iCCA-MAP: A Mobile Node Localization Algorithm for Wireless Sensor Networks

by

Shafagh Alikhani

A thesis submitted to the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering (OCIECE)

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario, Canada, K1S 5B6

May 2010

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis

iCCA-MAP: A Mobile Node Localization Algorithm for Wireless Sensor Networks

submitted by

Shafagh Alikhani, B. Eng.

in partial fulfillment of the requirements for

the degree of Master of Applied Science

Chair, Dr. Howard Schwartz, Department of Systems and Computer Engineering

---

Thesis Supervisor, Dr. Thomas Kunz

Thesis Supervisor, Dr. Marc St-Hilaire

Carleton University
May 2010

Abstract

Wireless Sensor Networks (WSNs) consist of spatially distributed wireless sensor nodes that cooperate with each other in order to monitor and collect data pertaining to physical or environmental conditions such as temperature, pressure, motion, sound, and other phenomena. The locations of the sensor nodes are not predetermined as they are usually randomly deployed in the region of interest. Therefore, algorithms that can compute the location of sensor nodes within a WSN are needed.

A new and efficient, accurate, and cost-effective algorithm, called iCCA-MAP, has been proposed for localizing mobile node(s) within a WSN. The proposed algorithm is based on the CCA-MAP algorithm, which applies an efficient nonlinear data mapping technique. Simulation results show that the localization error for both CCA-MAP and iCCA-MAP are similar. However, the computational time required for obtaining location estimates using iCCA-MAP is far smaller than that of the original CCA-MAP. Therefore, iCCA-MAP can be applied at closer time intervals in order to provide up-to-date estimates of the mobile node's location, which can result in a lower localization error. For a complete performance evaluation, iCCA-MAP has been compared to the well-known mobile node localization algorithms MCL and a variation of MCL, called Dual MCL. Simulation results show that iCCA-MAP outperforms MCL and Dual MCL by having a lower localization error when the minimum number of anchor nodes is used.

Acknowledgements

First and foremost I would like to thank my supervisors Dr. Thomas Kunz and Dr. Marc St-Hilaire for their immense guidance and support. This work would not have been possible without their continuous direction and feedback.

I would like to thank Dr. Li Li for her insightful discussions on CCA-MAP. I am also grateful to the administrative and technical staff of the Department of Systems and Computer Engineering for their assistance throughout my Master's program at Carleton University.

I would like to extend my thanks to all my friends and colleagues who have provided me with invaluable assistance, support, and inspiration throughout the course of my studies.

Finally, I would like to thank my family for their enormous support, strength, and encouragement.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ALE	Average Localization Error
AoA	Angle of Arrival
CCA	Curvilinear Component Analysis
CCA-MAP	Curvilinear Component Analysis-Map
CDL	Color-theory-based Dynamic Localization
DOI	Degree of Irregularity
Dual MCL	Dual Monte Carlo Localization
DV	Distance Vector
E-CDL	Enhanced Color-theory-based Dynamic Localization
EMAP	Extended Mobile Anchor Point
GPS	Global Positioning System
iCCA-MAP	Iterative Curvilinear Component Analysis-Map
IMCL	Improved Monte Carlo Localization
MAL	Mobile Assisted Localization
MAP	Mobile Anchor Point
MCF	Maximum Common Nodes First
MCL	Monte Carlo Localization
MDS	Multi-Dimensional Scaling
MDS-MAP	Multi-Dimensional Scaling-Map
NLM	Non-Linear data Mapping
RF	Radio Frequency
RGB	Red Green Blue
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
TDoA	Time Difference of Arrival
ToA	Time of Arrival
WSN	Wireless Sensor Network

Chapter 1

Introduction

Recent advancements in electronics and wireless communication technology has led the way to the development of tiny, low-power, low-cost sensor nodes which have the ability to sense physical phenomena, process data, and communicate with one another. A large number of these wireless sensor nodes are deployed across a geographical region to form a wireless sensor network (WSN). These WSNs create smart environments by providing access to information regarding the environment through collecting, processing, analyzing, and disseminating data whenever required. In order to use WSNs in inaccessible terrains or disaster relief operations, random deployment of the sensor nodes is required. As a result, the position of these nodes will not be predetermined and thus the nodes must have the ability to collaborate with each other to form self-organized networks in order to perform tasks including but not limited to determining their location [1].

Sensor nodes consist of sensing hardware, processor, memory, power supply, and a transceiver. These sensor nodes have limited energy, memory, and computational power and thus, a large number of them are required in order to be effective. Denser WSNs can provide higher accuracy and have a larger total amount of energy available [2].

Examples of possible applications of WSNs include environmental monitoring, specifically for planetary exploration, geophysical monitoring, habitat monitoring, wildlife tracking, and oceanography. Another promising application of WSNs is in the field of physiological monitoring and medical sensing. WSNs can also be used in smart homes and offices where they can efficiently regulate light, temperature, and humidity based on predefined individual preferences. Other uses of WSNs are in inventory tracking, precision agriculture, disaster detection, search and rescue operations, and commercial and residential security. In a military context, WSNs can be used for surveillance and battle-space monitoring. WSNs can also be used in transportation such that vehicles equipped with wireless sensors form local communication networks. These networks allow the vehicles to share information on weather and road conditions, plan routes, avoid traffic, and identify their position in areas where Global Positioning System (GPS) signals are unavailable.

In this chapter, the motivation behind this work is first provided. Then, the problem statement is described, followed by the objectives of this research. Finally, the main contributions and the overall organization of the thesis are outlined.

1.1 *Motivation*

Identifying the location of the nodes in a WSN is of great importance, given that without location information we would not know where in the network the collected data is coming from. As a result, the collected data would become less meaningful, and therefore, we would not make effective use of our WSN. Knowing a node's location is also required for many network protocols and middleware services that rely on location information, such as geographic routing protocols [3][4][5], context-based routing protocols [6][7], location-aware services [8], and enhanced security protection mechanisms [9].

In the literature, substantial research on location estimation of stationary nodes in WSNs has been presented [10]-[25]. However, not as much research pertaining to mobility in WSNs has been conducted. Recently however, the subject of mobility in WSNs has gained much interest due to the increasing number of applications that require mobile sensor nodes. Animal tracking, logistics applications, and elderly healthcare home monitoring are but a few of such applications.

Studies conducted on introducing mobility in WSNs have resulted in an overall improvement not only by increasing the overall network lifetime, but also by improving the data capacity of the network as well as addressing delay and latency problems [26] [27] . As a result, many researchers have started to investigate the concept of mobility in WSNs. A number of algorithms that can estimate the location of mobile nodes within WSNs have been developed [28]. There are however not very many localization algorithms that address the localization of nodes in WSNs containing a mixture of mobile and stationary nodes.

1.2 ***Problem Statement***

The process of determining the location of wireless sensor nodes in a WSN is called localization. The measured distance between two nodes and/or connectivity information amongst the nodes, and sometimes the known locations of other nodes are used for estimating the location of the sensor nodes. Nodes with known locations are called *anchor nodes* and nodes whose locations are to be determined are simply known as sensor nodes. Coordinates sought can be either global or relative. Global coordinates are aligned with a recognized system such as the Global Positioning System (GPS), while relative coordinates are simply a random transformation (rotation, reflection, and translation) of the global coordinate system.

Due to the context of the application and the potential for a high number of wireless sensor nodes, either manually configuring location information into each node or equipping every node with a GPS receiver becomes expensive and infeasible, as GPS may not work in all scenarios, for instance, in environments that do not have direct line-of-sight with the satellites that provide GPS signals. Therefore, to report data that has geographical significance, sensor nodes need to be able to estimate their own locations. The accuracy of the estimated node locations depends on the precision of the distance measurements between the nodes and the complexity of the algorithms used. The required accuracy level of node location estimations is ultimately determined by the application or protocol requirements. For example, in applications such as human sensor networks very high node location accuracy may be required, whereas in other applications, such as wild-life tracking, more lenient location estimations could be acceptable.

Localization algorithms for applications in which there exist both mobile and stationary nodes need to be developed. Such applications include scenarios in which there are one or more mobile agents and a large number of stationary nodes. One such scenario is homecare for the elderly; here stationary nodes can be placed all around the home of the patient and the sensor(s) on the patient are the mobile nodes. The stationary nodes provide information regarding temperature, humidity, movement, etc., and the mobile sensor(s) on the patient provide information regarding the patient and her whereabouts. Another example is that of fire rescue, where the stationary nodes are deployed in the area where the fire has occurred, providing local environmental conditions. The mobile nodes would be attached to the firemen, providing information regarding their conditions and location. A third example would be for tracking animals that travel vast distances. Stationary nodes are required in order to keep the network connected, nodes on the animals would be the mobile nodes and stationary nodes would be deployed in the regions in which the animals roam.

In order to minimize the cost, the algorithm should, ideally, not rely on additional hardware. To remain cost-effective and to simplify deployment, the algorithm should only require a minimum number of anchor nodes, namely three anchor nodes for 2-dimensional space and four anchor nodes for 3-dimensional space. Anchor nodes usually have a GPS mounted on them and may require higher computational capabilities and additional battery life, as GPS can require a higher energy consumption. For the case in which anchor nodes are pre-configured with their location information, having less anchor nodes eases the task of deployment.

1.3 *Research Objectives and Contributions*

The main objective of this research is to develop a localization algorithm in order to locate mobile nodes in wireless sensor network. More precisely, we want to:

1. Expand an existing algorithm, called CCA-MAP [25], such that it can localize mobile sensor nodes;

2. Compare the results of the proposed algorithm with the results of CCA-MAP to evaluate the effectiveness of the proposed algorithm;

3. Compare the results of the proposed algorithm with novel algorithms available in the literature.

Based on these objectives, the following contributions have been made.

1. iCCA-MAP, an efficient and accurate localization algorithm for mobile node(s) in WSN has been developed.

2. The computational time of iCCA-MAP has been compared to that of CCA-MAP, the original algorithm for stationary WSNs from which iCCA-MAP has been developed. Results show that iCCA-MAP is by far a more efficient algorithm that can be used for providing near real-time location estimation for mobile node(s) in WSNs. Based on these results, the following refereed conference paper has been published:

   S. Alikhani, M. St-Hilaire, and T. Kunz, "iCCA-MAP: A New Mobile Node Localization Algorithm", in *Proceedings of* the *5th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications* (WiMob 2009), Marrakech, Morocco, October 2009.

3. The location accuracy of iCCA-MAP has been compared to that of MCL [29] and Dual MCL [30]. Results show that iCCA-MAP outperforms both algorithms using a minimum number of anchor nodes. Based on these results, the following paper has been submitted to:

   S. Alikhani, T. Kunz, and M. St-Hilaire "iCCA-MAP versus MCL and Dual MCL: Comparison of Mobile Node Localization Algorithms for Wireless Sensor Networks", *9th International Conference on Ad Hoc Networks and Wireless* (ADHOC-NOW 2010), Edmonton, Canada, August 2010.

1.4 *Thesis Organization*

This thesis is organized as follows. Chapter 2 provides a classification of localization techniques and reviews the state-of-the-art in WSN localization. In Chapter 3, iCCA-MAP, the proposed localization algorithm for mobile node(s) in WSNs is introduced. Results and discussions are presented in Chapter 4, and Chapter 5 concludes the thesis and discusses future work.

Chapter 2

Localization Algorithms in WSNs

In this chapter, the state-of-the-art in localization algorithms for WSNs is reviewed. First, in Section 2.1 a classification of localization techniques for WSNs is provided. Then, in Section 2.2 the CCA-MAP algorithm, which is used to localize stationary nodes, is introduced. Section 2.3 presents other algorithms that are especially designed for mobile nodes and mobile anchor nodes. Finally, in Section 2.4 a performance analysis is conducted in order to compare the different algorithms and to identify a competitive solution that can be used for comparison against the proposed algorithm.

2.1 *Classification of Localization Techniques in WSNs*

In this section, a classification of localization techniques in WSNs is provided. Centralized and distributed techniques are described in Section 2.1.1. In Section 2.1.2, range-free and range-based techniques are presented. Section 2.1.3 introduces anchor-free and anchor-based techniques and finally, Section 2.1.4 introduces mobile and stationary localization techniques.

2.1.1 Centralized versus Distributed Localization Algorithms

Localization algorithms can be categorized as centralized [10][12] or distributed [14] [23] [31] algorithms based on their computational organization. In centralized algorithms, nodes send data to a central location where computation is performed and the location of each node is determined and sent back to the nodes. The drawbacks of centralized algorithms are their high communication costs and intrinsic delay. In most cases, the intrinsic delay of centralized algorithms increases as the number of nodes in the network increases, thus making centralized algorithms inefficient for large networks. As a result, distributed algorithms that distribute the computational load across the network to decrease delay and to minimize the amount of inter-sensor communication have been introduced [12]. In distributed algorithms, each node determines its location by communication with its neighbouring nodes. Generally, distributed algorithms are more robust and energy efficient since each node determines its location locally with the help of its neighbours, without the need to send and receive location information to and from a central server. Distributed algorithms however can be more complex to implement and at times may not be possible due to the limited computational capabilities of sensor nodes.

2.1.2 Range-Free versus Range-Based Localization Techniques

For determining the location of a sensor node, two types of techniques exist: range-free [19][20][21] and range-based [16][18][22][33]. Range-free techniques use connectivity information between neighbouring nodes to estimate the nodes' position, range-based techniques however require ranging information that can be used to estimate the distance between two neighbouring nodes. On the one hand, range-free techniques do not require any additional hardware and use proximity information to estimate the location of the nodes in a WSN, and thus have limited precision. On the other hand, range-based techniques use range measurements such as time of arrival (ToA), angle of arrival (AoA), received signal strength indicator (RSSI), and time difference of arrival (TDoA) to measure the distances between the nodes in order to estimate the location of the nodes. These different ranging techniques are described as follows.

Time of Arrival

In the Time of Arrival (ToA) technique, all sensors transmit a signal with a predefined velocity to their neighbours. Then, the nodes each send a signal back to their neighbours and by using the transmission and received times each node estimates its distance to its neighbour [32].

Received Signal Strength Indicator

Received Signal Strength Indicator (RSSI) is defined as the amount of power present in a received radio signal. Due to radio-propagation pathloss, received signal strength (RSS) decreases as the distance of the radio propagation increases. Therefore, the distance between two sensor nodes can be compared using the RSS value at the receiver, assuming that the transmission power at the sender is either fixed or known [32].

An advantage of this technique is that no additional hardware is required as it uses a standard feature found in most wireless devices, namely the received signal strength indicator. Also it does not significantly impact local power consumption or sensor size and thus cost. The disadvantage of this technique is its inaccuracy. For example, if the sensor network is deployed indoors, walls and other obstacles would severely reduce the precision of the method due to nonlinearities, noise, interference, and absorption.

Time Difference of Arrival

The Time Difference of Arrival (TDoA) technique requires the nodes to transmit two signals that travel at different speeds. In this technique, each node is equipped with a microphone and a speaker. Most systems use ultrasound while some use audible frequencies. In TDoA, a radio message is sent by the transmitter, which then waits some fixed interval of time and then produces a fixed pattern of chirps on its speaker. In listening mode, the nodes hear the radio signal and note the current time, and then they turn on their microphones to detect the chirp pattern and again note the current time. Once they have the different times, the nodes can compute the distance between themselves and the transmitter using the fact that radio waves travel much faster than sound in air [32].

If line-of-sight conditions are met and the environment is echo-free, TDoA techniques perform extremely accurately. The disadvantage of such systems is that they require special hardware which must be built into the sensor nodes. Also, the speed of sound in air varies with air temperature and humidity, which can introduce inaccuracies. Lastly, it is very difficult to meet line-of-sight conditions in many environments such as inside buildings or in mountainous terrains.

Angle of Arrival

Angle of Arrival (AoA) techniques gather data using either radio or microphone arrays. These arrays allow a receiving node determine the direction of a transmitting node. Optical communication techniques can also be used to gather AoA data. In these techniques, a single transmitted signal is heard by several spatially separated microphones. The phase or time difference between the signal's arrival at different microphones is calculated and thus the AoA of the signal is found.

This technique is accurate to within a few degrees but the downside is that AoA hardware is bigger and more expensive than TDoA ranging hardware, since each node must have one speaker and several microphones. Another important factor is the need for spatial separation between speakers which will be difficult to accommodate as the size of sensor nodes decreases [32].

In conclusion, range-based techniques can provide very accurate results but require expensive hardware, such as ultrasound devices for TDoA and antenna arrays for AoA. A disadvantage of range-based techniques is that distance information can be difficult to obtain in practice due to issues such as lack of omni-directional ranging and presence of obstacles which prevent line-of-sight.

