologically-inspired research in developing swarm intelligence was done by studying ant colonies. the **ant colony optimization** algorithm (**ACO**) is a probabilistic technique [47] for solving computational problems which can be reduced to finding good paths through graphs. In the real world, ants (initially) wander randomly, and upon finding food return to their colony while laying down chemical marker trails. These markers are referred to as 'pheromones'. If other ants find such a path, they are likely not to keep travelling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food.

Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. A short path, by comparison, gets marched over faster, and thus the pheromone density remains high as it is laid on the path as fast as it can evaporate.

Pheromone evaporation has also the advantage of avoiding the convergence to a locally optimal solution. If there were no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained.

Thus, when one ant finds a good (i.e., short) path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads all the ants following a single path. The idea of the ant colony algorithm is to mimic this behavior with "simulated ants" walking around the graph representing the problem to solve.

3.3 Virtual Pheromones:

‘Virtual pheromones’ are symbolic messages tied to the robots themselves rather than to the fixed locations in the environment. Virtual pheromones are locally transmitted without specifying a recipient. These decay over time which makes any obsolete information void. In the context of this thesis, this algorithm can be implemented for coordinating the actions of large numbers of small-scale robots to achieve useful large scale results in surveillance and path finding [48]. When applied to this research, pheromones can prove helpful for both the search and rescue phases of operation.

3.3.1 Attraction and Repulsion forces between robots:

Robots make use of repulsion and attraction to maintain reasonable distances from neighboring nodes and obstacles. Robots in close proximity repel each other to avoid collisions and maintain connectivity with the help of the attractive forces. Figure 3.8 illustrates the repulsion and attraction zones surrounding each robot.

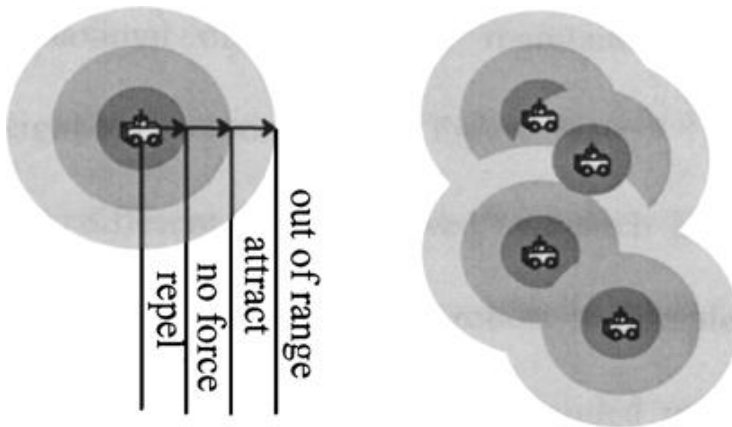


Figure 3.8: Zones surrounding a robot implementing Pheromone robotics [48]

3.3.2 Payton’s Guided Growth Models:

There are a number of techniques for coordinating the actions of large number of small-scale robots to achieve useful large-scale results in surveillance, reconnaissance, hazard detection, and path finding.

3.3.2.1 Gas Expansion Model:

Payton's gas expansion model [48] emulates the way gas particles fill a vacuum. The robots expand from an initial compact state, based on a competition between attraction and repulsion behaviors that depend on distances to obstacles and other robots. This model makes use of the attraction and repulsion forces between the robots to maintain a reasonable distance between them. These simple behaviors allow a robot swarm to expand from a tight grouping into a maximal dispersion while maintaining nearest-neighbor communications.

3.3.2.2 Biologically Inspired Guided Growth Model:

A more advanced form of guided growth is inspired by analogies to plant growth. One robot is designated as a "bud" and given a repulsive urge that is stronger than its attraction to other robots (Fig.3.2 (a)). As the bud moves away from the others, they expand to fill the space (Fig.3.2 (b)) to maintain communications connectivity. The bud emits a growth inhibitor pheromone, which inhibits the formation of other buds, so that the robots tend to string out in a column that stretches away from the user. When a bud determines it can make no more forward progress, it stops transmitting the growth inhibitor (Fig.3.2 (c)). In the absence of growth inhibitor, other robots that detect open space can become buds (Fig.3.2 (d)).

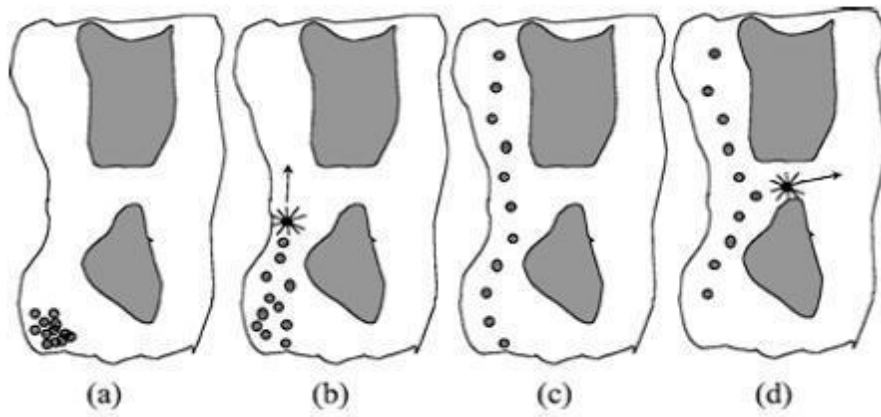


Figure 3.2: A “bud” acts as a single point until it hits a dead end, then a new “bud” takes over[48].

CHAPTER 4

ROBOT ARCHITECTURE AND SIMULATION MODELS DESIGNED

This chapter provides an in dept discussion of the experimental setup (both hardware and software), the various tools employed and the simulation models developed. The work presented in this thesis is one of many collaborative projects of the Cooperative Robotics Research (CRR) Team at Auburn University. This chapter briefly introduces the steps involved in the architecture of the constructed robots and the hardware test-bed [49]. These robotic nodes have already been used in the implementation of SARA-2 and other ongoing research that can be found on [42].

The latter part of this chapter discusses the various simulation models developed with the objective of running the code on this team of robots to design a search and rescue algorithm. Four models are explained based on the Random Way Point mobility model and the Manhattan mobility model. Through this chapter the terms nodes and robots are used interchangeably.

4.1 Laboratory set-up:

A team of six robots has been implemented in the laboratory. The functionality includes on-board processing, mobility and wireless capability. The laboratory experiment is carried out in a floor plan of dimensions 8 ft x 16 ft , with boundaries made of medium density fiberboard (MDF) walls. The walls are 12” tall and are also constructed from MDF. The layout consists of a 5 m long aisle is flanked by three rooms on either side with openings to serve as doorways.

A stationary target cone is deployed at a random position when required. The robots are to perform a collaborative search of the environment for the stationary target. The robots use two

2800 mAh /7.2 V batteries for operation and are capable of motion at the rate of 0.5 inch per second ($\frac{1}{4}$ 0.0381 m/s). The field is shown in Figure 4.1.