2.1.3 Anchor-Based versus Anchor-Free Localization Techniques

Another classification of localization algorithms for WSNs is based on whether or not external reference nodes are needed. These nodes, called anchor nodes (or simply anchors for short), usually either have a GPS receiver installed on them or know their position by manual configuration. They are used by other nodes as reference nodes in order to provide coordinates in the absolute reference system being used.

Anchor-based algorithms [13][14][17][25] use anchor nodes to rotate, translate and sometimes scale a relative coordinate system so that it coincides with an absolute coordinate system. In such algorithms, a fraction of the nodes must be anchor nodes or at least a minimum number of anchor nodes are required for adequate results. For 2-dimensional spaces, at least three noncollinear anchor nodes and for 3-dimensional spaces, at least four noncoplanar anchor nodes are required. The final coordinate assignments of the sensor nodes are valid with respect to a global coordinate system or any other coordinate system being used. A drawback to anchor-based algorithms is that another positioning system is required to determine the anchor node positions. Therefore, if the other positioning system is unavailable, for instance, for GPS-based anchors located in areas where there is no clear view of the sky, the algorithm may not function properly. Another drawback to anchor-based algorithms is that anchor nodes are expensive as they usually require a GPS receiver to be mounted on them. Therefore, algorithms that require many anchor nodes are not very cost-effective. Location information can also be hard-coded into anchor nodes, however, in this case careful deployment of anchor nodes is required, which may be very expensive or even impossible in inaccessible terrains.

In contrast, anchor-free localization algorithms [23][39] do not require anchor nodes. These algorithms provide only relative node locations, i.e., node locations that reflect the position of the sensor nodes relative to each other. For some applications, such relative coordinates are sufficient, however. For example, in geographic routing protocols, the next forwarding node is usually chosen based on a distance metric that requires the next hop to be physically closer to the destination, which can be perfectly expressed with relative coordinates.

2.1.4 Mobile versus Stationary Node Localization

The problem of mobility in WSNs has recently gained much interest as the number of applications that require mobile sensor nodes has increased. Studies conducted on introducing mobility in WSNs have resulted in an overall improvement in the network by not only increasing the overall network lifetime, but also by improving the data capacity of the network as well as addressing delay and latency problems [26][27]. Some authors have proposed algorithms in which mobile anchor nodes are used in order to aid with the localization of stationary sensor nodes [34] [35][36]; inventory management is an example of an application that takes advantage of such an approach. In other scenarios however, some or all of the sensor nodes are mobile [28][29][30][37][38][39][40]; this is where "mobility creates the problem of locating and tracking moving sensors in real time" [41]. In Section 2.3, algorithms that employ mobile anchor nodes for finding stationary sensor node locations, algorithms that locate sensor nodes that are mobile, and localization algorithms in which both anchors and sensor nodes are mobile are reviewed.

2.2 *Localization Algorithms Proposed for Stationary WSNs: CCA-MAP*

Many localization algorithms have been proposed for stationary WSNs [10]-[25]. However, in this section, the focus is on the CCA-MAP algorithm, which is the basis of the proposed iCCA-MAP. To the best of our knowledge, the CCA-MAP algorithm is amongst the highest performing algorithms proposed for stationary WSNs.

CCA-MAP, proposed by Li and Kunz [25], is a cooperative node localization algorithm, which applies an efficient non-linear data mapping technique called Curvilinear Component Analysis (CCA) [42], to localize nodes in a WSN. Cooperative localization schemes formulate the localization problem as a joint estimation problem and apply optimization techniques to derive location coordinates considering all constraints on internode distances, rather than considering only constraints between the sensor nodes and anchor nodes.

CCA, a self-organizing neural network, was originally proposed for dimensionality reduction and representation of multidimensional data sets. The self-organized neural network performs vector quantization of the input space and nonlinear projection of the quantized vectors toward an output space, mapping the data sets in a higher dimension to a lower one.

CCA-MAP uses CCA to build local maps at every possible node of the network and patches them together to form a global map. Each node uses only local information to compute its own local map. Here, CCA is employed for computing the node coordinates in the local map.

In the CCA-MAP algorithm, for networks with less than 1000 nodes, neighbours within $h=2$ hops are included in building the local map for each node. For networks with more than 1000 nodes, neighbours within $h=1$ hops suffice. The shortest distance matrix of the local map is computed and used as the approximate distance matrix. The shortest distance matrix of each local map is derived using the least number of hops from every node to every other node in the local map if the range-free version of CCA-MAP is being used, or it can be derived from distance measurements between the nodes if the range-based version of CCA-MAP is being used. Then, each node applies the CCA algorithm, generating the relative coordinates of each node in its local map. The complexity of computing each local map is $O(k^2)$, where k is the average number of 2-hop neighbours. The local maps are then merged. The merged map transforms to an absolute map based on positions of the anchor nodes. While for 2-dimensional space three anchor nodes are the minimum required, simulation results for CCA-MAP have shown that the algorithm can achieve very accurate localization performance by using only this minimal set of three anchor nodes [25]. In the merging step, for the starting map, the local map of a randomly selected node is used. After that, the neighbour node whose local map shares the most nodes with the current map is selected to merge its local map into the current map. Using the coordinates of their common nodes, two maps are merged. A linear transformation is used for merging a new local map into the current map. This scheme allows local maps to be merged in parallel in different parts of the network if CCA-MAP is implemented in a distributed fashion, otherwise the maps are merged in sequence. The complexity of the map merge step is $O(k^3n)$, where k is the average number of 2-hop neighbours and n is the total number of nodes.

Anchor nodes are not required while merging the maps. However, to obtain the absolute coordinates of the nodes, when at least three anchor nodes are found in the patched map of a subnetwork, an absolute map can be computed using the coordinates of the anchor nodes. The complexity of this step for a anchor nodes is $O(a^3+n)$. The CCA-MAP algorithm can be carried out in a distributed fashion where each node computes its local map or it can be carried out at more powerful gateway nodes of clusters if the sensor network uses a hierarchical structure or at a central computing platform if desired.

There are other localization algorithms such as MDS-MAP [12][13][14] and others [24][43] that generate local maps as well, but global maps are all constructed in a similar fashion. In two dimensions, three or more common nodes are needed to rigidly stitch two maps together by translating, rotating, and/or reflecting one map in order to place the corresponding nodes as close to each other as possible.

The order by which the local maps are merged can be either peer-to-peer or incremental [24]. In the peer-to-peer method, any two maps can be merged as long as they have enough common nodes. Therefore, there is no need for a global schedule because the stitching occurs concurrently throughout the whole network. In incremental stitching, a single map is declared as the core map and other maps are stitched to it one at a time. In [14] and [15], Maximum Common Nodes First (MCF) stitching, an algorithm which finds the map that has the maximum number of common nodes with the core map, is implemented. In [24], the authors propose a method that utilizes all available distances between two maps in order to prevent flip errors and provide better accuracy.

2.3 *Localization Methods Proposed for Mobile WSNs*

In this section, algorithms addressing mobility in WSNs are reviewed. These algorithms are divided into three subsections: algorithms requiring mobile anchor nodes which aid in finding stationary node locations; algorithms proposed for mobile sensor nodes; and algorithms in which both anchor nodes and sensor nodes are mobile.

2.3.1 Algorithms for Mobile Anchor Nodes

In mobile-assisted localization (MAL) [34], a mobile user is utilized to assist with measurements between node pairs. Measurements are made until the distance constraints, which depend on the number of visible nodes in a given region, form a globally rigid structure guaranteeing a unique localization. The mobile's movement and the minimum number of measurements it must collect determine the required constraints. In this approach, "a roving human or a robot wanders through an area, collecting distance information between the nodes and itself" [34]. The performance of MAL is evaluated for indoor environments using the Cricket location system [18].

In [35], a localization technique using a single mobile anchor node, aware of its position by use of a GPS receiver, is presented. Proximity constraints, which are inferred by the sensor nodes receiving data packets from the mobile anchor node, are used to construct and maintain position estimates. In this method, Bayesian inference is used. Here again the mobile anchor node can be a human operator, an unmanned vehicle, or a plane from which the nodes are deployed. This method is radio frequency (RF) based, meaning that the received signal strength indicator is used for ranging measurements, although other ranging measurement methods could also be used. The disadvantage of using RSSI is its low accuracy, given that noise, interference, and absorption can significantly affect the precision, especially in indoor environments where walls are major obstacles.

In [36], Galstyan *et al.* propose a distributed online algorithm wherein nodes use geometric constraints caused by radio connectivity as well as sensing information to decrease the uncertainty of their location. A moving target provides commonly sensed data as a means of sensing constraints, where these constraints are tighter than the radio connectivity constraints. First, a moving anchor node is used to dynamically self-configure a network, broadcasting its coordinates to nodes that are within its sensing range, as it moves through the network. A new quadratic constraint is generated every time a node senses the anchor node. The constraint is used to further reduce the uncertainty in the nodes' positions. The sensing region is approximated by a rectangular bounding box, hence replacing the quadratic constraint by a simpler linearized constraint which guarantees that the actual position of the node is within the bounding box. The authors then use a moving target with unknown coordinates. The target can be localized if there are enough nodes that know their approximate positions. The information received from the target can be used to impose new constraints on the position of the node, improving the node's position accuracy.

Triangulation is one of the most popular methods for target localization, but it requires at least three sensors with known locations in 2D, which sometimes cannot be achieved. For this reason, the authors use a bounding rectangle for the target which guarantees that the actual target position is always inside the bounding box. Nodes can use target information in the following two ways: (i) constraints are imposed on a node's position by observing the target, and (ii) if a target that is in the vicinity of a node is not detected by the node but detected by the node's neighbours, the node can use this "negative" information to impose tighter constraints on its own position. A major drawback with this algorithm is the time it may take for the mobile anchor node to broadcast its coordinates to nodes within its sensing range, which could depend on the size of the network, the deployment area, and the anchor node's radio range. Another concern is the time that may be required for the target to localize if only a few nodes within its vicinity know their positions.

Another algorithm that uses mobile anchor nodes for achieving node localization in static sensor networks is called Mobile Anchor Point (MAP) [28]. In this algorithm, anchors broadcast beacon messages with their location information as they move through the sensor field. This algorithm supposes that the range of communication between the anchors and nodes is bounded by a circle and the node is located at the center of this circle. The node uses the locations of the moving anchors that are chosen in order to form the chord of the circle. The geometric corollary stating that a perpendicular bisector of a chord passes through the center of the circle is used for determining the location of the node by calculating the intersection point of two perpendicular bisectors of the cords. A visitor list storing both the mobile anchors whose messages have been received by the sensor node and their associated lifetime is maintained by each sensor node. The node checks whether the anchor node has been inserted in its visitor list every time it receives a beacon message from a mobile anchor. If the anchor node hasn't been inserted, the beacon message will be selected as a beacon point and a predefined lifetime for the anchor will be added in the visitor list. If the mobile anchor node already exists in the node's visitor list, its lifetime is prolonged. When the lifetime of an anchor node expires, the last beacon message of that anchor node is recorded as a beacon point and the corresponding entry in the visitor list is removed. The accuracy of this algorithm is dependent on the speed of the mobile anchor nodes and the number of mobile anchor nodes available in the network.

2.3.2 Algorithms for Mobile Sensor Nodes

The Color-theory-based Dynamic Localization (CDL) algorithm [38] is a centralized localization algorithm which uses color theory in order to localize nodes in mobile WSNs. The function of this algorithm is to build a location database in the centralized server in order to map a set of red-green-blue (RGB) values to a geographic position. This algorithm works as follows: a sensor node receives RGB values from anchor nodes and, using these values, it calculates its own RGB values. It then sends its RGB values to the server, where the closest possible location is queried in the database. The distance measurements in the CDL algorithm are based on the DV-hop scheme, in which the derived shortest path is generally larger than the corresponding Euclidean distance, which can cause inaccuracy in location estimates.

Yu and Yu [39] propose a range-based, anchor-free algorithm using rigidity theory for localization of mobile nodes. Their algorithm is based on the following assumptions:

- Each node has a constant speed and has a random initial location;
- The initial position of each node is known, either by use of a stationary localization algorithm or by initial node placement;
- There are at least two neighbours in the transmission range of each node;
- Techniques such as RSSI, ToA, TDoA, and AoA can be used to find the distance between any two sensor nodes.

By solving for the three unknown node locations using four equations, the final position of the nodes are found if there exists a unique solution. If there are multiple solutions to the problem, other pairs of neighbouring nodes that are also neighbours to each other must be used until a unique solution is found. A downside to this approach is the amount of memory required to keep track of all neighbouring nodes and the neighbouring nodes' neighbours, and on top of that, all predicted positions for a node must also be kept in memory until a final unique solution is determined.

2.3.3 Algorithms for Mobile Sensor Nodes and Mobile Anchor Nodes

One of the early works on locating mobile nodes with the aid of mobile anchor nodes in the context of range-free localization is by Hu and Evans [29], who introduced the sequential Monte Carlo Localization (MCL) method for mobile WSNs. More specifically, the authors show that such a method utilizing mobility can improve accuracy and reduce localization costs. Their method is based on adopting the Monte Carlo Localization (MCL) algorithm developed for localization in robotics. In this method, the posterior distribution of possible locations as the valid solutions is represented by a set of weighted samples.

Initially, nodes have no knowledge of their location and thus a set of N random locations in the deployment area are chosen as initial samples. Each step consists of two phases, referred to as the prediction and filtering phase. In the prediction phase, a movement occurs and the uncertainty increases, whereas, in the filtering phase, updating the data is done based on new observations. In the prediction phase, the node calculates its possible locations based on the previous possible locations and its maximum velocity. Anchor nodes then transmit their location information, and, based on the nodes' observation of the anchor node locations, samples inconsistent with observations are filtered out. There are two types of observations: direct anchor observations and indirect anchor observations. In a direct anchor observation, if the node hears the anchor, it must lie within a circle of radius r of the anchor's location. An indirect anchor observation is when a node does not hear an anchor but one of its neighbours does, thus indicating that the node must lie within distance r and $2r$ of the anchors location. This process is repeated until a satisfactory estimate of the nodes' locations is made.

In [30], Stevens-Navarro *et al.* propose and analyze two variations of the Monte Carlo Localization algorithms, calling them Dual MCL, which can be considered as the logical inverse of the original MCL algorithm, and Mixture MCL, which is a combination of the original MCL and Dual MCL. Simulations in which the number of seeds, nodes, samples, velocity of nodes, and radio pattern degree of irregularity are varied are performed. The authors report that both Dual MCL and Mixture MCL are more accurate than the original MCL algorithm. However, in terms of a trade-off between the computational time and estimated location accuracy, Mixture MCL outperforms both the Dual and the original MCL algorithms.

In the Dual MCL, the prediction and filtering steps of the original MCL algorithm are rather inverted. In the prediction step of Dual MCL, samples are generated from the deployment area and are validated based on anchor nodes being heard by nodes or by their neighbouring nodes. In the Dual MCL filtering step, at every time interval, the validated predicted samples are filtered based on the previous location of the node and the maximum velocity the node can travel.

The Mixture MCL algorithm combines the Dual MCL and the original MCL algorithm by generating samples using both methods and mixing these samples together using a mixing rate. The results generated using Mixture MCL is not as accurate as that of Dual MCL, but in terms of the trade off between computational time and estimated location accuracy, Mixture MCL outperforms Dual MCL. One major drawback of the MCL family of algorithms is their requirement for high anchor node density.

Extended Mobile Anchor Point (EMAP) [37] is an extension of the MAP algorithm [28] adapted for mobile sensor nodes. The MAP algorithm uses mobile anchors in a static WSN, whereas EMAP extends this algorithm to deal with mobile sensor networks. In this algorithm, anchor nodes broadcast their location information as they move through the sensing field. The corollary of the perpendicular bisector of a chord is used for predicting a node’s location. EMAP assumes that mobile nodes know their distance and moving direction using measuring devices such as compass and odometer. A major drawback of this algorithm is the assumption that mobile nodes have knowledge of their distance and moving direction. This assumption necessitates the use of additional hardware which increases cost.

Enhanced Color-theory-based Dynamic Localization (E-CDL) proposed in [40] is based on the CDL algorithm [38], whose location accuracy depends on the accuracy of the average hop distance derivation. Here the authors employ mobile anchor nodes rather than stationary ones in order to enhance the accuracy of the measurements, as well as to decrease the possibility of sensor node isolation in the multihop environment. The anchor nodes are placed in the four corners of the square field and move a distance of a radio range (r) in every time slot. This results in a decrease of the hop count for some nodes to the anchors and an increase for others. Mobility of the anchor nodes is intended to resolve the problem of possible disconnections from the network. Another improvement to the CDL algorithm is the introduction of two new methods for calculating the average hop distance measurements; the first calculates the expected value of the next hop distance based on the next hop being located between $0.5r$ and r (r being the radio range) and the second proposed method adjusts the average hop distance based on the ratio of the Euclidean distance to the shortest path length. The accuracy of E-CDL is dependent on the granularity of the RBG model used in the algorithm.

2.4 *Performance Analysis*

Focusing on mobile WSNs, this section presents a brief performance analysis of the existing algorithms for such networks. This analysis will eventually help us select the best candidate for comparison with iCCA-MAP, our proposed algorithm.

2.4.1 Simulation and Experimental Results

In [40], the proposed algorithm, E-CDL, is compared to CDL [38] and MCL [29], for which both simulation and experimental results have been provided. The first enhancement in E-CDL has been to set the average hop distance to $7r/9$; this has resulted in a 47% reduction of the location error on average. The second enhancement, which is computing the ratio of the Euclidean distance to the shortest path length, has resulted in a 69% reduction of the location error on average. The final enhancement is the use of mobile anchor nodes that are allowed to move around the corners. The location error in this case has been reduced by about 29% on average. In comparison to CDL and MCL, the location accuracy of E-CDL is about 50% and 75% higher, respectively. Simulation results show an increase of location accuracy with the increase of sensor node density. Furthermore, if node density is low and there is a high probability that nodes may not be uniformly distributed in the local area, then the average hop distance would be inaccurate. When node density is below 8, the location accuracy of E-CDL is close to that of CDL.

The experimental results are evaluated and implemented on a MICAz Mote Development Kit in a 20x20 m^2 area with 7 motes; four motes as sensor nodes and three motes as anchor nodes. The anchor nodes are deployed in two ways: (i) randomly and (ii) in the corners. In the former case, in comparison with CDL, the location error decreases by about 50%. However, when the anchor nodes are deployed in the corners, the location accuracy of E-CDL as compared to CDL doubles, concluding that placing the anchor nodes in the corners significantly improves the location accuracy.

In [37] , the authors evaluate and compare the performance of EMAP to its earlier version MAP [28] using the NS-2 simulator. A total of 300 mobile sensor nodes have been deployed randomly in a simulation field of size 140x140 m^2 and a sensing field of size 100x100 m^2, where the radio model of the mobile anchors is modeled by the free space model with r = 20 m. At the start of the simulation, the anchors are placed in the corners of the simulation field. Each node selects a random destination and moves to it with a fixed speed. After arrival at the destination, it moves to a new random destination. The mobility model used for all the nodes is based on the Random Waypoint mobility model [44]. The total simulation time for each simulation is 500 seconds.

As beacon intervals increase from 0.1s to 0.5s, the location error of EMAP increases from $0.02r$ to $0.12r$, however, the location error of MAP remains around $1.6r$. It is common practice to normalize the localization error with respect to radio range r as it allows for comparison of localization errors. Moving speed impacts location error in a similar way:

As node speed increases, so does location error. As the percentage of anchor nodes increase from 1% to 5%, the location error of the EMAP algorithm remains at approximately $0.2r$ whereas the location error of the MAP algorithm reduces to approximately $1.0r$.

In [30], the original MCL algorithm is compared to the two proposed algorithms, namely, Dual MCL and Mixture MCL. The simulation field is 250x250 m^2 and the transmission range for the sensor nodes and anchor nodes is 25m. Both sensor nodes and anchor nodes use a Random Waypoint mobility model [44]. Different values of node density and anchor density are used. Note that the definition of node density and anchor density can vary in the literature. In MCL node density (*nd*) and anchor density (*sd*) are defined as follows:

$$nd = \frac{\pi r^2 n_m}{Total\ Area} \tag{2.1}$$

$$sd = \frac{\pi r^2 s_m}{Total\ Area} \tag{2.2}$$

Where n_m is the total number of sensor nodes and s_m is the total number of anchor nodes.

Both Dual MCL and Mixture MCL algorithms provide lower estimation errors than the original MCL algorithm. The Mixture MCL algorithm results in a reduction of the estimation error by about 23% in comparison to the original MCL, whereas the Dual MCL method outperforms the original MCL by about 33%.

Defining a mixing rate parameter φ between 0 and 1, Mixture MCL uses the original MCL sampling method with probability 1-φ, and that of Dual MCL with probability φ. When compared to the original MCL, better performance is obtained with Mixture MCL by varying φ between 0.2 and 0.4. Furthermore, it is shown that the higher the anchor density, the higher the deployment cost. However, when anchor density is greater than

3.5, which corresponds to 111 anchor nodes for the described simulation scenario, no significant improvement of estimation error is observed.

Using the radio model presented in [20], the authors of [30] consider an irregular radio pattern model. In such a model, the anchors can be heard by the nodes within half of the maximum transmit radio range. However, depending on the radio pattern in a certain direction, nodes between the maximum radio range and half of that range may or may not hear from the anchor. The degree of irregularity (DOI) is defined to be the maximum radio range variation per unit degree change in direction. As DOI increases, the estimation error for all three MCL algorithms increases; however, this increase is less for the Dual MCL and Mixture MCL methods as compared to the original MCL. Although the Dual MCL outperforms the other two methods with regards to the estimation error, it is worth noting that more time is required to obtain samples for node estimation, making Mixture MCL a good trade-off between computational time and location accuracy.

2.4.2 Evaluation Criteria

The criteria that are of great interest to us and that will be used for evaluating the existing algorithms are described below.

Location error:

Normalizing the localization error with respect to radio range has become a common practice in order to allow for comparison of various algorithm performances regardless of the radio range of the sensor nodes used. Localization error is evaluated based on the following scale:

Very low	0.01r-0.09r
Low	0.1r-0.5r
Medium	0.6r-0.9r
High	1.0r-1.8r

Computational time:

The amount of time required to execute an algorithm is defined as computational time. Computational time is measured by the complexity of the algorithm.

Anchor density:

Anchor density indicates the percentage of anchor nodes required for better performance results and is evaluated based on the following scale:

Low	<4%
Medium	5-10%
High	10-20%

Extra hardware requirement:

Certain algorithms require extra hardware in order to estimate the location of the sensor nodes. This requirement adds to the cost of the algorithm and can possibly increase the size of the sensor nodes. Algorithms that do not have this requirement are of interest to us.

Node density:

Node density indicates the number of nodes within the transmission range of a node, and is evaluated based on the following scale:

Low	1-3
Medium	4-8
High	9-15

Network size:

Network size is evaluated based on the following scale:

Small	< 100 nodes
Medium	101-350 nodes
Large	> 351 nodes

Mobility of anchors nodes and sensor nodes:

In some algorithms, only the sensor nodes are mobile and in others both sensor nodes and anchor nodes are mobile.

Sensor node/anchor node velocity:

Sensor node/anchor node velocity indicates the speed at which the sensor nodes and/or the anchor nodes move and is defined as *r* distance per time unit. It is evaluated based on the following scale:

Low	0.1-0.9
Medium	1.0-5.0
High	5.1 -10

Memory requirement:

The amount of memory required to store neighbour lists and such determines the memory requirement of the algorithm.

Centralized/distributed:

Algorithms can be either centralized where the algorithm is executed in a central server or distributed, where each node contributes in finding its location.

A summary of the comparison criteria for the algorithms being evaluated is provided in Table 2-1. All the metrics described above are taken into consideration.

Table 2-1: Performance Analysis Comparison Chart

	CDL [38]	**E-CDL** [40]	**MAP** [28]	**EMAP** [37]	**MCL** [29]	**DMCL** [30]	**MMCL** [30]
Location error	L-M	L	H	VL	L	L	L
Computational time	M-H	H	L	M	M	H	M-H
Anchor density	M	H	M	L	M-H	M-H	M-H
Extra hardware	No	No	No	Compass, Odometer	No	No	No
Node density	H	H	H	H	M-H	M-H	M-H
Network Size	M	M	M-L	M-L	M-L	M-L	M-L
Mobility	N	N&A	A	N&A	N&A	N&A	N&A
Node/anchor Velocity	L-M	L-M	L-M	L-M	L-M	L-M	L-M
Memory requirement	H	H	H	H	M	M	M
Centralized or distributed	C	C	D	D	D	D	D

The abbreviations in Table 2-1 are as follows:

VL: Very Low
L: Low
M: Medium
H: High
S: Small
L: Large

C: Centralized
D: Distributed
N: Sensor Nodes
A: Anchor Nodes
DMCL: Dual MCL
MMCL: Mixture MCL

The algorithm of choice for our comparison purposes is one in which the nodes are mobile; the anchors may or may not be mobile. The preferred algorithm has relatively low location error, computational complexity, memory requirement, and anchor density, and performs well in medium to large networks with medium to high node density and node/anchor velocity. Most importantly, the algorithm should not require any extra hardware. Being a centralized or distributed algorithm does not matter per se since our algorithm can be implemented for both cases.

Based on the above criteria, the algorithms chosen for comparison with our proposed algorithm are the MCL and the Dual MCL algorithms [29] [30]. The reasons MCL and Dual MCL have been chosen for comparison with iCCA-MAP is that these algorithms do not require extra hardware, they provide high location accuracy and can be employed for medium to large networks. The Dual MCL algorithm outperforms the original MCL algorithm, but comparisons to the original MCL have also been made given that MCL is one of the well-known methods proposed for mobile nodes in WSNs.

2.5 *Conclusions*

Localization in WSNs is an emerging field with growing importance. Many methods have been proposed for estimating the location of stationary nodes in WSNs. Recently, however, localization methods for mobile nodes in WSNs have become an important research area. Common classifications of localization methods for WSNs can be described as follows:

1) Centralized versus distributed algorithms;
2) Range-based versus range-free methods;
3) Anchor-based versus anchor-free methods;
4) Stationary nodes versus mobile nodes.

In this chapter, the CCA-MAP algorithm, which was proposed for stationary WSNs and is the basis for the iCCA-MAP algorithm was reviewed. Algorithms proposed for mobile WSNs were reviewed with importance given to the more recently published work. Based on the evaluation criteria described in Section 0, MCL and Dual MCL have been chosen as the comparison algorithms for the performance evaluation of iCCA-MAP in terms of location accuracy.

Chapter 3

The iCCA-MAP Algorithm

In this chapter, the proposed iCCA-MAP algorithm is described. Section 3.1 presents an overview of the original CCA-MAP algorithm, outlining its strengths and limitations. Section 3.2 presents the assumptions made and then describes iCCA-MAP and the modifications made to CCA-MAP in order to address mobility.

3.1 *Overview of the CCA-MAP Algorithm*

The original CCA-MAP algorithm is a localization solution that requires only a minimum number of anchor nodes in order to facilitate a rapid deployment process and it can achieve a high level of position accuracy with or without the assistance of range measurements. CCA-MAP uses a self-organizing neural network originally proposed for non-linear data mapping, called Curvilinear Component Analysis (CCA) [42] to build local maps for every node in the network. Each node uses only local information to compute its own local map. The local maps are then patched together to form a global map. This global map has node coordinates that indicate relative node placements, but these coordinates are not tied into any external coordinate system. Using anchor nodes, in a final step, this relative global map can then be translated via the Procrustes method [13][14][25] into an absolute global map, where node coordinates reflect the node positions based on the coordinates used to localize the anchor nodes.

In the CCA-MAP algorithm, neighbours within h hops are included in building the local map for each node. The shortest distance matrix of the local map is computed and used as the approximate distance matrix. In the case of the range-free scheme, the shortest distance matrix is derived using connectivity information among the nodes. When using the range-based scheme, local distance between each pair of neighbouring nodes is measured and used in the shortest distance matrix. Each node then applies the CCA algorithm generating the relative coordinates for every node in its local map. The local maps are then merged. The merged map transforms to an absolute map based on positions of the anchor nodes. For the starting map, the local map of a randomly selected node is used. Then, the neighbour node whose local map shares the most nodes with the current map is selected to merge its local map into the current map. Using the coordinates of their common nodes, two maps are merged. A linear transformation is applied for merging a new local map into the current map. In CCA-MAP, local maps can be merged in parallel in different parts of the network. Anchor nodes are not required during map merging. However, when at least three anchor nodes are found in the patched map, coordinates of the anchor nodes can be used to compute the absolute coordinates of the nodes in that map.

3.1.1 Strengths of CCA-MAP

The major advantages of the CCA-MAP algorithm are its low computational complexity and accurate location estimation. CCA-MAP uses CCA, which has a computational complexity of $O(n^2)$ for n data sets as compared to other Non-Linear data Mapping (NLM) techniques such as Multi-Dimensional Scaling (MDS) which has a computational complexity of $O(n^3)$. When CCA is applied to networks, data sets correspond to the average number of nodes in the local neighbourhood of each node. CCA provides a computational complexity of $O(k^2)$ for each node where k is the average number of nodes in a node's local neighbourhood (as described in Section 2.2). Another advantage of CCA is that it provides improved accuracy and does not require a refinement step, unlike other NLM techniques. Since CCA-MAP uses CCA for building local maps for each node in the network, it benefits from CCA's lower computational complexity and precision, outperforming other well-known localization algorithms such as MDS-MAP used for localizing nodes in a stationary WSN.

3.1.2 Limitations of CCA-MAP

CCA-MAP has been proven to be very accurate for stationary WSNs of size 30 nodes and higher with an average node density of about 10 neighbours per node. However, it has been shown that the computational time required to obtain node coordinates is of order $O(k^2n)$ where k is the average number of neighbours and n is the total number of nodes in the network. When dealing with stationary sensor networks, computational time is not very important since the algorithm is executed once and coordinates remains constant. However, when dealing with mobile nodes, the location of the nodes must be continuously monitored. In this scenario, computational time becomes an important parameter since it can have an impact on the accuracy of the results by providing outdated location estimates if the algorithm takes a long time to provide location updates of the mobile node. Results do not become less accurate due to computational time, but long computational time can result in delayed location estimates so that by the time they are obtained, they less accurately reflect the node's current location. As a result, in order to use CCA-MAP for localizing mobile nodes, changes must be made such that the algorithm becomes fast enough to be able to provide an estimate of the mobile node(s) location in near real-time.

Thus, in WSNs that contain mobile nodes, it is essential to use an algorithm that is efficient and provides results in real-time. By reducing the computational time, the algorithm could be executed more often and thereby improve the accuracy of the mobile node location. The proposed iterative CCA-MAP algorithm, referred to as iCCA-MAP, is a much more efficient method for localizing a mobile node within a WSN. This algorithm takes a fraction of the computational time needed for localizing the mobile node when compared to running the original CCA-MAP algorithm. The iCCA-MAP algorithm is introduced in the next section.

3.2 *The iCCA-MAP Algorithm*

In this section, the iCCA-MAP algorithm is described by first presenting the assumptions made followed by a detailed description of the algorithm.

3.2.1 Assumptions

In order to use the iCCA-MAP algorithm, the following assumptions were made.

- The mobile node's identity is known as a result of the application context. If the application context is such that it does not provide the mobile node's identity, we would at best be able to determine relative mobility by detecting neighbourhood changes through periodic Hello messages, for example. Therefore, the iCCA-MAP algorithm would have to run on multiple nodes (in essence on each node whose local neighbourhood has changed either because the node itself is moving, relative to its neighbours, or because it has a mobile neighbour).

- All nodes have the same transmission range, which is assumed to be a perfect circle.

- All messages are sent and received without error and/or collisions (the physical channel is error free).

- All nodes have the same computational power and memory capacity. A central server can be used for running the iCCA-MAP algorithm. The algorithm can also be executed in a distributed fashion provided that the processors on each node have the computational power to run the algorithm. If the sensor node processor is not powerful enough and limits it from running this algorithm in a distributed fashion, more powerful nodes acting as cluster heads can be used to form clusters and the algorithm can be executed per cluster.

- The Range-free version of CCA-MAP is used for estimating the distance between nodes: hop counts (connectivity information) are used rather than range-based techniques that provide approximated distances between nodes.

- The network is a connected network.

- For CCA-MAP to work well, a minimum average node density of about 10 is required [25]. The same assumption holds for iCCA-MAP.

- Anchor nodes have exact information regarding their location. This is a realistic assumption since anchors nodes could be mounted with a GPS module which obtains the global position of the node. Other techniques are also possible, such as manually placing the anchor nodes and keeping track of their location either by using GPS or an arbitrary user-defined positioning system.

In the next subsection, the iCCA-MAP algorithm is described in more detail.

3.2.2 The iCCA-MAP Algorithm

The iCCA-MAP algorithm computes a single local map for the mobile node rather than computing the local map of every node in the network as is performed by CCA-MAP. At every interval that iCCA-MAP is executed, a new local map for the mobile node is computed and is patched into the relative global map. The proposed algorithm, as shown in Figure 3-1, is similar to the original CCA-MAP algorithm in that it also employs CCA, the self-organizing neural network used for non-linear data mapping, in order to compute the relative coordinates of the nodes in its local map and later uses a map patching step to patch its local map into the relative global map of the network.