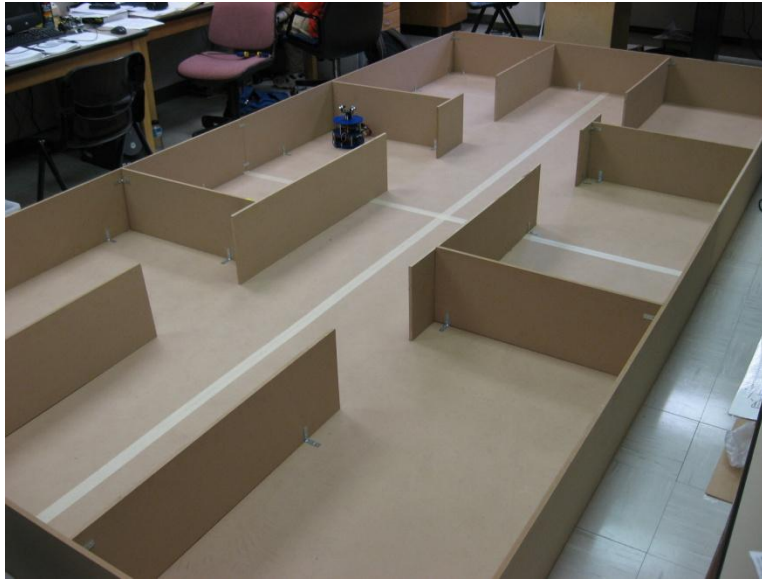


Figure 4.1 Image of the hardware testbed.

4.2 Robot Architecture:

The robot structure consists of five basic parts : Chassis, Drive System, Sensors, Control System and Batteries.

4.2.1 Chassis :

A BudgetRobotic's ScooterBOT II chassis kit was chosen for the chassis. This kit consists of a 7" diameter round chassis assembly designed for differential drive steering. It consists of the base platform with cut-outs for the drive wheels and an upper platform for housing the computer and interface board. A third platform was used to serve as a base in mounting the rotating sensor turret.

4.2.2 Drive Systems:

The kit used its own drive systems which consists of GW Servo S03NXF STD continuous rotation servos and allows speed control based on standard servo timing parameters.

The wheels used have a diameter of 2.5 inches, giving the robot a maximum speed of 22.16 cm/s.

4.2.3 Sensors:

Two types of sensors were used in the architecture: Distance sensors to observe the world around the robot and plot a course through it and Odometric Sensors, to determine the robot's position in the world.

Sharp GP2D12 infrared (IR) sensors were employed for distance sensing. These sensors output an analog voltage that is linear with the inverse distance. They have a minimum range of 10 cm and a maximum range of 80 cm. Each sensor should also have a filter capacitor from the IR signal line to ground to short any high frequency noise. These sensors were mounted on a turret to aid in calibration. Figure 4.2 shows a picture of the completed IR distance sensor turret.

A floor encoder system based on Optical mice was employed for odometric sensing. Floor encoder systems measure the movement of the robot with respect to the floor.

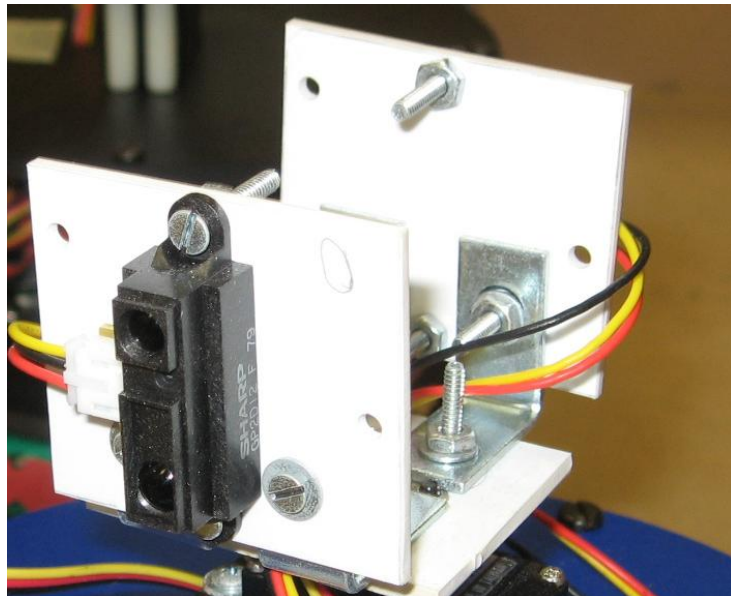


Figure 4.2: IR sensor turret assembly.

4.2.4 Control System:

The Gumstix platform was used to control the systems. The control system consisted of an on-board Linux OpenEmbedded platform built on a Gumstix Connex 400xm-bt motherboard based on the 400MHz Intel XScale PXA255 processor. The motherboard featured a 64 MB SDRAM and 16 MB Flash Memory. Wireless connectivity was enabled by a Wifistix expansion board that connects to the Connex motherboard via the 92-pin bus header. The Wifistix FCC (with Wifi antenna) used standard 3.5V- 6V input. The on-board Marvell 88W8385 chipset provided IEEE 802.11a/b/g WLAN capability. (The 88W8385 enjoys embedded CPU and on-chip memory). Fig. 4.2 shows the equipment used in the laboratory.

4.2.5 Batteries

To power the robots, the decision was made to use normal RC car batteries. The selected batteries provided 7.2V at 2800 mAh. These were primarily chosen due to familiarity and reliability. Two batteries were used per robot. One battery to power the electronics, the sensors and the microcontroller stack, while the other to power the three servos, two drive servos and one turret servo. An image of the batteries mounted on the robot can be found in Figure 4.3.



Figure 4.3: Batteries and mounted optical mice.

The Search and Rescue Algorithm (SARA- 2) uses six of the previously described robots to search for a stationary target [50]. Each robot follows a set of behaviors in the process: map-building, error correction, path planning and path following. A complete series of these behaviors constitutes one step. Thus, one step involves the robot building a map, correcting sensor errors, planning and following a path to the next destination point. The process repeats with each robot performing these steps until the target is found.

4.3 Simulation Model Fundamentals:

4.3.1 Boundary Conditions:

A probabilistic approach is used for checking boundary conditions. A robot which can go outside the set boundary is probabilistically restricted from making a movement during that iteration thus keeping it at the same position, this helps satisfy the boundary conditions and keep the robot within the region.

4.3.2 Obstacle Avoidance:

A basic obstacle avoidance algorithm is used. If a node encounters an obstacle, along its path to the next coordinate, the robot retreats back to its previous position and a new coordinate and speed is assigned to it.

4.3.3 Communication:

The nodes communicate with each other. Communication involves a broadcast of each node's explored coordinate after every iteration. This greatly reduces redundancy of the search area. This information can be used to build a global map of the environment from each node's local map.

4.3.4 Repulsive forces between nodes:

To reduce redundant search area, the robots need to have a mechanism to avoid collisions with each other. The nodes need to be aware of the neighboring robots and avoid collisions while staying within the communication range of the repelling robot and maintaining the network. This also helps distinguish between obstacles and other robots.

This is done with the help of attractive and repulsive forces between the nodes. Each robot is simulated to be circular in shape and has a circular repulsion zone around it. For a robot not in the repulsion zone of other robots, the next coordinate and speed is randomly generated coordinate and assigned. For those lying within each others' repulsion zones, the speed and direction are calculated as follows:

A circular region centered on a robot with a radius of r_r is considered as the repulsion region of that robot. If a robot is within one or more repulsion regions of other robots, then it will calculate its movement vector based on the repulsion force(s), instead of based on the random movement vector. Let $\vec{f}(i, j, t)$ be the repulsive force induced from i to j iteration t and $\vec{m}_j(t - 1)$ be the movement of robot j at time $t - 1$, regardless whether it is due to random movement or repulsion forces. The movement vector of j at iteration t is

$$\vec{m}_j(t) = \vec{m}_j(t - 1) + \sum_{i \in I/j} \vec{f}(i, j, t)$$

where I is the set of all robots and with $d_{i,j}$ being the distance between $i: (x_i, y_i)$ and $j: (x_j, y_j)$,

$$|\vec{f}(i, j, t)| = (r_r - d_{i,j}),$$

$$\angle \vec{f}(i, j, t) = \arctan \frac{(y_i - y_j)}{(x_i - x_j)}$$

4.4 Simulation Mobility Models Developed:

As mentioned earlier, four simulation models are developed based on existing basic mobility models. Many of the existing mobility algorithms for multi-node networks do not provide realistic movement scenarios; they are either purely random in nature without taking into account obstacles or do not incorporate communication between nodes.