The iCCA-MAP algorithm differs from CCA-MAP in that it iteratively computes the local map of the mobile node, estimates the coordinates of the nodes within the mobile node's local map, and patches this new local map into the existing global map to obtain the mobile node's location at every interval it is invoked. By only calculating the local map of the mobile node, the complexity of local map computation in iCCA-MAP decreases from $O(k^2n)$ to $O(k^2)$, where k is the average number of 2-hop neighbours and n is the total number of nodes in the network. This reduction results in much lower computational time for iCCA-MAP as opposed to CCA-MAP for networks with larger number of nodes (assuming constant density).

If the CCA-MAP algorithm was executed in a distributed fashion, the computational complexity on each node would also be $O(k^2)$ as all nodes could compute their local maps in parallel. However, wireless sensor nodes may not be adequately equipped to execute the CCA algorithm, and so most likely, the computation of local maps would be delegated to a cluster head, which would calculate the local maps of every node in its cluster and then patch each local map sequentially.

In the initialization step of the algorithm, the network is assumed to be stationary and CCA-MAP is used for obtaining the location estimate of every node in the network including the initial location of the mobile node. Once all the local maps of the nodes are patched together, the relative global map computed by CCA-MAP is saved. The mobile node is then removed from the saved relative global map.

The remaining steps are the iterative steps and are repeated at desired intervals to provide the location estimate of the mobile node. We can now assume that the mobile node is moving. In the second step, the local map of the mobile node is obtained using its 2-hop neighbours.

In the third step, CCA, the efficient nonlinear neural network data mapping technique, is applied to the local map of the mobile node. The mobile node's local distance matrix, which in the case of range-free measurements is the hop count matrix, is used as the input data set to CCA. The output of CCA is the relative coordinates of all the nodes within the mobile node's local map.

To merge the local map of the mobile node with the global map, the relative global map obtained from running CCA-MAP is saved, the mobile node is removed from the original global map, and then the mobile node's local map is merged with the global map using the common nodes. Merging the global map obtained from the initial run of the CCA-MAP algorithm and the local map of the mobile node, decreases this step's complexity from $O(k^3n)$ to $O(k^3)$, since now only one map is being merged rather than the local map of every node in the network.

In the last step, using at least 3 anchor nodes for 2-dimensional space and 4 for 3-dimensional space, the merged map (global map and mobile node local map) is transformed to an absolute map based on the absolute positions of the anchor nodes. For a anchor nodes, the complexity of this step is $O(a^3+n)$, the same as CCA-MAP. Steps 2 to 5 are executed repeatedly in order to provide the coordinates of the mobile node as it moves.

Initialization Step

1. Use CCA-MAP to estimate the location of all nodes in the WSN

- Save a copy of the global map which contains all nodes' relative coordinates.
- Remove the mobile node from the relative global map.

Iterative Steps

2. Build the local map of the mobile node:

- Neighbours within $h= 2$ hops are used to build the mobile node's local map.
- Compute the shortest distance matrix of the mobile node's local map and use it as the approximate distance matrix.

3. Apply CCA to the local map of the mobile node:

- The mobile node applies the CCA algorithm using the local distance matrix as the input data set and the distance matrix of the local map.
- The relative coordinates for each node in the local map (h hop neighbourhood) of the mobile node is generated.

4. Merge the local map of the mobile with the relative global map:

- A linear transformation (translation, reflection, orthogonal rotation, and scaling), using the Procrustes function, is applied for merging the local map of the mobile node with the relative global map (the mobile node had previously been removed from the global map).
- The mobile node's local map is merged with the relative global map using their common nodes.

5. Use anchor nodes to get the absolute global map:

- Using at least 3 anchor nodes for 2D space and 4 for 3D space, the merged map is transformed to an absolute map based on the absolute positions of the anchor nodes.

Figure 3-1: The iCCA-MAP Algorithm Steps

In a centralized deployment, where we assume a sequential execution of the five steps of the algorithm, the total complexity of iCCA-MAP is as follows:

$$O(k^2) + O(k^3) + O(a^3+n) \approx O(n) \tag{3.1}$$

The overall complexity of iCCA-MAP is $O(n)$, assuming that, as the network size grows, the density (and therefore k) stays constant. Also, we are usually interested in only a fixed number of anchor nodes, which results in a being constant as well.
The total complexity of CCA-MAP is as follows:

$$O(k^2 n) + O(k^3 n) + O(a^3+n) \approx O(n) \tag{3.2}$$

The total overall complexity of iCCA-MAP is asymptotically the same as that of CCA-MAP (again assuming that k and a are constant). But as indicated by the k^3 factor, the constant may be significantly higher. Experimental results that demonstrate this linear performance of both algorithms and the much lower constant factor in the case of iCCA-MAP are discussed in Section 4.3.1.

To localize the mobile node, another approach could have simply been to use the fixed nodes as anchor nodes, but the drawback to this approach is that the neighbouring nodes that are being used as anchor nodes have localization errors. Their localization error will then propagate and cause a larger location error for the mobile node. Another problem with this approach is that there may be times when the mobile node does not have three neighbouring nodes, and therefore a unique solution may not be found. Certainly, the problem of not finding a unique solution could occur during map patching whenever a node does not have at least three neighbouring nodes. However, it is very unlikely for this problem to arise since an average node density of at least 10 is maintained for all networks using CCA-MAP and iCCA-MAP.

Chapter 4

Simulation and Performance Analysis

In this chapter, the performance of the iCCA-MAP algorithm is evaluated. To that end, Section 4.1 describes the experimental setup used for the simulations. A detailed example of an iCCA-MAP simulation is provided in Section 4.2. Then, two sets of simulations are presented in Sections 4.3 and 4.4. In the first set, iCCA-MAP and CCA-MAP are compared in terms of computational time and localization accuracy. In the second set, iCCA-MAP is weighed against competitive algorithms found in the literature, namely MCL and Dual MCL, in terms of localization accuracy and the affects of varying the number of anchor nodes and speed.

4.1 *Experimental Setup*

In this section, the experimental setup in terms of software, hardware, test set, mobility model, comparison algorithms, performance metrics, and simulation steps used for the experiments are described.

4.1.1 Software

For the simulation of iCCA-MAP and CCA-MAP, Matlab R2009b, Version 7.9.0.529 for Windows was used. For the simulation of MCL and Dual MCL, Fedora Eclipse Platform, Version 3.3.2 with Java Version 1.6.0 was used.

The MCL simulation code was provided by its authors, Hu and Evans [29]. However, in order to allow both iCCA-MAP and MCL implementations to read in the same network topologies and mobility scenarios as well as to produce the same error statistics, changes to the original MCL code were made. Using the modified MCL code, Dual MCL was implemented according to the description provided by Navarro *et al.* as outlined in [30].

4.1.2 Hardware

Windows 7 Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz with 12.0 GB memory was used as the computing platform for iCCA-MAP and CCA-MAP, while a Linux workstation with a 2.66 GHz CPU and 3.25GB memory was used as the computing platform for MCL and Dual MCL.

4.1.3 Test Set

All simulations were done on a random square network configuration. They were all performed using range-free measuring techniques. The radio range (r) was set to 8 m for all nodes and anchors. In all cases, the average node density was kept constant by increasing the network area as the number of nodes increased. The average node density for the networks was approximately 12 neighbours per node.

The length of one side of the simulation area and the simulation area was calculated as follows:

$$length = \sqrt{(nodes * 10)} \tag{4.1}$$

$$Area = length^2 \tag{4.2}$$

4.1.4 Mobility Model

To generate a path for the mobile node, the Random Waypoint mobility model [44] was used. The Random Waypoint mobility model selects random speeds and directions and incorporates pause times in between each movement. The pause times allow the mobile node to pause for a specified length of time before travelling to its next destination point. This model has three components: minimum speed, maximum speed, and pause time. The pause time has been chosen to be zero for the simulations, which makes the waypoint mobility model similar to the Random Walk mobility model.

4.1.5 Algorithms Used for Comparison

iCCA-MAP has been compared to CCA-MAP in terms of computational time and location accuracy, and to MCL and Dual MCL in terms of location accuracy and the impact of the number of anchor nodes used.

In the first set of simulations, the localization accuracy and computational time of iCCA-MAP was compared to that of CCA-MAP. In these experiments, a continuous-time simulation model was used. In the continuous-time simulation model, as the respective localization algorithm was being executed, the mobile node kept moving. Therefore, the algorithm that terminates faster is more likely to result in more accurate localization results as well.

In the second set of experiments, iCCA-MAP is compared to MCL and Dual MCL. As iCCA-MAP and MCL/Dual MCL are not implemented using the same programming language, directly comparing execution times is meaningless, so a discrete-time simulation model was adopted, where at every sampling time, a location estimate of the mobile node is obtained and the time elapsed as the algorithm executes is neglected. Thus the sampling times may not correspond to real-time and are assumed to be 1 time unit.

The performance criterion in this case is localization accuracy; however, an important factor also considered is the number of anchor nodes required for an algorithm to perform well. For 2-dimensional space, a minimum of 3 anchor nodes are required, however the use of more anchor nodes can result in better performance for many algorithms.

4.1.6 Performance Metrics

Two performance metrics are used for evaluating iCCA-MAP: average localization error of the mobile node and computational time of the algorithm.

For measuring localization error, the Euclidean distance between the real location of the mobile node and the estimated location has been calculated and normalized by r as follows:

$$Localization\ Error = \frac{\sqrt{(X_{real} - X_{estimated})^2 + (Y_{real} - Y_{estimated})^2}}{r} \tag{4.3}$$

Localization error has been calculated at every sampling interval for both continuous-time and discrete-time simulations. However, in the continuous-time simulations the algorithm does not execute and localize the mobile node at every sampling interval, but rather at specified periodic intervals, referred to as localization intervals. For the discrete-time simulations, the algorithm is executed at every sampling interval, therefore the localization interval and sampling interval for these set of simulations are the same. For all continuous-time and discrete-time simulations the average localization error over the entire simulation time has been reported.

The second performance metric, algorithm computational time, is simply an indication of how long the algorithm takes before the location of the mobile node is determined. The execution time of iCCA-MAP and CCA-MAP were measured by the time elapsed while executing the algorithm using the ‘tic’ and ‘toc’ functions in Matlab.

For the continuous-time simulations, time is important, so a location sample at, for example, time 14 may use a new or an old estimated location value depending on the length of time it takes for an algorithm to execute and provide location estimates. However, for the discrete-time simulations since execution time is not taken into account, the latest localization result after each sampling interval is used (i.e., sampling interval and localization interval for the discrete-time simulations are equal).

The two performance metrics, namely average localization error and algorithm computational time were chosen due to the requirement for cost-effective methods of obtaining accurate location information of mobile node(s) within a WSN. Another important parameter that can play an important role in the accuracy of the location estimates is the number of anchor nodes used. As anchor nodes are usually mounted with a GPS, they may be required to be more powerful nodes in terms of energy consumption, making them more costly as compared to regular sensor nodes.

4.1.7 iCCA-MAP Simulation Steps

The iCCA-MAP simulation steps are described in Figure 4-1. In the initialization phase, a random network is generated and the average node density is calculated. Then the CCA-MAP algorithm is used for obtaining the initial location of every node in the network, including the mobile node. A copy of the relative global map that has been generated by CCA-MAP is saved, from which the mobile node is removed. At desired localization intervals: the local map for the mobile node is obtained and CCA is used for finding the relative coordinates of the mobile node and nodes in its local map. Then the mobile node’s local map is merged with the relative global map and finally, the anchor nodes are used to translate the relative global map to an absolute global map from which the location of the mobile node is obtained. At every sampling interval the mobile node localization error is calculated.

Step 1: Initialization

1.1 Create a random network with the desired number of nodes.

1.2 Check network connectivity and calculate the average node density.

Step 2: Run CCA-MAP for initial estimate of all node coordinates

2.1 Save a copy of the global map that contains all nodes relative coordinates.

2.2 Remove the mobile node from the relative global map.

Step 3: For every localization interval

3.1 Find neighbours within *h* hops of the mobile node for building its local map.

3.2 Compute the shortest local distance matrix for the mobile node

3.3 Apply the CCA algorithm on the mobile node, generating the relative coordinates for the mobile node local map.

3.4 Merge the local map of the mobile node with the original relative global map.

3.5 Using the anchor nodes, transform the merged map to an absolute map.

Figure 4-1: The iCCA-MAP Simulation Steps

4.2 *iCCA-MAP: A Detailed Example*

In this section, a detailed example of the localization process is presented. As shown in Figure 4-2, a randomly generated network of 20 nodes (1 mobile node, 3 anchor nodes and 16 stationary sensor nodes) has been used for this example. The mobile node is designated by the shaded diamond symbol. The lines connect nodes within radio range of one another.

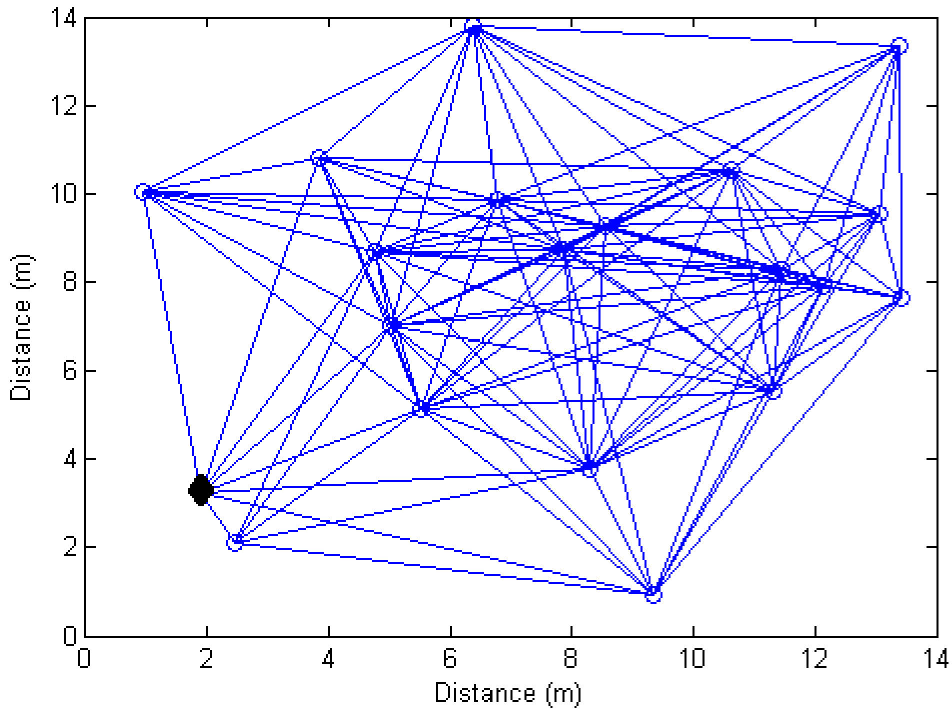

Figure 4-2: Randomly Generated Network of 20 Nodes.

To generate a mobility pattern for the mobile node, the Random Waypoint mobility model was used**.** The path generated for this example is illustrated in Figure 4-3, where the initial location of the mobile node is designated by a shaded diamond symbol. The real location of the mobile node at every sampling interval is illustrated using black circles while the estimated locations of the mobile node computed by iCCA-MAP at every localization interval are shown using Xs. As can be seen, the mobile node traverses the path generated by the Random Waypoint mobility model at different speeds as indicated by the black circles that denote the real location of the mobile node at every sampling interval.

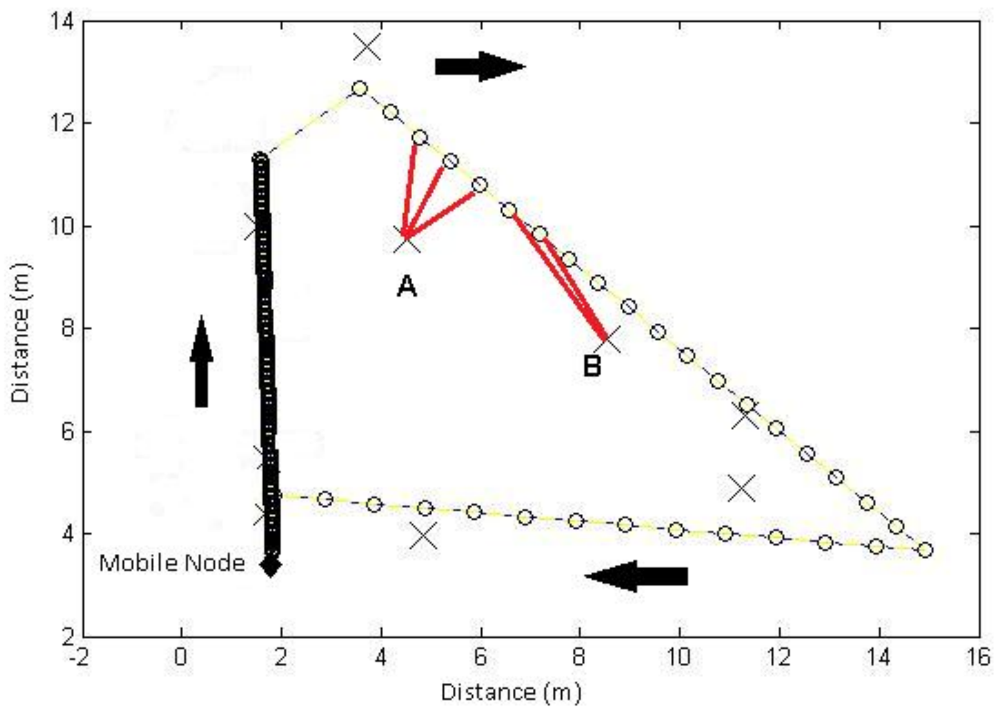

Figure 4-3: Randomly Generated Path Using the Waypoint Mobility Model.