This section discusses two models based on the Random Way Point mobility model and two based on Manhattan mobility model.

The RWP model is our first choice due to its ease of implementation. The second choice of mobility model is made because the Manhattan model introduces a more realistic pattern which is based on movement of vehicles in the streets in an urban area. A few variations in these conventional models have been made to suit the requirements of this research and to realize practical mobility patterns.

4.4.1 Random Mobility without Communication and no Obstacle Avoidance:

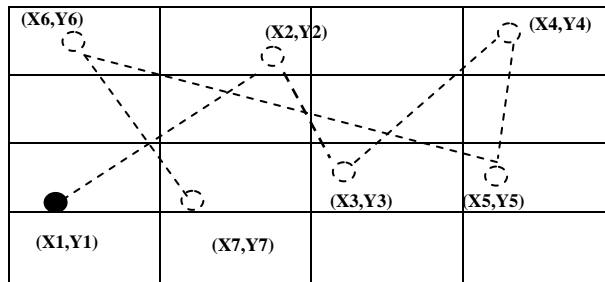


Figure 4.4: Travelling pattern of a mobile node using Random Way Point mobility model. The grid size is 1 m x 1m.

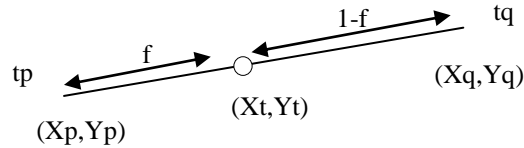
The conventional Random Way Point Model, as discussed in Chapter 3, includes a fixed pause time between two successive movements. This model is slightly different from its traditional design, in that, there is no pause time between iterations.

Figure 4.4 shows the movement of a mobile node (N_1) using this mobility model. Let (X_1, Y_1) be the initial location of node N_1 at time t_1 (mostly, $t_1=0$, the starting time). N_1 randomly selects a

co-ordinate (X_2, Y_2) to move to. N_1 also selects a random velocity that is uniformly distributed between $V_{min}(0m/s)$ and $V_{max}(5 m/s)$ and moves from (X_1, Y_1) to (X_2, Y_2) with that velocity. N_1 reaches (X_2, Y_2) at time t_2 .

After reaching (X_2, Y_2) , N_1 changes direction by randomly choosing a co-ordinate (X_3, Y_3) and moves to the chosen co-ordinate at time t_3 using a randomly chosen velocity. The node continues to move using this procedure until the end of the simulation time period.

The position of node 'i' at any time t is found as described next.



The successive waypoints p and q are found such that $tp < t < tq$. If (Xp, Yp) and (Xq, Yq) are the co-ordinates at time instants tp and tq respectively, then the distance traveled between these two coordinates is given by:

$$d(p - q) = \sqrt{(Xp - Xq)^2 + (Yp - Yq)^2} \quad (1)$$

$$f = \frac{t - tp}{tq - tp} \quad (2)$$

$$Xt = f * Xq + (1 - f) * Xp \quad (3)$$

$$Yt = f * Yq + (1 - f) * Yp \quad (4)$$

(1), (2), (3) and (4) help determine the consecutive positions of a mobile node in a Random Way Point mobility model. Figure 4.5 is a flow chart of this simulation model.

In this model, each node treats other nodes as obstacles.

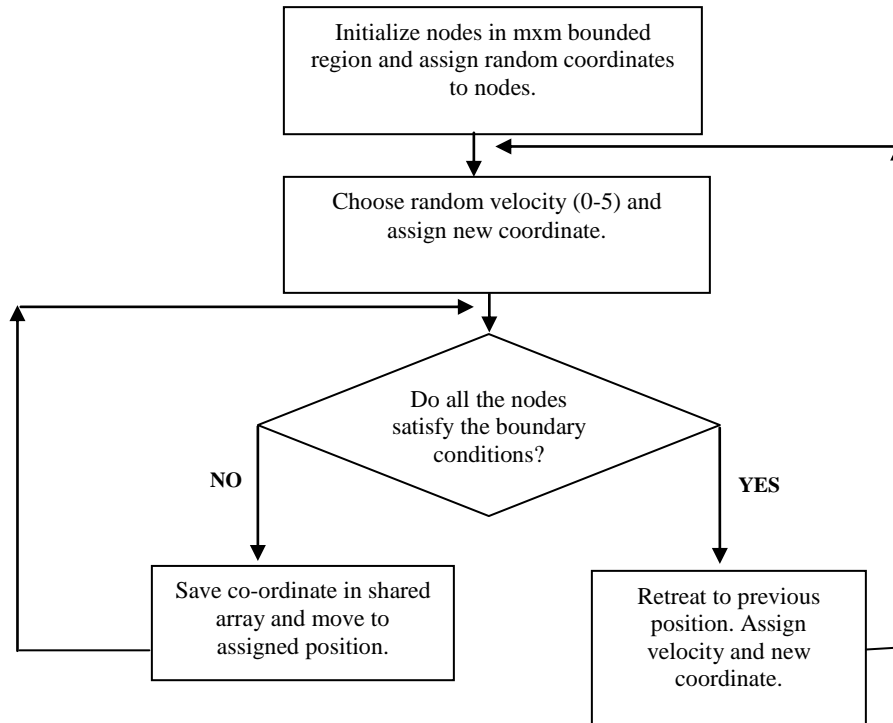


Figure 4.5: Simulation setup for the Random Way Point model with no obstacle avoidance and no communication between nodes.

4.4.2 Random Mobility with Obstacle Avoidance and Communication.

This variation of the RWP mobility model incorporates obstacle avoidance and inter-node communication. Communication involves a broadcast of every node's explored coordinates after every iteration, greatly reducing redundancy.

In this model, robots distinguish between obstacles and other robots by the use of repulsive forces as mentioned in Chapter 3.

4.4.3 Manhattan Mobility Model with Obstacle Avoidance and no Communication.

In this model, each node chooses a random direction along the grid when it reaches a junction and keeps moving in the chosen direction until it reaches the next junction. .

The nodes are initially assumed to be randomly placed at street intersections. The movement of a node is decided one street at a time. To start with, each node has an equal probability of choosing any of the streets leading from its initial location. After a node begins to move in the chosen

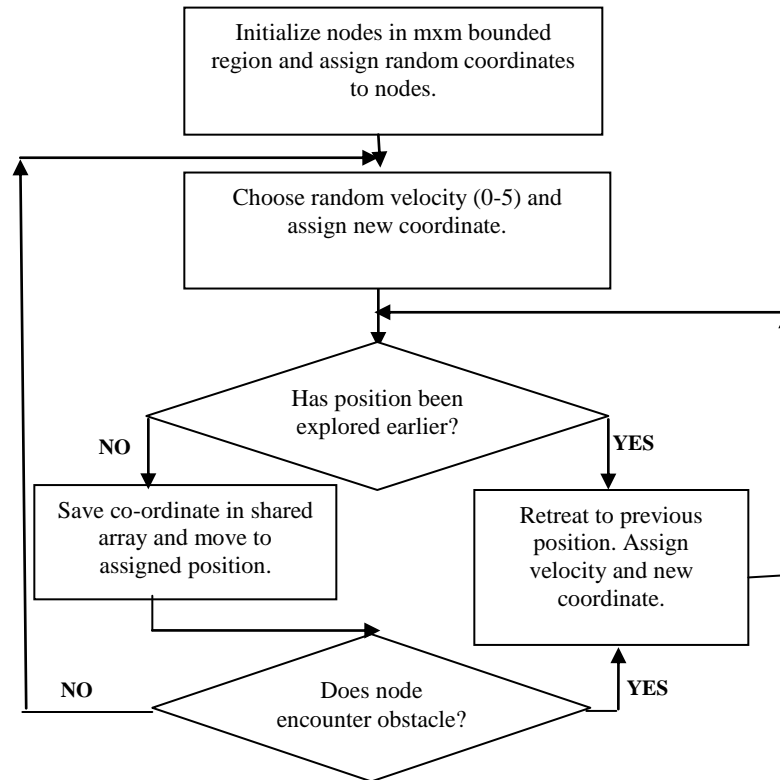


Figure 4.6: Simulation setup for the Random Way Point model with obstacle avoidance and inter-node communication.

direction and reaches the next street intersection, the subsequent street in which the node will move is chosen probabilistically.

If a node can continue to move in the same direction or can also change directions, then the node has 50% chance of continuing in the same direction, 25% chance of turning to the east/north and 25% chance of turning to the west/south, depending on the direction of the previous movement. If a node has only two options, then the node has an equal (50%) chance of exploring either of the two options. If a node has only one option to move (this occurs when the node reaches any of the four corners of the network), then the node has no other choice except to explore that option.

Figure 4.8 depicts the movement of a model using the Manhattan mobility model. The node reiterates its position once it reaches an intersection or junction.

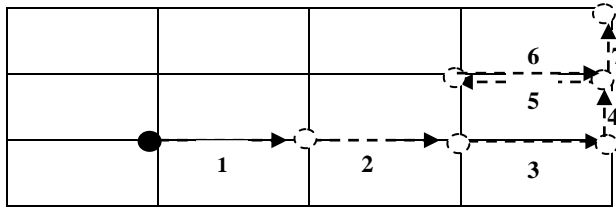


Figure 4.8: Travelling pattern of a mobile node using Manhattan mobility model.

A simple obstacle avoidance technique has been included in this model. Once an obstacle is encountered, the node returns to its previous position and a new coordinate is assigned to it. The flow chart of this model is given in Figure 4.9.

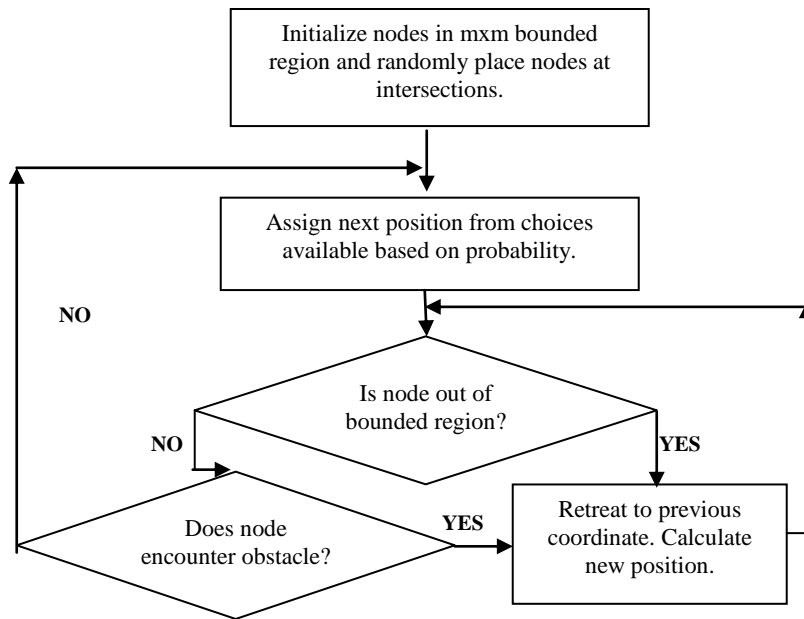


Figure 4.9: Simulation setup for the Manhattan mobility model with obstacle avoidance no inter-node communication.

CHAPTER 5

RESULTS

Mobility models have long been used to capture movement patterns of mobile users, be it PDA's or vehicular networks. When applied to robotic nodes, slight variations in these models are imperative to ensure realistic motion. Since this thesis aims at designing mobility models to be used in search and rescue, two vital aspects are introduced into the simulation environment; obstacles and inter-node communication.

This chapter analyses the effects of these characteristics on the movement of mobile robotic nodes using the simulated algorithms.

5.1 Simulation Setup:

The performance evaluation of the two mobility models is made via simulations using MATLAB 7.0. Each robot is simulated to be circular in shape with a radius of 5cm and a repulsion radius of 20cm. Initially, the nodes are randomly deployed in a 1000 x 1000 m area. The number of nodes is varied from 1 to 50 in the simulations. The velocity of the nodes is randomly varied between 0-5 m/s. A timer is incorporated for simulations with both models. The limit of the timer is 120s and the simulation is aborted if the timer times out. Table 5.1 summarizes the various parameters considered. Each data point on the graphs is an average of

100 runs to average out the randomness. The simulation is halted when one node finds the target or the timer times out, whichever happens first.

5.2 SIMULATION PARAMETERS:

Parameters	Value
Simulator	MATLAB 7.0
Area	1000x1000m
Speed	0-5cm/s
No of Nodes	1-50
Robot Radius	5cm
Repulsion Radius	20cm
No of runs	100

Table 5.1: Simulation Parameters

For simulation purposes the following parameters are varied:

- Target Motion: Stationary Vs Mobile.
- Communication Interval: None Vs Continuous.
- Obstacle Avoidance.
- Repulsion between robots.
- Number of robots.

5.3 Results and Analysis:

The factors above are varied; the effects are examined and analyzed for causes and implications. Since these models are designed for search and rescue, time is the most crucial factor in all these simulations. Time is accounted for by the number of steps taken by the robots

collectively before finding a target. Thus graphs are plotted between various other characteristics and the number of steps taken before a target is found.

5.3.1 Target Mobility:

The first two graphs examine varying target motion from stationary to mobile. In Fig 5.1, the two RWP models are simulated for a mobile target and the number of nodes is increased from 0 to 50 in steps of 10. Figure 5.2 examines the effect of increasing the number of nodes when the search is carried out for a stationary target.

As the number of nodes increase, the steps taken to find the target reduce for both models. From Figure 5.1 it is evident that for a mobile target with an increase in the number of nodes, the number of steps needed for completion of the search task was relatively less with continuous communication than that in the absence of communication. The results are however more drastic for a stationary target. As seen in Figure 5.2, the number of steps to detect the target drops considerably when continuous communication is employed. The greater time taken to locate a mobile target can be accounted to the fact that a moving target further increases the randomness of the simulation.

A mobile target scenario would be better suited to a military based search in combat. Since this study is aimed at searching for targets in the aftermath of disasters, further search simulations are carried out for stationary targets.