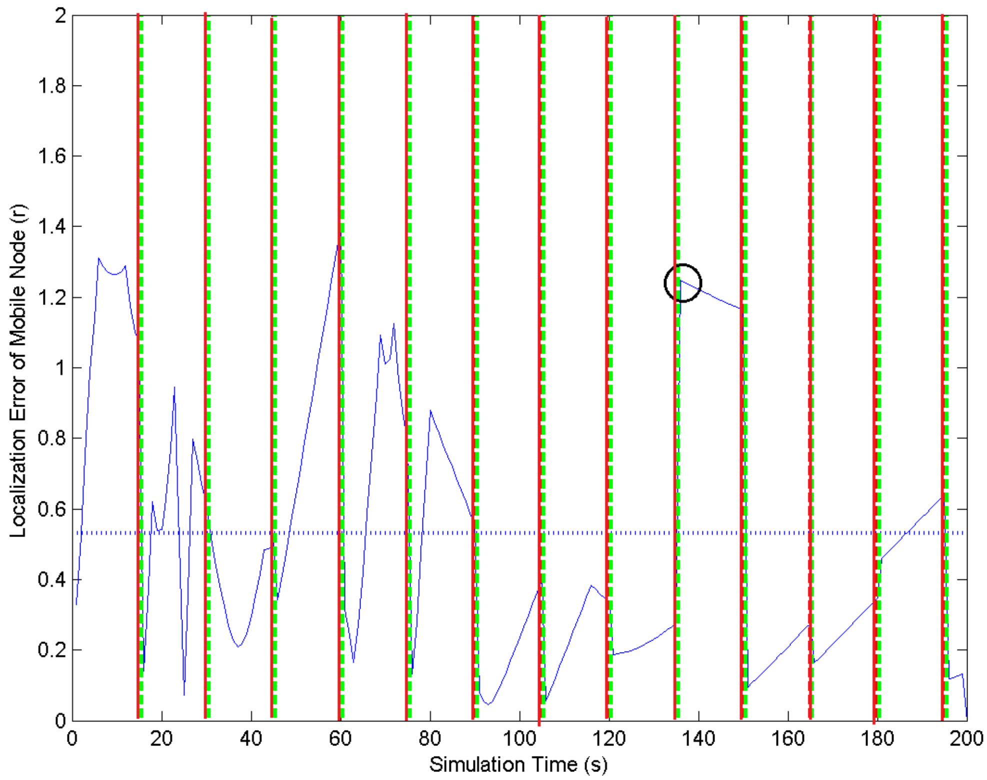

Figure 4-4: Localization Error of iCCA-MAP Every Second for a 200 Second Continuous-Time Simulation.

Simulation results for a continuous-time simulation are shown in Figure 4-4. The dotted horizontal line is the average localization error of the mobile node over the total simulation time of 200 seconds. In this example, iCCA-MAP was executed every 15 seconds (i.e., the localization interval is 15 seconds) as shown by the solid vertical lines. Then, at every sampling interval (i.e., every second), the mobile node's location error was measured with respect to the most recent location information of the mobile node. The dashed vertical lines illustrate the execution time of the algorithm (about 0.5 seconds in this example). In most cases, the localization error of the mobile node decreases after the execution of the algorithm as updated location information is provided. In rare cases however, the localization error increases as depicted by the black circle in Figure 4-4. This increase in localization error occurs when the new node location estimate is farther away from the current location of the node than the previous estimate. An example of this situation has been demonstrated in Figure 4-3 using solid lines that connect the real node locations to their corresponding estimates. As can be seen, the location estimate B is farther away from its corresponding real node location than the previous location estimate A, thus resulting in an increased localization error.

4.3 *Comparing iCCA-MAP with CCA-MAP*

In this section, the results of the continuous-time simulations, comparing computational time and mean localization error for iCCA-MAP and CCA-MAP are presented.

iCCA-MAP is an extension of CCA-MAP, which, as mentioned in Chapter 2, is a high performing localization algorithm proposed for stationary WSNs. CCA-MAP however, cannot provide location estimates for mobile WSNs in real-time as its computational time grows significantly with network size, hence iCCA-MAP has been proposed. In order to evaluate the performance of iCCA-MAP in terms of computational time, comparisons to CCA-MAP have been made to demonstrate the improvements made.

For the comparison of iCCA-MAP and CCA-MAP, networks of sizes ranging from 20 to 50 nodes in increments of 10 were used. For each network size, 20 random networks were generated and the average computational time and localization error were measured. Three randomly selected anchor nodes were used for these sets of simulations as CCA-MAP and iCCA-MAP are designed for fast deployment and therefore require only the minimal number of anchor nodes for good localization results. Four sets of experiments have been conducted. The first set consisted of running CCA-MAP every 15, 30, 45, and 60 seconds for networks of size 20, 30, 40, and 50 nodes respectively. The second set consisted of running iCCA-MAP every 15, 30, 45, and 60 seconds for networks of size 20, 30, 40, and 50 nodes respectively. These localization intervals were chosen based on the execution time of CCA-MAP: they are long enough to allow CCA-MAP to complete its run and provide location estimates of the mobile node before the next localization interval. The next set of experiments consisted of running iCCA-MAP at 20% of the CCA-MAP execution time, namely every 3, 6, 9, and 12 seconds for networks of size 20, 30, 40, and 50 nodes respectively. The last set of experiments consisted of running iCCA-MAP at a constant frequency of every 3 seconds for all network sizes. The time interval of 3 seconds was chosen so that iCCA-MAP could complete the simulation for all network sizes before the next localization interval. A simulation time of 200 seconds was used for all three simulation sets.

For the continuous-time set of simulations, the mobility model parameters were set as follows:

- Minimum speed: 0 m/s;
- Maximum speed: 2 m/s;
- Pause time: 0 seconds.

These parameters were chosen to simulate walking speed.

4.3.1 Computational Time of iCCA-MAP versus CCA-MAP

In the continuous-time simulations, in order to calculate the average localization error, the real location of the mobile node was compared to its latest estimate at every sampling interval, namely one second. For example, for a 20 node network for which iCCA-MAP and CCA-MAP are executed every 15 seconds, at every second the mobile node's real location was compared to the most recent estimate of the mobile node location. Therefore, since it takes iCCA-MAP on average about 0.5 seconds to provide an estimate of the mobile node location in a 20 node network, at time t = 16, this location information can be used. However, it takes CCA-MAP about 11 seconds to provide results, so at t = 16 only estimates from the previous run are available and used for location estimation.

In a first set of experiments, the computational time of the proposed algorithm was compared against the computational time of the original CCA-MAP algorithm. Typically, as the number of nodes increase in a WSN, the computational time needed to localize the nodes also increases. In the case for iCCA-MAP, the increase is much smaller than that of CCA-MAP, as demonstrated in Figure 4-5 and Figure 4-6.

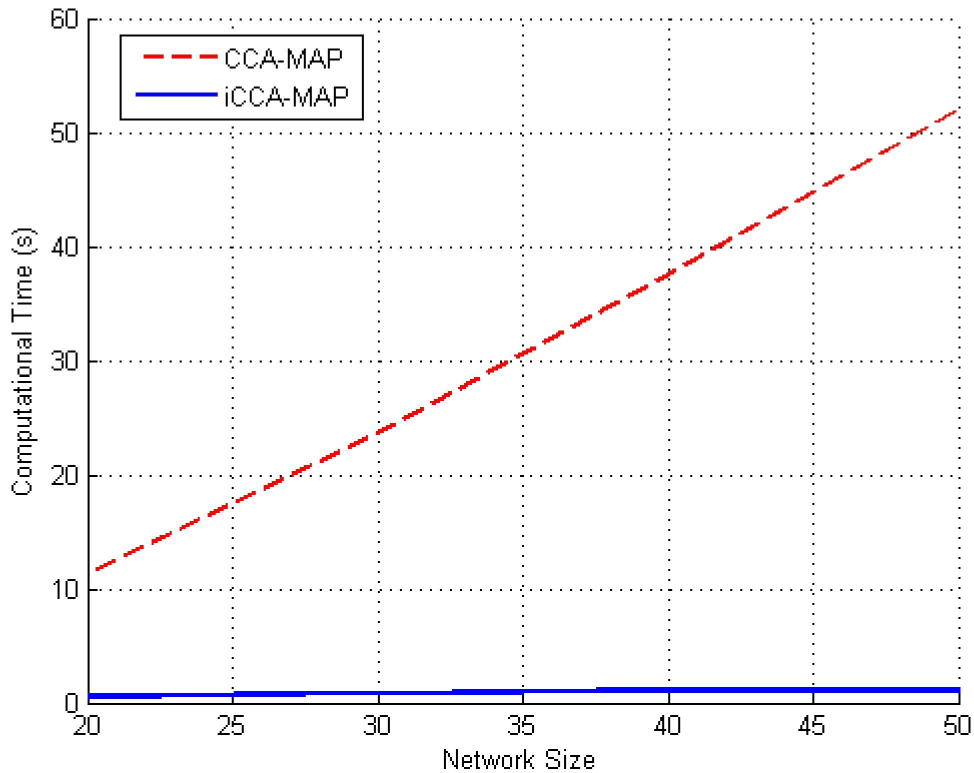

Figure 4-5: Computational Time of iCCA-MAP versus CCA-MAP for Networks of Size 20, 30, 40, and 50 Nodes.

For the sake of completeness, computational times for larger network sizes (namely those used for the comparison of iCCA-MAP versus MCL and Dual MCL presented in Section 4.4) are shown in Figure 4-6.

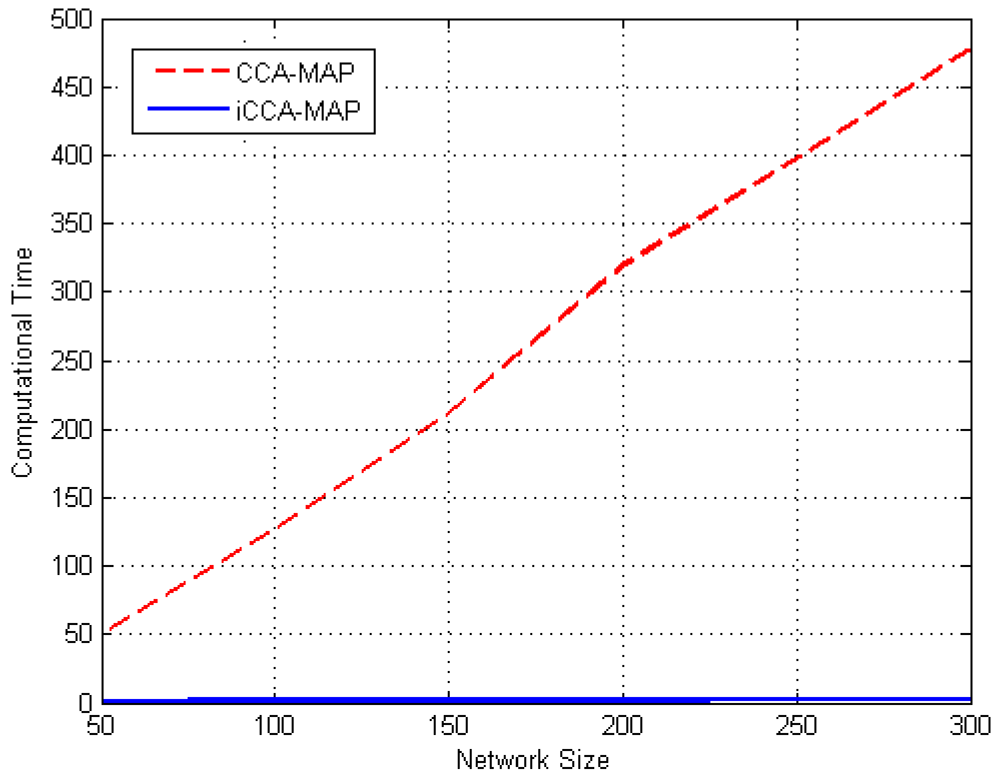

Figure 4-6: Computational Time of iCCA-MAP versus CCA-MAP for Networks of Size 50, 100, 150, 200, 250, and 300.

As shown in Figure 4-5 and Figure 4-6, the computational time of CCA-MAP increases linearly with respect to the network size. As the number of nodes increases, it becomes apparent that the original CCA-MAP algorithm will not be able to provide sufficiently accurate results, as it takes significantly more time to localize the mobile node, thus providing out-dated location estimates. On the other hand, iCCA-MAP demonstrates a very small increase in execution time with an increase of the network size, thus providing up-to-date and (at least potentially) accurate results. This allows iCCA-MAP to be applied at shorter time intervals. Consequently, near real-time information regarding the mobile node's location can be obtained and therefore, a lower average distance error may be achieved by allowing for closer localization intervals. Also, note that if it is assumed that node density is kept constant and the number of anchor nodes is fixed, these experimental results are consistent with the complexity notations discussed in Section 3.2.2.

4.3.2 Mean Localization Error of iCCA-MAP and CCA-MAP

In order to demonstrate the effectiveness of low computational time and the importance of obtaining results as close to real-time as possible, the mean localization errors of iCCA-MAP and CCA-MAP have been compared.

Figure 4-7 shows the mean localization error of the mobile node for network sizes of 20, 30, 40, and 50 nodes. For each network size, 20 random networks were generated and the average localization error of the mobile node was calculated. The error bars depicted in the figures illustrate the 95% confidence interval. It is apparent from Figure 4-7 that as the network size increases, the mean localization error for both algorithms increases. This behaviour can be explained by the fact that as the problem size increases, the CCA-MAP algorithm takes much longer to compute the new location of the mobile node. Since both algorithms are executed the same number of times and at the same localization interval, it means that, as the problem sizes increases, the algorithms are executed less often. This will inevitably increase the average localization error since the location of the mobile node will be updated less frequently and result in out-of-date location information. When executed the same number of times, iCCA-MAP provides better results as compared to CCA-MAP. This is because since iCCA-MAP takes less time to execute, the location information at the beginning of each interval is more up-to-date and more accurate, thus resulting in a lower average localization error.

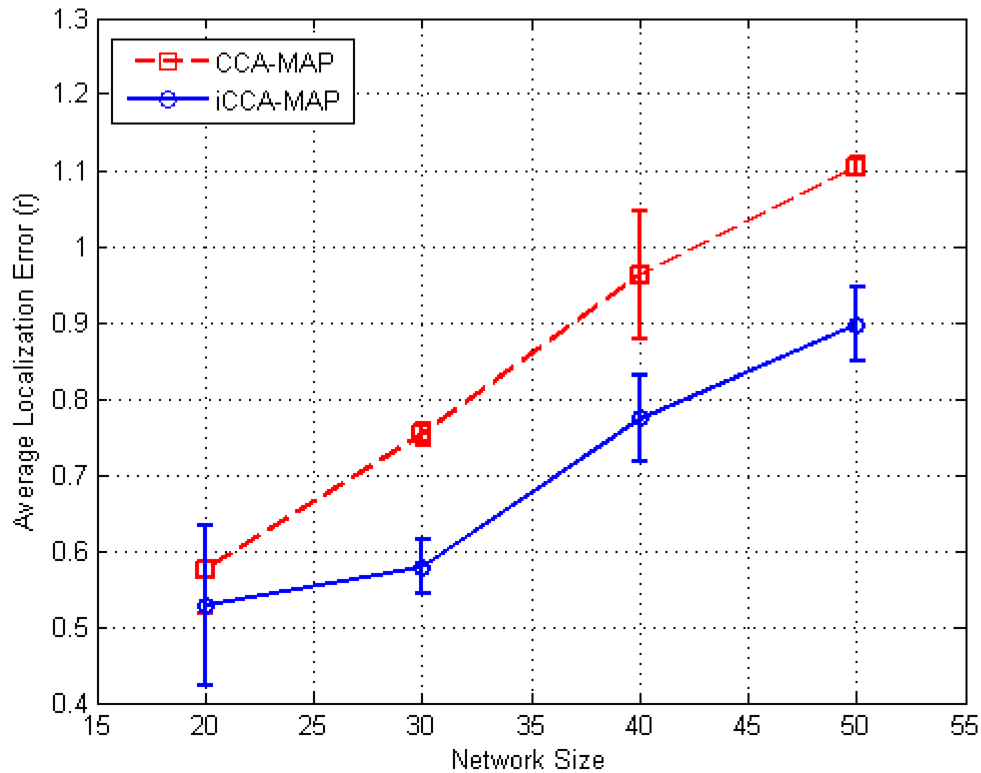

Figure 4-7: Average Localization Error of iCCA-MAP versus CCA-MAP for Localization Intervals of 15, 30, 45, and 60 Seconds and Network Size of 20, 30, 40, and 50.

As mentioned previously, the advantage of iCCA-MAP over the original CCA-MAP algorithm is its ability to provide near real-time results. Knowing this, iCCA-MAP can be applied at closer time intervals, which will provide updated location information regarding the mobile node and thus result in lower localization errors.

The purpose of the next set of simulations conducted was to run iCCA-MAP at higher frequencies (i.e., closer localization intervals) to see if better results could be maintained as hypothesised. First, iCCA-MAP was executed at 20% of the time required for the execution of CCA-MAP, namely every 3, 6, 9, and 12 seconds for network sizes of 20, 30, 40, and 50 nodes respectively. iCCA-MAP was also executed at a constant frequency of three seconds for all network sizes. The results of executing iCCA-MAP at higher frequencies have been depicted in Figure 4-8.

By comparing Figure 4-7 and Figure 4-8, it is clear that iCCA-MAP provides better results than CCA-MAP as it takes much less time to complete its execution. The error bars depicted in Figure 4-7 and Figure 4-8 illustrate 95% confidence intervals and demonstrate that in most cases the improved performance is statistically significant. In the case of the 20 node network, the wider range of average localization errors and the overlap of the error bars can be explained by the fact that in the 20 random networks generated, one network resulted in high localization error due to the randomness of anchor placements and thus caused an increase the average localization error range.

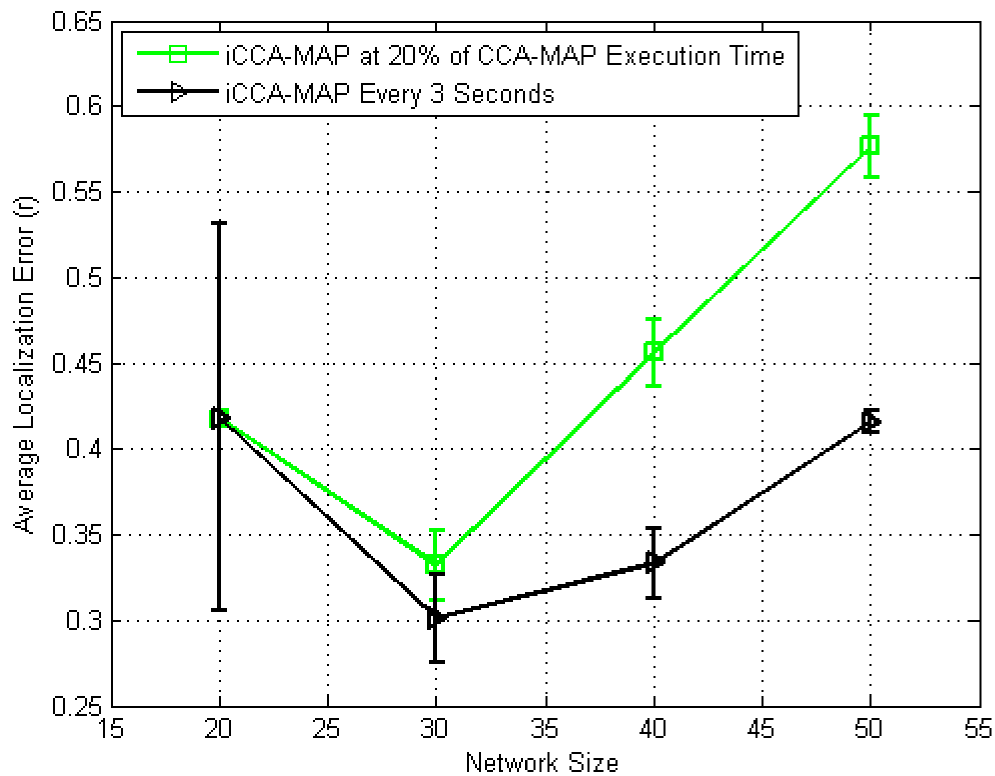

Figure 4-8: Average Localization Error for iCCA-MAP When Executed at 20% of CCA-MAP Execution Time and at Every 3 Seconds in All Cases.

Figure 4-9 shows an overview of all the results comparing iCCA-MAP and CCA-MAP in terms of localization error. As can be seen, the best results are obtained when iCCA-MAP is executed at higher frequencies, namely every three seconds for all network sizes. These results are as expected since when localization is performed at shorter time intervals, the location estimates provided are more recent and up-to-date.

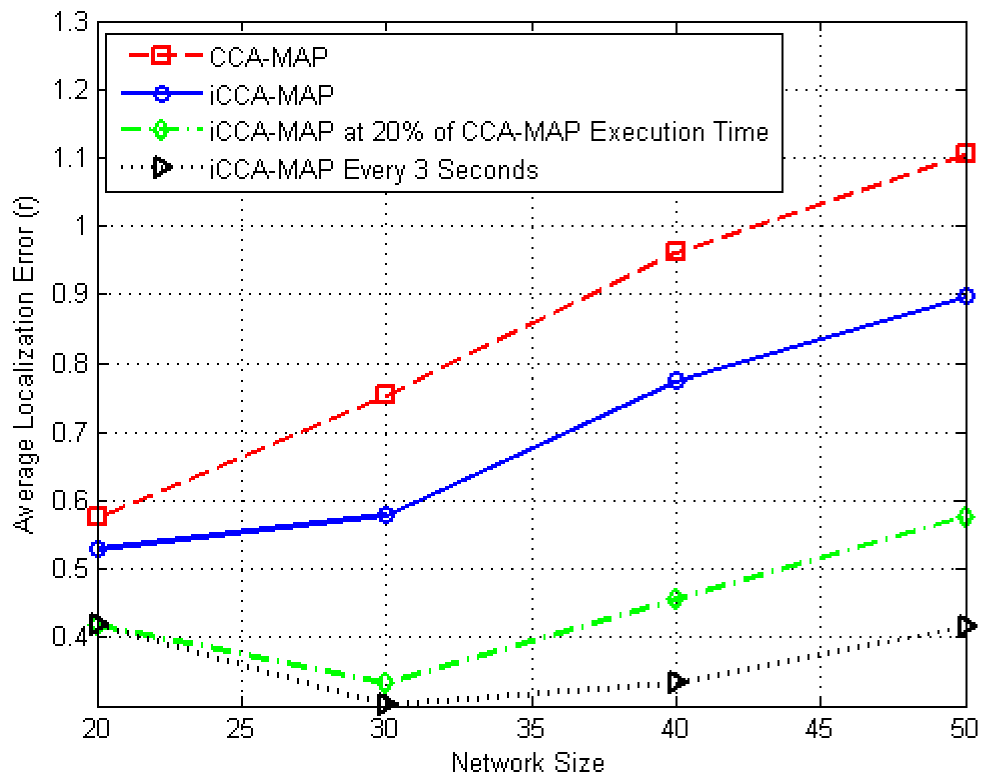

Figure 4-9: Continuous-Time Simulations Comparing iCCA-MAP and CCA-MAP Executed at the Same Frequency and iCCA-MAP Executed at Higher Frequencies.

Simulation results show that iCCA-MAP can provide better results than CCA-MAP for WSNs that contains a mobile node since its execution time is a fraction of that of CCA-MAP, allowing it to be used in larger networks. In the next section, the performance of iCCA-MAP is compared with two state-of-the-art mobile node localization algorithms proposed by Hu and Evans [29] and Stevens Navarro *et al.* [30].

4.4 *Comparing iCCA-MAP to MCL and Dual MCL*

In this section, the performance of iCCA-MAP is evaluated with respect to localization accuracy. In order to demonstrate iCCA-MAP's performance with regard to localization accuracy, comparisons to two high performing algorithms, namely MCL and Dual MCL have been made. The reasons MCL and Dual MCL have been chosen as the comparison algorithms are that these algorithms do not require additional hardware, similar to iCCA-MAP, provide rather accurate localization estimates, and can be used in scenarios where both mobile and stationary nodes are deployed. They are also range-free algorithms and thus can fairly be compared to the range-free version of iCCA-MAP.

MCL and Dual MCL can be used for scenarios in which nodes are mobile and anchor nodes are static, nodes are static and anchor nodes are mobile, and both nodes and anchor nodes are mobile. Comparisons are made using the first scenario in which a node is mobile and the anchor nodes are static, since that is the scenario iCCA-MAP has been designed for.

Additionally, MCL is among the well-known mobile localization algorithms from which other algorithms have been derived and compared against, making it a popular reference point. The second algorithm chosen for comparison is Dual MCL. Dual MCL was chosen because it provides better estimation of node locations as it is an improvement to the original MCL algorithm as explained in Chapter 2, at the cost of increased computational effort.

When comparing iCCA-MAP to MCL and Dual MCL, network sizes of 50, 100, 150, 200, 250, and 300 nodes were used, and for each network size, 10 random networks were generated. Two sets of simulations were performed where the number of anchor nodes was varied and the localization error measured. In the first set, the number of anchor nodes is the minimum required number, namely 3 and in the second set 20% of the total nodes have been designated as anchor nodes. A total simulation time of 100 seconds was used for all simulation sets. For these sets of simulations, the mobility parameters were set as follows:

- Minimum speed: 0 m/s;
- Maximum speed: 2 m/s, 8 m/s;
- Pause time: 0 seconds.

These two speeds were chosen to simulate walking and running speeds. It was also of interest to see the effect speed has on the performance of the algorithms. When normalized by radius *r* which is set to 8 for all simulations, the corresponding speeds become 0.25 *r/unit time* and 1 *r/unit time* respectively. Results pertaining to 1 *r/unit* time are presented in Sections 4.4.1 and 4.4.2 and results pertaining to 0.25 *r/unit time* are presented in Section 4.4.3 in which the effects of node speed on iCCA-MAP, MCL, and Dual MCL are discussed.

4.4.1 Average Localization Error of iCCA-MAP, MCL, and Dual MCL

As mentioned above, the average localization error for networks with 50, 100, 150, 200, 250, and 300 nodes has been calculated for iCCA-MAP, MCL, and Dual MCL. Three anchor nodes were used in all three algorithms. The maximum speed of the mobile node for all algorithms was set to 8 m/s corresponding to 1 *r/unit* time when normalized by radius *r*. Simulation results, as depicted in Figure 4-10, show that the average localization error of iCCA-MAP is lower than that of MCL and Dual MCL.

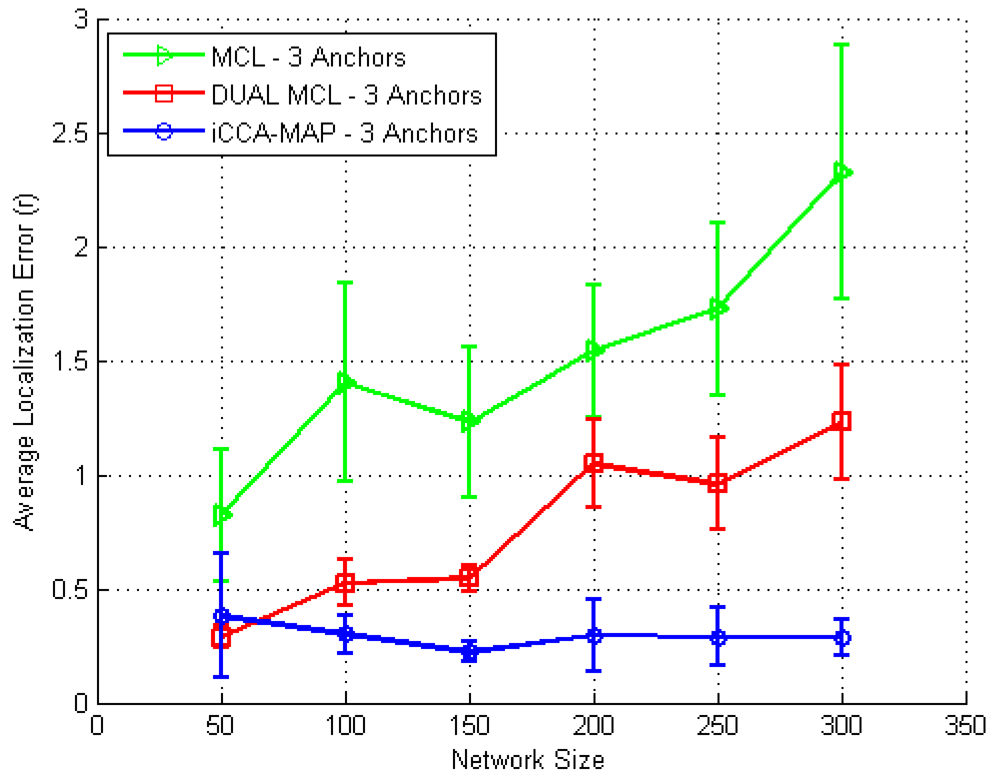

Figure 4-10: Average Localization Error of iCCA-MAP versus MCL and Dual MCL for Network Size of 50, 100, 150, 200, 250, and 300 Nodes with 3 Anchor Nodes and Speed of 8m/s.

The error bars depicted in the figure illustrate the 95% confidence interval. As the confidence intervals of iCCA-MAP, MCL and Dual MCL do not overlap, it can be concluded that the differences in their localization performance are statistically significant. The only exception is the 50 node network where all 3 algorithms provide similar results and there is an overlap of their confidence intervals. For iCCA-MAP however, the confidence intervals for different network sizes overlap, signifying that the average localization error does not change very much with network size. The performances of MCL and Dual MCL, on the other hand, deteriorate as the network size increases. Dual MCL outperforms MCL as expected and corresponds to results reported in [30].

The biggest advantage that iCCA-MAP has over MCL and Dual MCL is the ability to provide accurate estimates of node location with the minimum number of anchors required, namely three for the 2-dimensional space. Increasing the number of anchors results in a higher cost and energy consumption for nodes since anchor nodes usually require a GPS receiver to be mounted on them. While CCA-MAP has been shown to not benefit significantly from additional anchors, MCL and Dual MCL are both very sensitive to the number of anchors used. In the next section, the impact of adding more anchor nodes to the network is explored.

4.4.2 Varying the Number of Anchors in iCCA-MAP, MCL, and Dual MCL

In this section, the performance of iCCA-MAP, MCL, and Dual MCL are evaluated with respect to varying the number of anchor nodes. Again, randomly deployed networks of 50, 100, 150, 200, 250, and 300 nodes were used for evaluating the effectiveness of increasing the percentage of anchors in the networks at a maximum mobile node speed of 8 m/s.

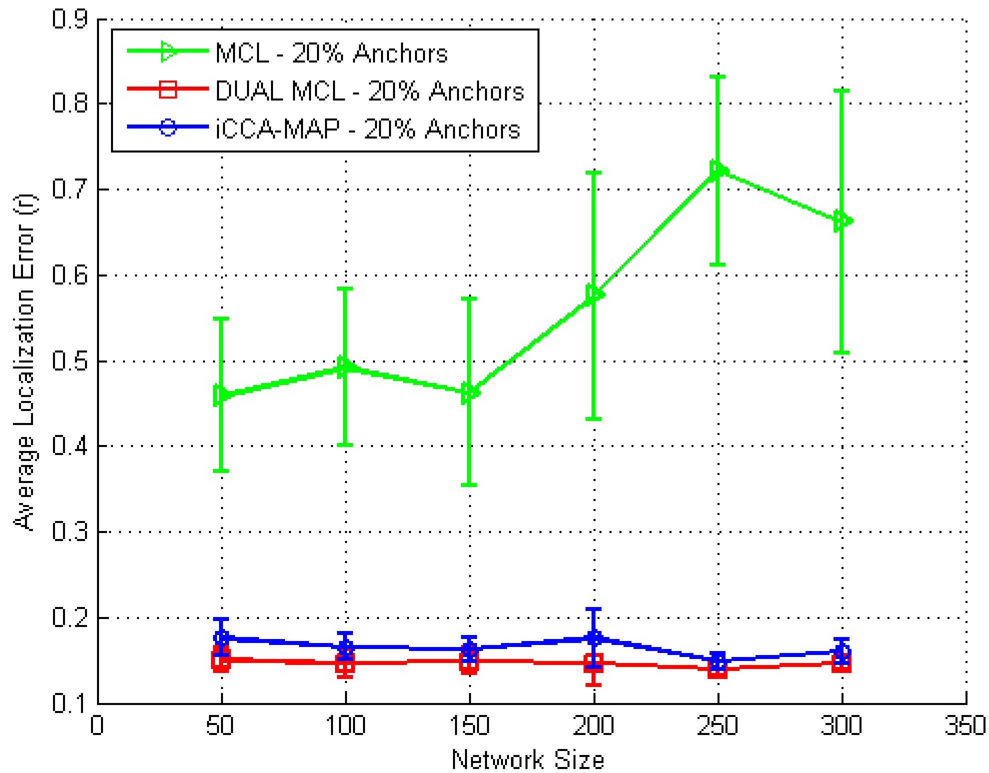

Figure 4-11: Average Localization Errors of iCCA-MAP versus MCL and Dual MCL for 20% Anchor Nodes in the Network and Speed of 8m/s.