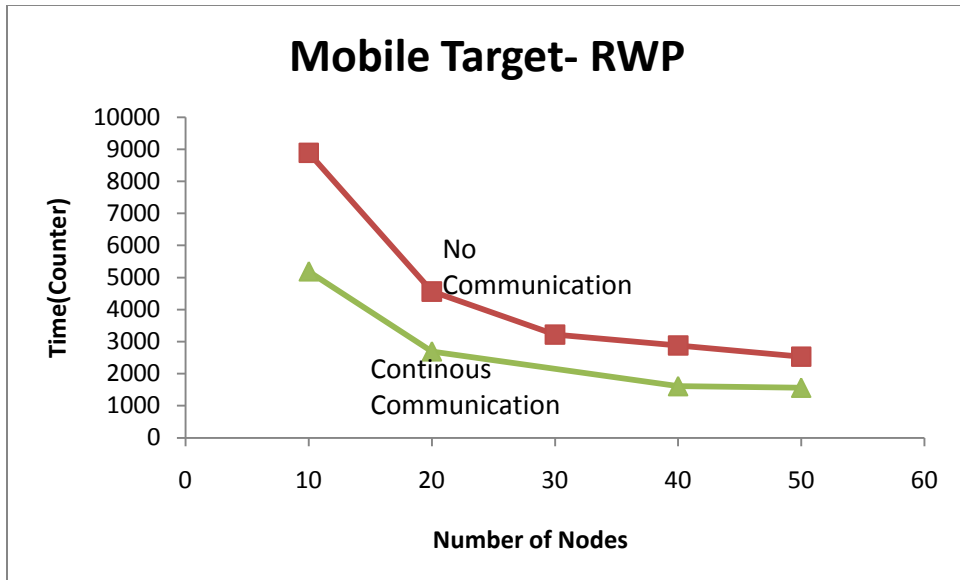


Figure 5.1: Mobility follows the RWP model. The frequency of communication is varied from none to continuous for a mobile target.

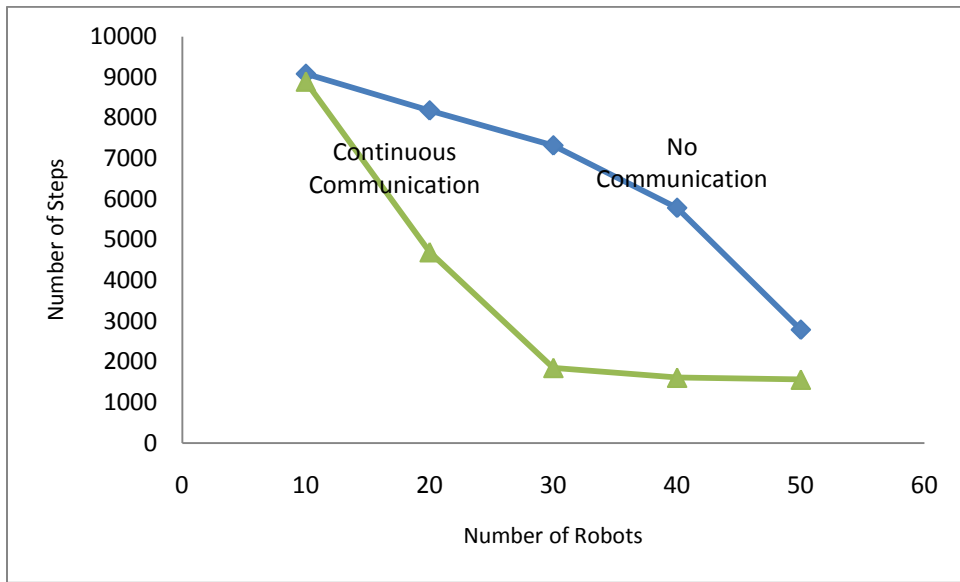


Figure 5.2: Mobility follows the RWP model. The frequency of communication is varied from none to continuous for a stationary target.

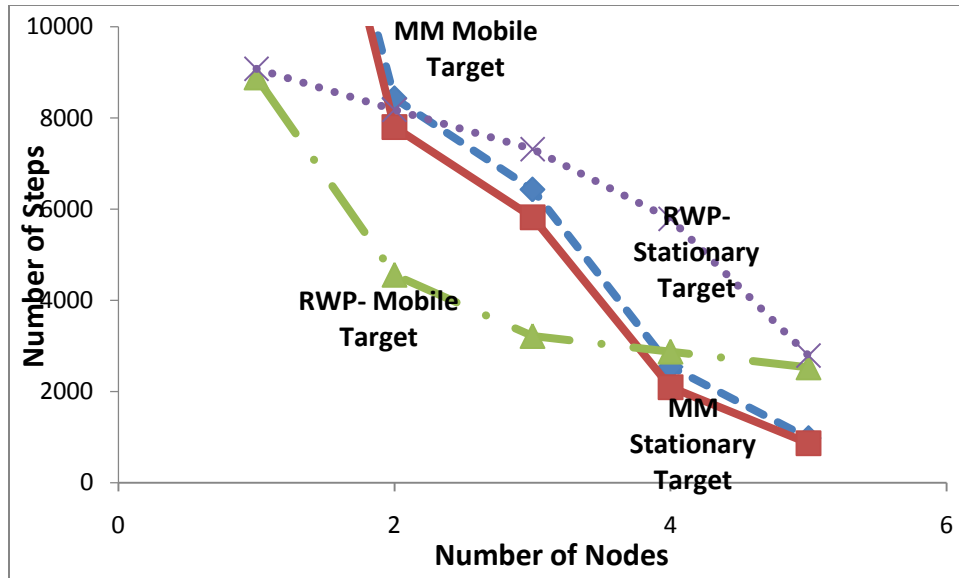


Figure 5.3: Both RWP mobility and Manhattan mobility models represented. No communication between nodes. Target motion is varied between stationary and mobile for both models.

A comparison of the simulated random model and the Manhattan model is presented in Figure 5.3. The first set of simulations for the Manhattan model when one node is searching for the target returns no result leading to the timer timing out. This is attributed to the fact that a node following Manhattan mobility has a high probability of getting stuck in one part of the simulation area. As the number of nodes increase however, there is a very steep decline in the time to completion. Also, on comparison with the RWP mobility model, the Manhattan mobility model requires less number of steps for a search task for both stationary and mobile target scenarios.

5.3.2 Varying intervals of communication: None, Occasional, Continuous.

The intervals of communication are varied to study how much communication is necessary to result in efficiently and successfully finding the target in minimal time. The simulations for occasional communication are similar to those for continuous. This time communication was carried out at every other iteration.

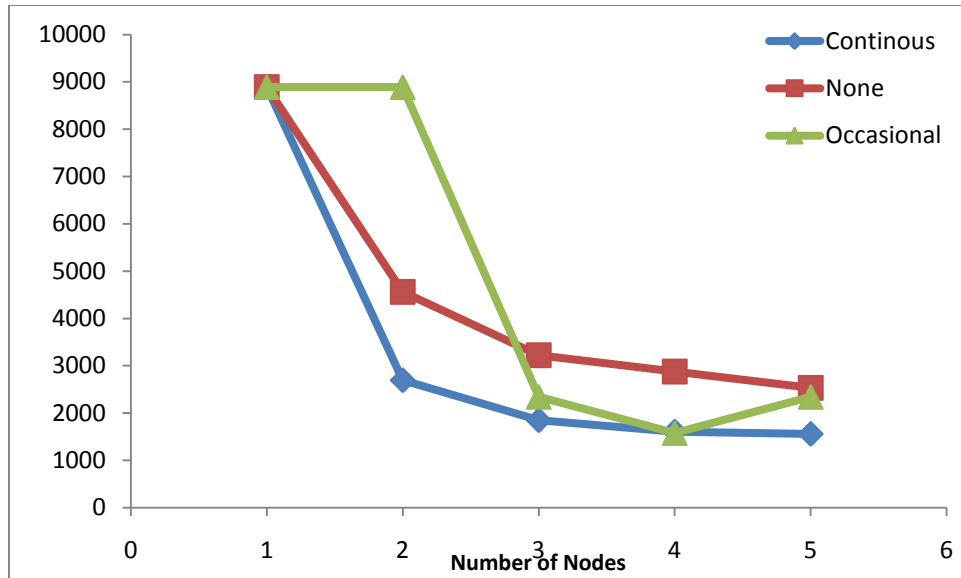


Figure 5.4: Mobility follows the RWP model. The communication interval is varied from none to occasional to continuous.