Typically, as the number of anchor nodes increases in a WSN, the accuracy of the localization estimates also increases, as there are more nodes that have exact information about their location, and thus can provide precise location information to the regular nodes. In iCCA-MAP, the transformation (rotation, scaling, and translation) of the global map in order to obtain its absolute position, will be computed using more nodes. For MCL and Dual MCL, the increase in anchor nodes translates into a higher number of observations for non-anchor (or regular) nodes, thus improving their location estimation.

When comparing Figure 4-10 and Figure 4-11, we can see that an increase in the percentage of anchor nodes improves the performance of all three localization algorithms: iCCA-MAP, MCL, and Dual MCL. iCCA-MAP's performance is less sensitive to the number of anchor nodes, and the localization accuracy it provides with the minimum number of anchor nodes, namely three, still outperforms MCL's performance even with a higher percentage of its nodes as anchor nodes. However, Dual MCL provides slightly better results when the number of anchors nodes in the network has been increased to 20%. Figure 4-12 has been included in order to demonstrate the overall improvements for all three algorithms.

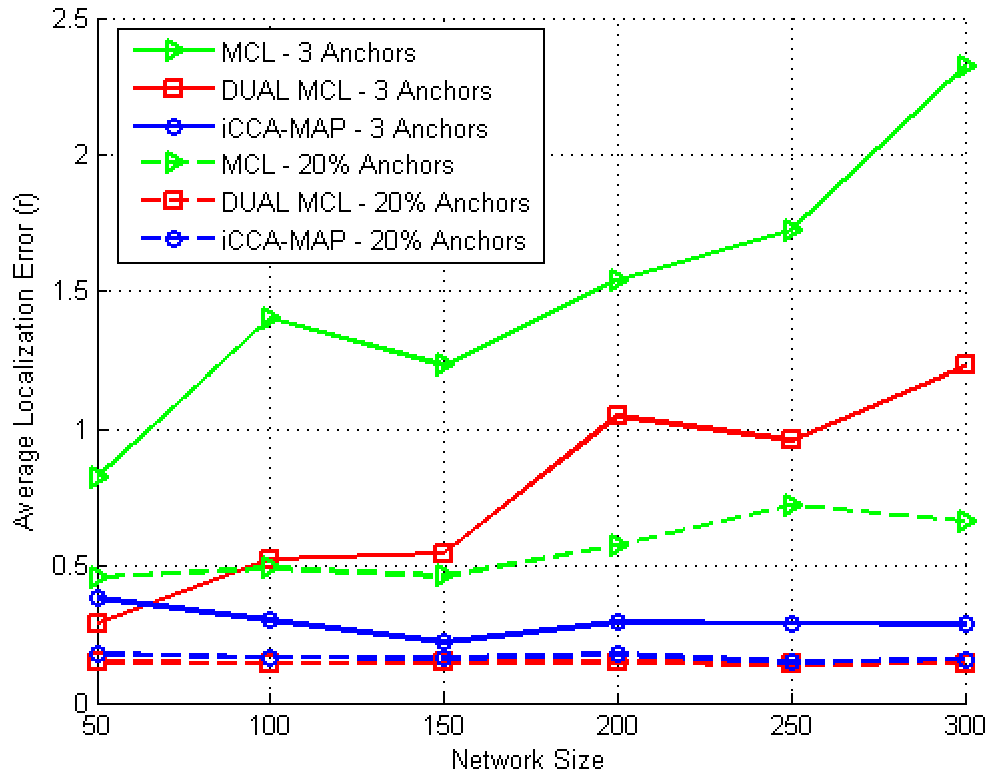

Figure 4-12: Average Localization Error for iCCA-MAP versus MCL and Dual MCL for Three Anchor Nodes and 20% Anchor Nodes and Speed of 8m/s.

Although both MCL and Dual MCL show significant improvements in localization estimates with the increase in the number of anchor nodes, they do not outperform iCCA-MAP when only three anchor nodes are present in the network. As reported in [30], due to the increase in time required to obtain results for Dual MCL, the Mixture MCL algorithm has been proposed, which has a lower computational time as well as lower location accuracy. Therefore, it could be concluded that for the Dual MCL algorithm, better accuracy is obtained by sacrificing computational time which is unfavorable for mobile nodes in WSN. In Section 4.3, it was shown that obtaining accurate location estimates in near real-time is of high importance in the continuous-time simulations as they resemble real world conditions. Using the obtained results in this chapter and the results provided by [30] regarding the computational time of Dual MCL, it can be concluded that iCCA-MAP is a more accurate, efficient, and cost-effective algorithm than MCL and Dual MCL.

4.4.3 Effect of Velocity on iCCA-MAP, MCL, and Dual MCL

In this section, simulation results for mobile node speed of 2 m/s equivalent to 0.25 *r/unit time* are presented. The results are compared to results pertaining to node speed of 8 m/s equivalent to 1 *r/unit time* and the effect of varying the speed of the mobile node is analyzed and discussed.

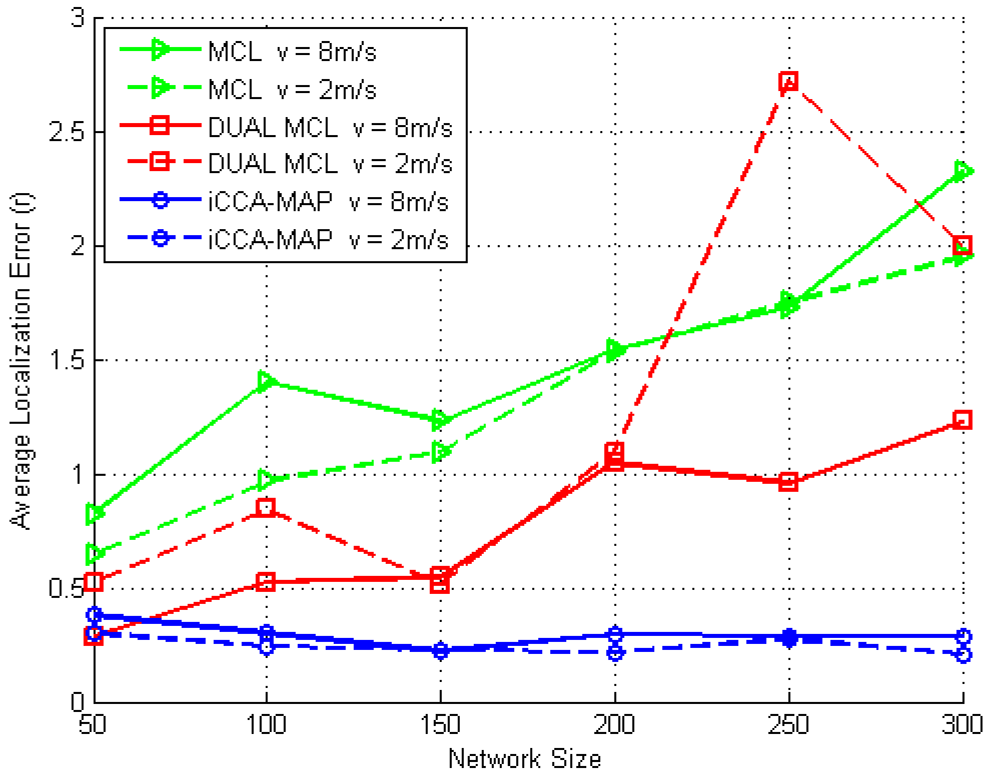

Figure 4-13: Average Localization Error of iCCA-MAP versus MCL and Dual MCL for Network Size of 50 - 300 Nodes with 3 Anchor Nodes and Speeds of 2m/s and 8m/s.

Figure 4-13 depicts the average localization error of iCCA-MAP, MCL and Dual MCL for network sizes of 50, 100, 150, 200, 250, and 300 with 3 anchor nodes for both speeds of 2 m/s and 8 m/s. Figures 4-14, 4-15, and 4-16 depict the average localization errors of iCCA-MAP, MCL, and Dual MCL for speeds of 2 m/s and 8 m/s when only 3 anchor nodes are used. All error bars illustrate 95% confidence intervals. As seen in Figures 4-14 and 4-15, for iCCA-MAP and MCL there is a slight increase in localization error as the speed increases from 2 m/s to 8 m/s for most network sizes. However, in most cases the error bars either completely overlap or there is quite a bit of overlapping, indicating that the improved performance may not be statistically significant. In the case of Dual MCL, as seen in Figure 4-16, for most network sizes the average localization error decreases with the increase of speed. This occurs because at higher speeds, the Dual MCL filtering step, which filters samples based on the mobile node's previous location and its maximum velocity per unit time, allows for the possibility of collecting more samples that meet the condition and thus can provide a better estimate of the node's location. At low speeds, since the area in which the node can reside based on the node's previous location and its maximum velocity is very small, the chances of finding enough sample points is rather low. In that case, the node's previous location is used as an estimate of its current position and if the node's previous location is not available (i.e., node has not been initialized yet), a location near the center of the simulation area is assigned to the node. The fact that only 3 anchor nodes are present in the network also contributes to the problem of finding enough sample points, since the number of observations will be low when the number of anchor nodes is low. This can be seen in Figures 4-13 and 4-16, with the significant increase in the average localization error of Dual MCL for network sizes of 250 nodes. This increase can be due to the randomness of the network topologies as well as the randomness of the anchor node placements. When the anchor nodes are placed in such a way that neither direct nor indirect observations can be obtained, the samples generated cannot be validated and if there is no information on the previous location of the node, a random location near the center of the simulation area is estimated for the node. If this occurs often enough, as it has for the 250 node networks, the average localization error increases significantly since many of the location estimates are arbitrary chosen.

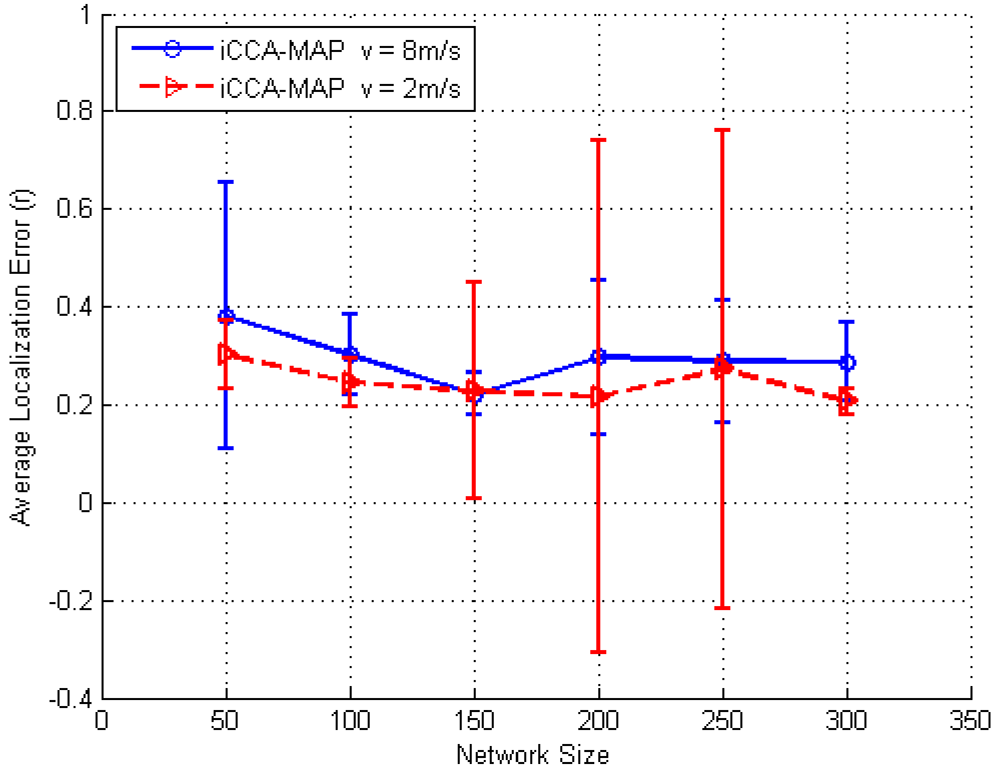

Figure 4-14: Average Localization Error of iCCA-MAP for Node Speeds of 2m/s and 8m/s in Network Size of 50-300 Nodes with 3 Anchor Nodes.

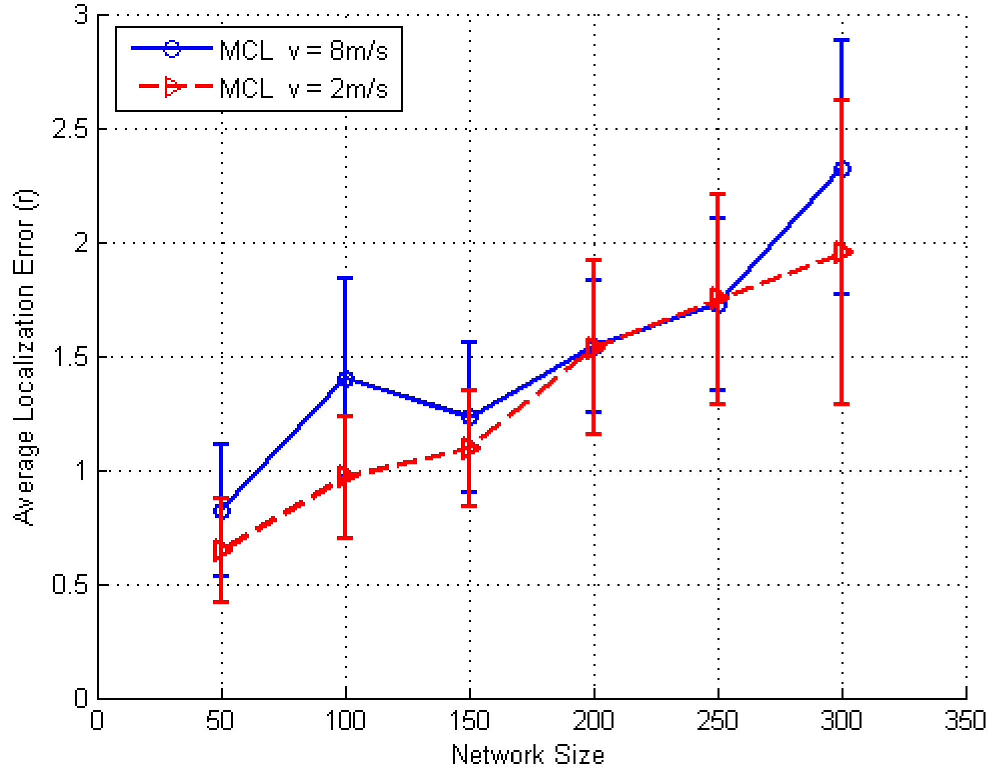

Figure 4-15: Average Localization Error of MCL for Node Speeds of 2m/s and 8m/s in Network Size of 50-300 Nodes with 3 Anchor Nodes.