The intervals of communication are varied to study how much communication is necessary to result in efficiently and successfully finding the target in minimal time. The simulations for occasional communication are similar to those for continuous. This time communication was carried out at every other iteration. Ideally, this should help reduce the time to find the target, as lesser time is spent on communicating and more is spent on searching for the target. The negative byproducts of employing occasional communication are increased inter robot interference and redundant search area.

As seen in Figure 5.4, when the robots increase in number, there is a steep drop in the number of steps taken to find the target with occasional communication between nodes. But after a threshold of forty robots is reached, the number of steps starts increasing again. This can be attributed to the fact that other robots are treated as obstacles, leading to a greater number of iterations before the target is found.

5.3.3 Obstacle avoidance:

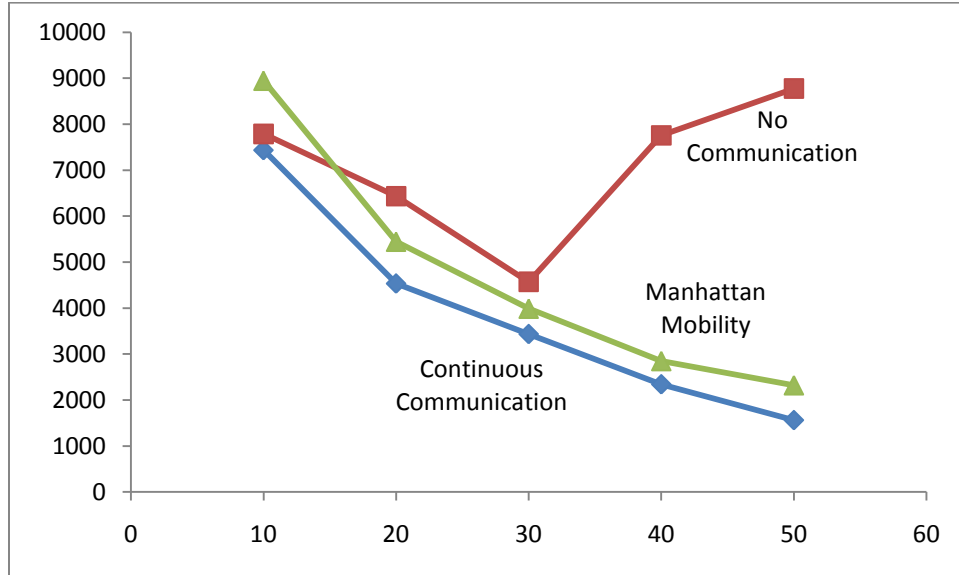


Figure 5.5: Mobility follows the RWP model. Obstacles are included in the simulation at random locations.

The incorporation of obstacles has a profound effect on the number of steps taken to complete the search task. In Figure 5.5, as the nodes increase, the iterations to find the target with no communication between nodes drastically increase. This can be attributed to the fact that every time an obstacle is confronted, the node retreats back to its original position and recalculates a new coordinate. In the random model with no communication, this has the effect of doubling the number of iterations and redundancy in search area. The initial drop may be due to the random nature of node movement. When communication is employed however, the inclusion of obstacles has the reverse effect of reducing the number of iterations by half. Continuous communication leads to an awareness of the positions of the obstacles among the network of nodes, thus reducing redundant search.

The absence of communication in the Manhattan model has the same effect as on the no communication random model. As the number of nodes increase so do the reiterations. This has the effect of increasing the number of steps to find the target.

5.3.4 Varying the number of obstacles:

As seen in 5.3.3, introducing obstacles cuts the number of iteration in half and help in faster tracking of the target. In this section we try to examine how the models perform when the obstacles are increased. This experiment also determines the maximum number of obstacles this simple obstacle avoidance algorithm can handle efficiently.

For the following set of experiments the number of robots was kept constant at 30 and the number of obstacles is increased from 5 to 35 in steps of 5.

As the number of obstacles increase, when no communication is employed between robots, the steps to find the target increase linearly. This is due to the fact that the other robots are also seen as obstacles and thus there is a lot of redundancy in search area. With communication between nodes, the time to completion reduces initially and then increases linearly when more than 15 obstacles are introduced. This is attributed to the fact that more obstacles mean more reiterations and thus more steps taken before the target is found. Communication helps keep the steps to find target low till the number of obstacles stays below 15. Thus we see that the threshold of the number of obstacles this algorithm can avoid successfully is 15.

As the number of obstacles increase from 0 to 30, the simulation results for nodes following Manhattan mobility are similar to the continuous communication random model. When the obstacles are less in number, the Manhattan model outperforms the communication model as shown in Figure 5.6. This can be attributed to the sequential search nature of the Manhattan model.

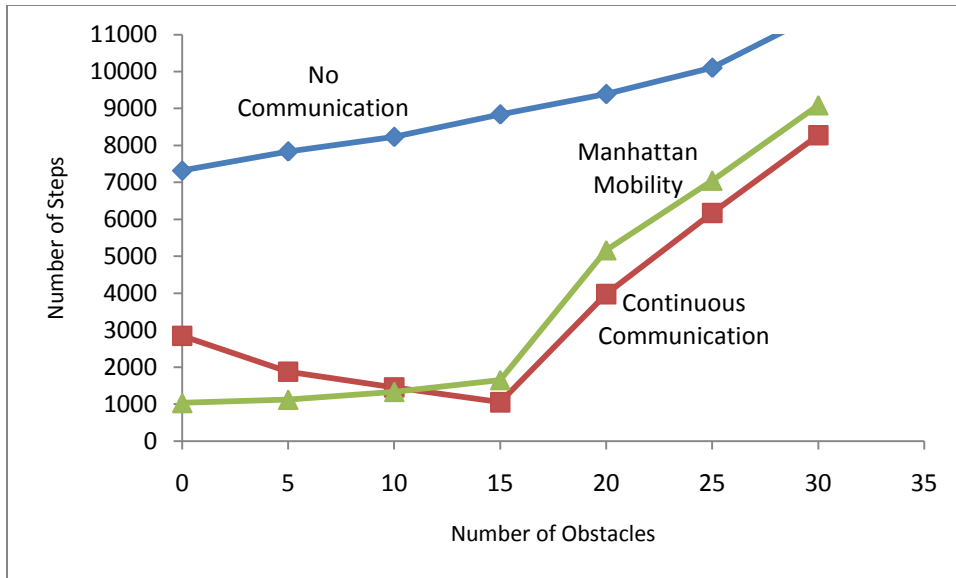


Figure 5.6: Number of obstacles is increased. Nodes follow RWP mobility model.

5.3.5 Varying size of the robots:

Simulations are run where the radius of the robots is varied from 0-50 cm in steps of 10. This has an interesting effect on the results. As before, the repulsion radius is set at 50cm and 30 nodes are simulated.

In the case of no communication, as seen in Figure 5.7, initially there is a sharp drop in the time to find the target, as nodes increase. This can be attributed to the large size of the robots which means a larger area being sensed by the robot's sensors. After the size increases beyond 30 cm, owing to larger sized obstacles, the steps to find the target increase steadily.

In the case of the communication model, the time to task completion drops steadily till the robot radius = 25 cm. Once the radius is greater than the repulsion radius, repulsions cease to exist and redundancies increase. From here on, it takes longer time to find the target.

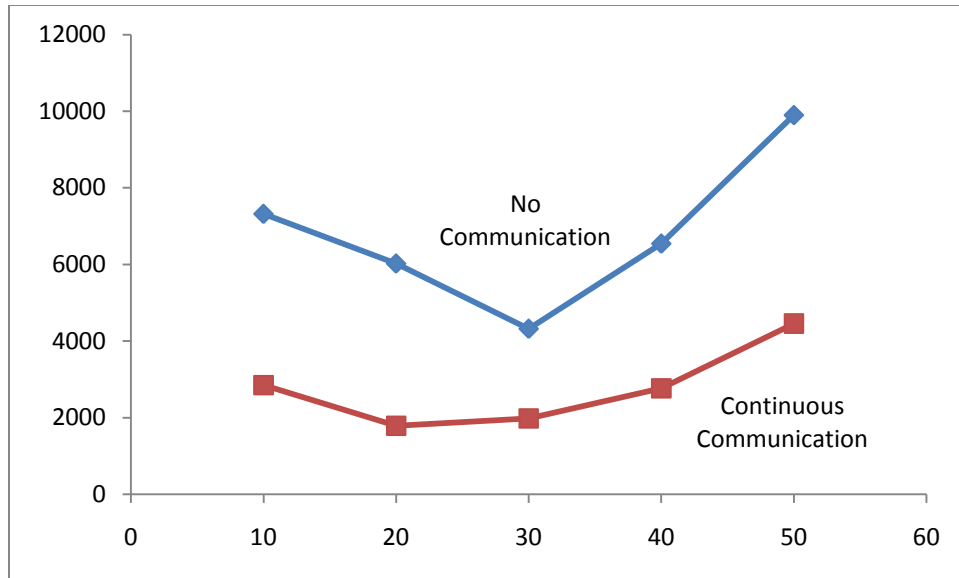


Figure 5.7: The radius of the robot is varied while the repulsion radius is kept constant.

5.3.6 Increasing Maximum speed:

As mentioned in Chapter 4, at every step in a random mobility model two parameters are randomly generated; speed and the next coordinate. Speed assigned can be anywhere between V_{\min} to V_{\max} . In earlier experiments the maximum speed was set to 5cm/s. In this section the maximum speed is increased in steps of 5 from 5cm/s to 40 cm/s.

Increasing maximum speed advertently has the effect of increasing the average speed of the robots as shown in Figure 5.8. Due to the presence of repulsive forces, there is an almost exponential increase in the average speed of the simulation for random mobility with communication.

Since the average speed increases, with no communication and no repulsive forces, the collisions between nodes increase. This results in reducing the average distance travelled by the robots, thus increasing time to find the target. In the random model with communication and Obstacle avoidance, when average speed increases, the average speed of robots within each

other's repulsion radius further increases increasing the average distance travelled. Since more distance is covered, the time to find the target reduces.

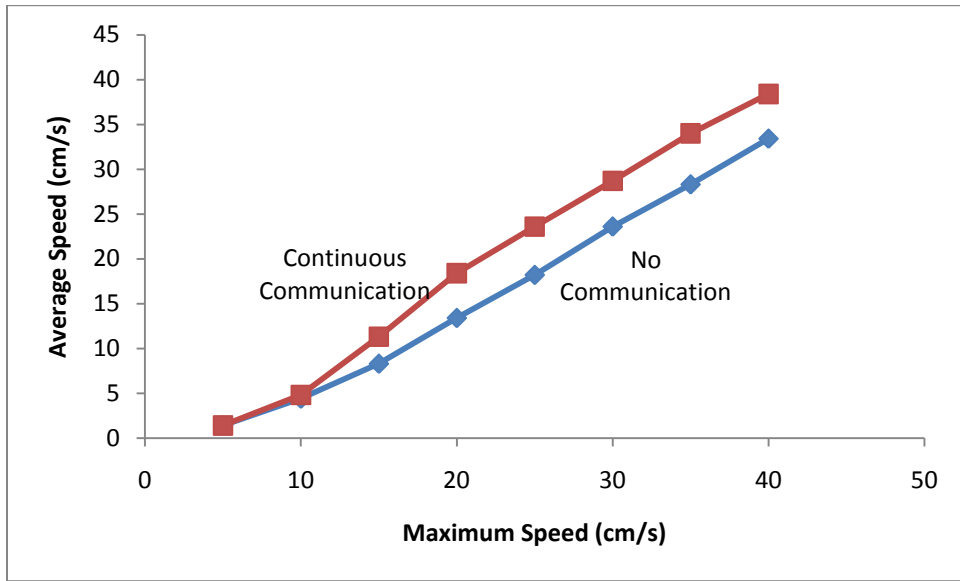


Figure 5.8: The maximum speed of the nodes is increased and the effect on average speed is studied.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This thesis puts forth the possibility of applying conventional mobility models for cooperative search and rescue using robots. The experiments involve a group of robots searching for a stationary or mobile target. Various characteristics are varied: the number of robots, inter-robot communication interval, the average speed of the robots, the number of obstacles and the size of the robots.

6.1 Conclusions:

Three mobility models are simulated: the random mobility model with no communication, the random mobility model with communication and obstacle avoidance and the Manhattan mobility model with obstacle avoidance and no communication.

Simulation results presented in this work suggest that equipped with a simple obstacle avoidance algorithm and inter-node communication, the robots exhibit significant improvement in completing the task of finding a stationary target.

Increasing the number of nodes displayed obvious results. There is an inverse relationship between the number of nodes and the number of steps for task completion in all the models. On comparison of the three models, it is evident that the Manhattan Mobility model gets more efficient as the number of nodes increases.

Results indicate that inter-robot communication considerably reduces the time to completion of a cooperative search task for the random mobility model. Also, in the absence of

communication, the number of steps required is less for the Manhattan model compared with the random models whether the target is stationary or mobile.

In the communication experiments, the communication interval was varied from continuous to occasional to none. Both sets of experiments using communication, and therefore cooperation, outperformed the *no communication* runs in terms of time for task completion. The occasional strategy outperformed the continuous strategy when the robots were less in number. As the number increased, the time taken for task completion with occasional communication dropped owing to lesser communication and redundancy in search area.

Results also indicate that the obstacle avoidance algorithm fails when the number of obstacles exceeds a certain limit. With the help of simulations, this threshold was found to be 15. Work can be done to come up with a more sophisticated obstacle avoidance algorithm that can deal with more obstacles and robots can work their way around the obstacles instead of retracing their steps.

The next section presents some more suggestions for future work.

6.2 Suggestions for Future Work:

- 1) Implement these mobility models on CRR laboratory's mobile robots.
- 2) Consider the use of cognitive radios for inter-robot communication.
- 3) Introduce inter-node communication in the Manhattan mobility model.

REFERENCES

- [1] Dave Scheiber. "Robots to the rescue". Internet:
http://www.sptimes.com/2003/03/02/Floridian/Robots_to_the_rescue.shtml, March 2, 2003
[April 10, 2010]
- [2] CNN Wire Staff, "Robots succeed, cut well pipe, oil gushes into gulf". Internet:
<http://www.cnn.com/2010/US/06/01/gulf.oil.spill/index.html>", June 1, 2010 [June 10, 2010]
- [3] C. Luo, S. X. Yang, "A Real-Time Cooperative Sweeping Strategy for Multiple Cleaning Robots," International Symposium on Intelligent Control, pp. 660-665, 2002.
- [4] Y.U. Cao, AS. Fukunaga, and A.B. Khang." Cooperative mobile robotics: Antecedents and directions." Autonomous Robots, 4, 1997.
- [5] G. Dudek, M.. Jenkin, E.E. Milios, and D. Wilkes. "A taxonomy for multi-agent robotics." Autonomous Robots, 3(4), 1996.
- [6] D. Guzzoni, A. Cheyer, L. Julia, and K. Konolige. "Many robots make short work." AIMagazine, 18(1):55-64, 1997.
- [7] Sanchez M, Manzoni P. "A java-based ad hoc networks simulator." In Proceedings of the SCS Western Multiconference Web-based Simulation Track, January 1999.
- [8] C.G. Wilson and T. Roppel, "Low-cost Wireless Mobile Ad-hoc Network Robotic Testbed", Testbeds and Research Infrastructures for the Development of Networks & Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on , vol., no., pp.1-6, 6-8 April 2009.
- [9] A. Davids, "Urban search and rescue robots: from tragedy to technology," in IEEE Intelligent Systems and their Applications, vol. 17, no. 2, 2002, pp. 81-83.
- [11] National Science Foundation. "Search-and-Rescue Robots Practice Emergency Response To Simulated Earthquake." ScienceDaily 1 September 2003. 31 March 2010
<<http://www.sciencedaily.com/releases/2003/08/030829071707.htm>>.
- [12] T. Fukuda and S. Nakagawa, "A dynamically reconfigurable robotic system," in Proceedings of the International Conference on Industrial Electronics, Controls, and Instrumentation, vol. 2, 1987, pp. 588-595.

- [13] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level Measurements from an 802.11b Mesh Network. In Proceedings of SIGCOMM, Aug. 2004.
- [14] J. Heidemann, N. Bulusu, and J. Elson. Effects of Detail in Wireless Network Simulation. In Proceedings of the SCS Multiconference on Distributed Simulation, Jan. 2001.
- [15] M. Takai, J. Martin, and R. Bagrodia. Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks. In Proceedings of ACM MobiHoc, Oct. 2001.
- [16] A. Woo, T. Tong, and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *SenSys*, Nov. 2003.
- [17] P. De, A. Raniwala, S. Sharma, and T. Chiueh. MiNT: A Miniaturized Network Testbed for Mobile Wireless Research. In *Proceedings of IEEE INFOCOM*, Mar. 2005.
- [18] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the ORBIT Radio Grid Testbed for Evaluation of Next Generation Wireless Network Protocols. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, Mar. 2005.
- [19] B. N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D. C. Parkes, J. Shneidman, A. C. Snoeren, and A. Vahdat. Mirage: A Microeconomic Resource Allocation System for SensorNet Testbeds. In *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors*, Sydney, Australia, May 2005.
- [20] Amir Reza Momen, Jahangir Dadkhah Chime and Rey Branch. A New Analytical Method in User Mobility Modeling in Wireless Network. In Proceedings of the International Conference on Information and Communication Technologies: From theory to applications, 2004.
- [21] Tracy Camp, Jeff Boleng and Vanessa Davis. A Survey of Mobility Models for Ad Hoc Network Research. In *Wireless Communications and Mobile Computing: Special Issue on Mobile Ad Hoc Networking: research trends and applications*, vol 2 no. 5 483-502, 2002.
- [22] J. Broch, D. A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pages 85–97, Dallas, Texas, October 1998.
- [23] V. Davies. Evaluating Mobility Models Within an Ad hoc Network. Master’s thesis, Colorado School of Mines, 2000.
- [24] Z. Haas. A New Routing Protocol for Reconfigurable Wireless Networks. In Proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC), pages 562–565, October 1997.

- [25] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A Group Mobility Model for Ad hoc Wireless Networks. In Proceedings of the ACM/IEEE MSWIM'99, Seattle, WA, August 1999.
- [26] B. Liang and Z. Haas. Predictive Distance-based Mobility Management for PCS Networks. In Proceedings of the IEEE Conference on Computer Communication (INFOCOM), New York, NY, March 1999.
- [27] C. Bettstetter. Smooth is Better than Sharp: A random Mobility Model for Simulation of Wireless Networks. In Proceedings of the 4th ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWIM), Rome, Italy, July 2001.
- [28] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.
- [29] R. A. Guerin. Channel Occupancy Time Distribution in a Cellular Radio System. *IEEE Transactions on Vehicular Technology*, 36(3):89–99, 1987.
- [30] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser. An Analysis of the Optimum Node Density for Ad hoc Mobile Networks. In Proceedings of the IEEE International Conference on Communications, pages 857–861, Helsinki, Finland, June 2001.
- [31] J. Broch, D. A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pages 85–97, Dallas, Texas, October 1998.
- [32] C. Schindelhauer, Mobility in wireless networks. In the 32nd Annual Conference on Current Trends in Theory and Practice of Informatics, Czech Republic, 2006
- [33] S. Sail. On the Applicability of Random Mobility Models for Swarm Robot Movements, Master's Thesis, Rochester Institute of Technology, 2007.
- [34] J. A. Boyan, and A. W. Moore. "Generalization in reinforcement learning safely approximating the value function", in *Advances in Neural Information Processing Systems* 7. 1995.
- [35] W. D. Smart, and L. P. Kaelbling. "Practical reinforcement learning in continuous spaces", in proceedings of the 7th International Conference on Machine Learning. 2000.
- [36] Zheng Liu; Ang, M.H., Jr.; Seah, W.K.G., Reinforcement learning of cooperative behaviors for multi-robot tracking of multiple moving targets," *Intelligent Robots and Systems*, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on , vol., no., pp. 1289-1294, 2-6 Aug. 2005.

- [37] R. Sim and G. Dudek, .Learning visual landmarks for pose estimation. In Proc. of the IEEE International Conference on Robotics & Automation (ICRA), 1999.
- [38] S. Boga. Vision-Enhanced Localization for Cooperative Robotics. Master's Thesis, Auburn University, 2010.
- [39] A. Ray.Cooperative Robotics using Wireless Communication. Master's Thesis, Auburn University, 2005.
- [40] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," in Proceedings of ICAR International Conference on Advanced Robotics, 1997, pp. 193-200.
- [41] M. Vainio, P. Appelqvist, and A. Halme, "Generic control architecture for a cooperative robot system," in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, 1998, pp 1119-1125.
- [42] Cooperative Robotics Research Team's Home Page: <http://crr.eng.auburn.edu> .
- [43] STRAW - STreet RAndom Waypoint - vehiclar mobility model for network simulations Home Page: <http://www.aqualab.cs.northwestern.edu/projects/STRAW/>.
- [44] M. Fiore, "Mobility models in inter-vehicle communications literature," Tech. Rep., Nov.2006.
- [45] R. Wood,Fly, robot, fly. IEEE Spectrum 45(3):25–29, 2008.
- [46] G. Miller, Snake Robots Home Page: <http://www.snakerobots.com/about.html>.
- [47] M. Dorigo, Ant colony optimization Home Page: iridia.ulb.ac.be/mdorigo/ACO/ACO.html.
- [48] David Payton, Mike Daily, Regina Estowski, Mike Howard and Craig Lee. Pheromone Robotics. In Autonomous Robots 11, 319324, 2001, c 2001 Kluwer Academic Publishers.
- [49] C. Wilson, Hardware Testbed for Collaborative Robotics using Wireless Communication. Masters Thesis, Auburn University, 2009.