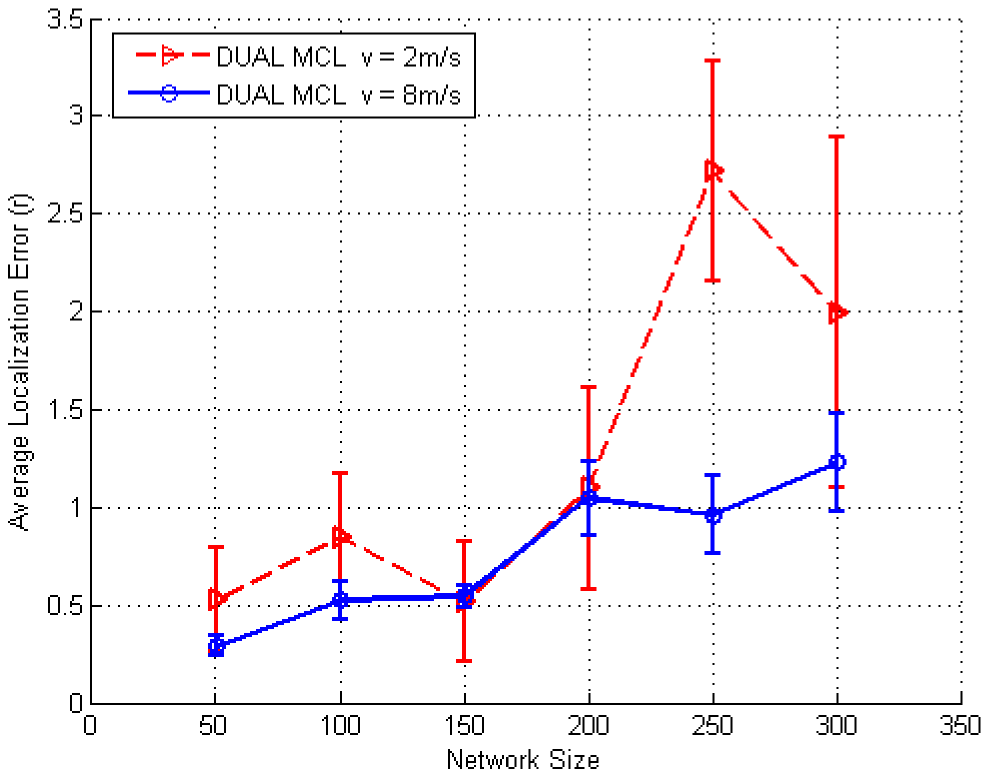

Figure 4-16: Average Localization Error of Dual MCL for Node Speeds of 2m/s and 8m/s in Network Size of 50-300 Nodes with 3 Anchor Nodes.

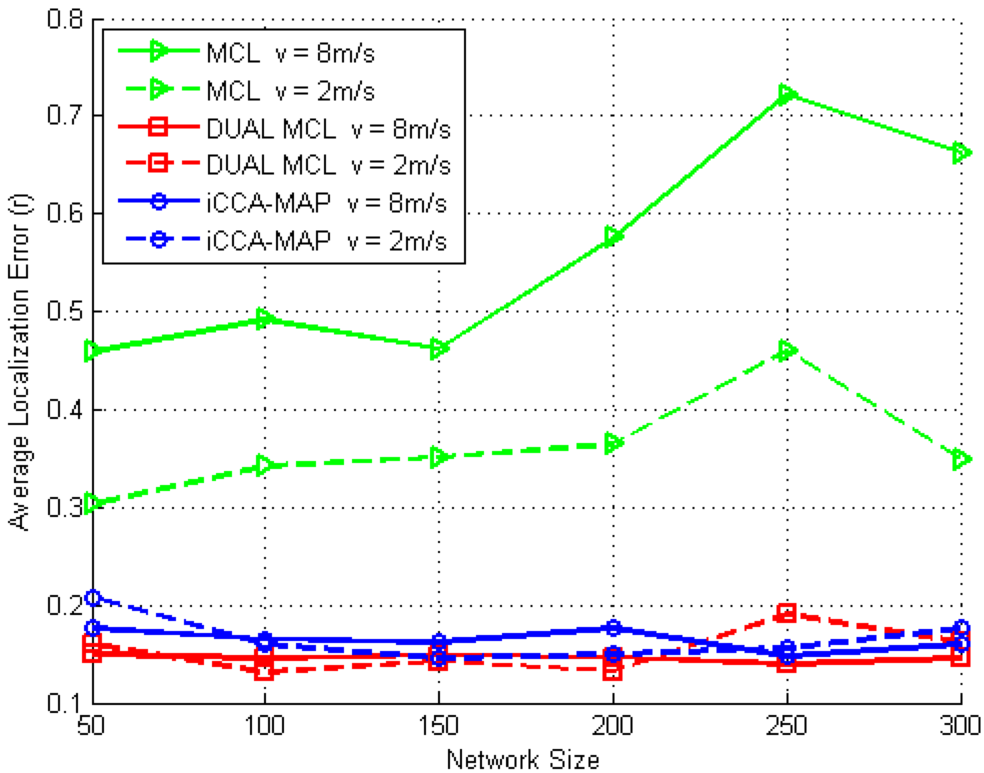

Figure 4-17: Average Localization Error of iCCA-MAP versus MCL and Dual MCL for Network Size of 50-300 Nodes with 20% Anchor Nodes and Speeds of 2m/s and 8m/s.

Figure 4-17 depicts the average localization error of iCCA-MAP, MCL and Dual MCL for speeds of 2 m/s and 8m/s when 20% of the nodes in the network are anchor nodes. As depicted in Figures 4-17 and 4-18, for most network sizes, MCL's performance deteriorates as the speed of the mobile node increases. This can be explained by the fact that samples predicted in MCL are based on the previous predicted location of the node and the node's maximum speed per unit time. As the mobile node's speed increases, the area in which the predictions can be obtained also increases resulting in less accurate estimations. In the case of iCCA-MAP and Dual MCL, shown in Figures 4-19 and 4-20, no definite conclusions can be made with respect to varying speeds as there exists no obvious trend and there is quite a bit of overlap of the error bars illustrating low statistical significance.

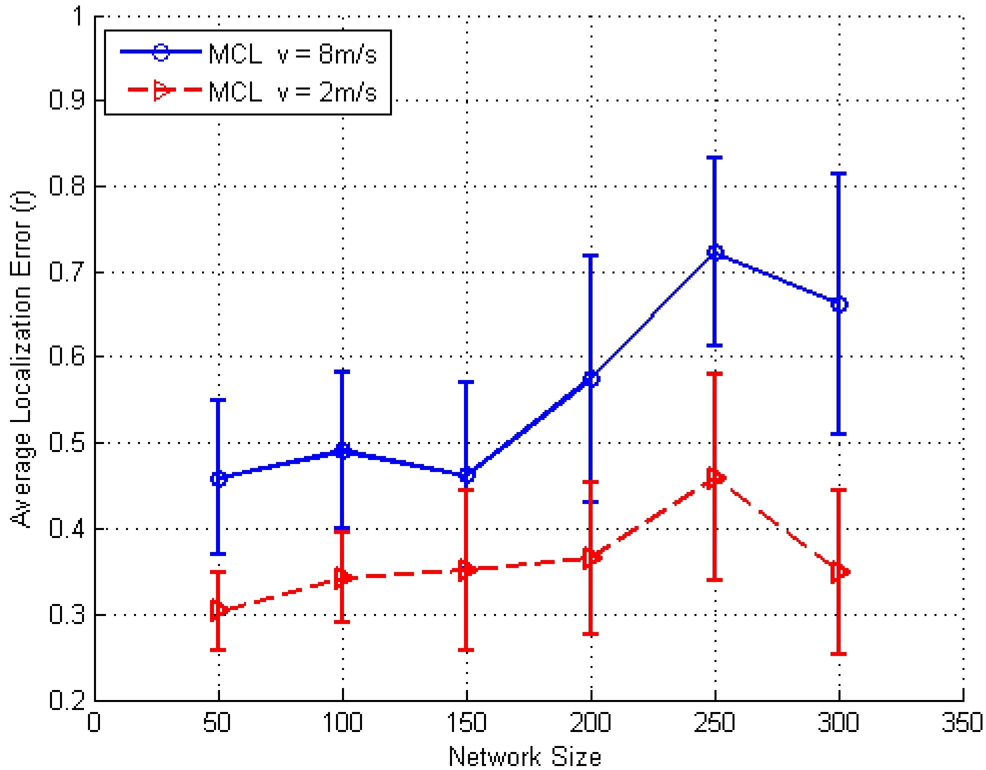

Figure 4-18: Average Localization Error of MCL for Node Speeds of 2m/s and 8m/s in Network Size of 50-300 Nodes with 20% Anchor Nodes.

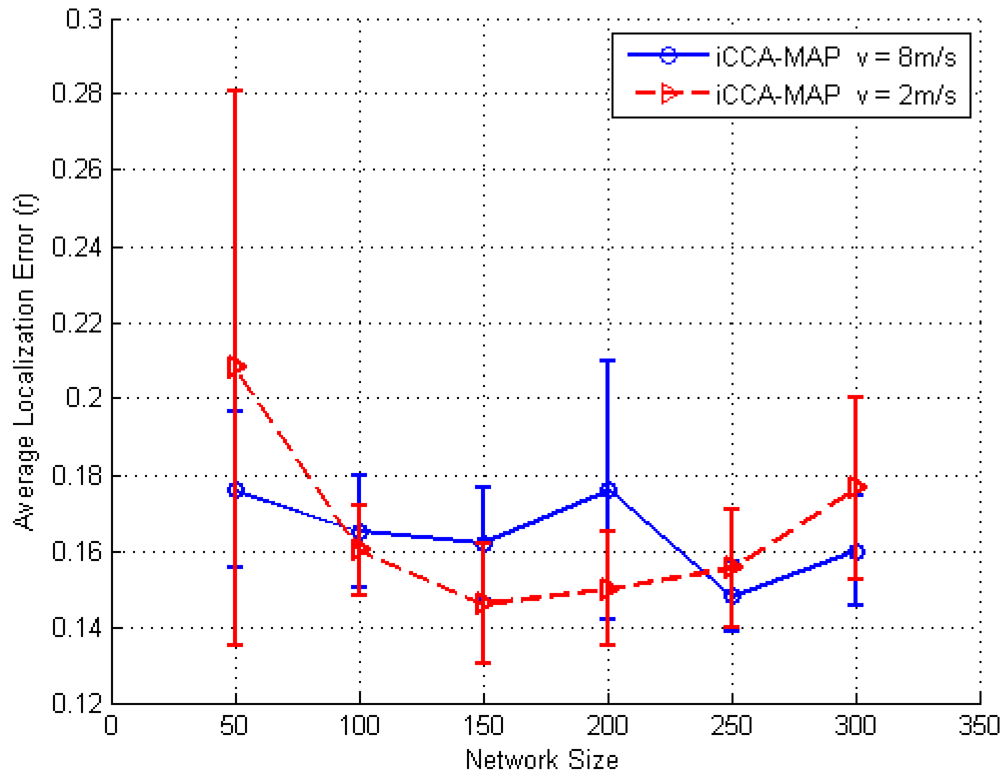

Figure 4-19: Average Localization Error of iCCA-MAP for Node Speeds of 2m/s and 8m/s in Network Size of 50-300 Nodes with 20% Anchor Nodes.

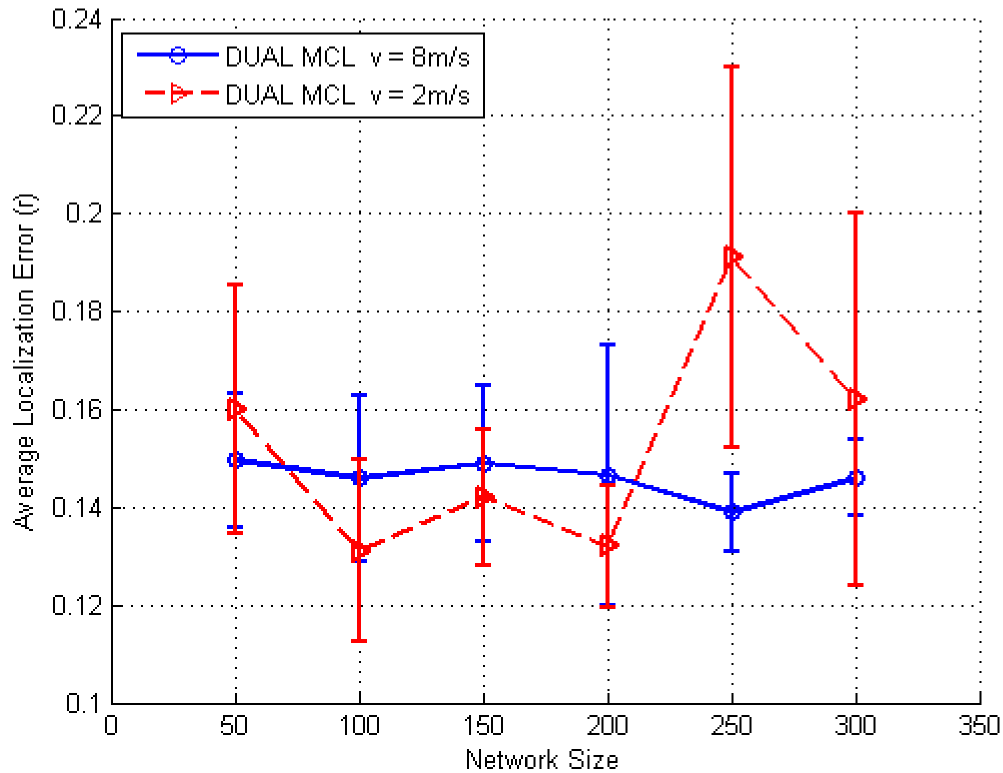

Figure 4-20: Average Localization Error of Dual MCL for Node Speeds of 2m/s and 8m/s in Network Size of 50-300 Nodes with 20% Anchor Nodes.

Based on the presented results, it can be concluded that MCL and Dual MCL are impacted by mobility depending on anchor node density. For low anchor node density, Dual MCL is adversely affected by low speed, whereas for a high anchor node density MCL suffers from higher mobility. iCCA-MAP, on the other hand, performs consistently well independent of speed and the number of anchor nodes.

Chapter 5

Conclusions and Future Work

In this last chapter, an overview of the contributions of this thesis is presented. Then, some limitations of the proposed algorithm are discussed and finally, recommendations for future work are made.

5.1 *Overview of the Contributions*

The main objective of this research was to develop a localization algorithm in order to locate mobile nodes in a wireless sensor network. More precisely this involved:

1. Modifying an existing algorithm called CCA-MAP such that it localizes mobile sensor nodes.

2. Comparing the results of the proposed algorithm with the results of CCA-MAP so as to evaluate the effectiveness of the proposed algorithm.

3. Comparing the results of the proposed algorithm with current state-of-the-art algorithm, specifically MCL and Dual MCL.

In order to accomplish these three objectives, the following contributions have been made as a result of this research:

1. A new mobile node localization algorithm for WSNs, namely iCCA-MAP, has been proposed.

2. The computational time of iCCA-MAP has been compared to that of CCA-MAP, the original algorithm for stationary WSNs from which iCCA-MAP was developed. Results show that iCCA-MAP is by far a much more efficient algorithm that can be used for providing near real-time location estimation for mobile node(s) in WSNs. Compared to the original CCA-MAP algorithm, the main advantage of iCCA-MAP is the significantly reduced computational time, allowing iCCA-MAP to be executed more frequently, thus providing near real-time location information and decreasing the average localization error.

3. The localization error of iCCA-MAP has been compared to that of MCL and Dual MCL. Simulation results show that iCCA-MAP outperforms the MCL and Dual MCL algorithms in finding the location of a mobile node in a WSN with respect to localization error using the minimum number of anchor nodes. When the number of anchor nodes is increased to 20% of the total nodes in the network, both MCL and Dual MCL show significant improvement, as they require a high anchor node density in order to perform well. iCCA-MAP, however, demonstrates very slight improvement with the increase in the number of anchor nodes. Results for iCCA-MAP and Dual MCL are very similar when there are 20% anchor nodes present in the network, with Dual MCL outperforming iCCA-MAP by about $0.02r$. The other parameter varied in these sets of simulations was speed, which seemed to have negligible effect on iCCA-MAP. MCL's performance deteriorates with increase of speed when there are 20% anchor nodes in the network. Higher speeds improve the performance of Dual MCL when anchor node density is very low (i.e., 3 anchors in the network).

5.2 *Limitations*

The proposed algorithm works well in our simulation environment. However, iCCA-MAP does have a number of limitations. They are listed below.

1. The algorithm has thus far only been implemented to localize a single mobile node.

2. As presented, the algorithm requires prior knowledge of the mobile node identity.

3. An intrinsic limitation passed on to iCCA-MAP from CCA-MAP is that accurate results are achieved for networks with average node density of 10 and higher and results are less accurate for network sizes of 30 nodes and smaller.

5.3 *Future Work*

For future work, several avenues could be followed. To that end, here are some directions in which more investigation could lead to intriguing results.

- Determine the optimal frequency for running iCCA-MAP by running it at various frequencies and comparing the average localization error of the mobile node at each frequency. The best results would be a trade-off among localization accuracy, power consumption, the amount of resources available to the node for attending to application-related tasks, and number of message send across the air interface.

- Eliminate the assumption of knowing the identity of the mobile node. Furthermore, the algorithm should detect which nodes are moving and compute their locations accordingly.

- Obtain iCCA-MAP localization results for different network topologies. Examples include: grid square networks, C-shaped networks, loop networks, and pipeline networks. The obtained results can be compared to results from other algorithms in order to draw conclusions as to which network topologies iCCA-MAP is better suited for.

- Implement iCCA-MAP on a small-scale testbed in order to compare simulation results with real world results.

- Expand iCCA-MAP so that a small percentage of mobile nodes can be localized.

- Using simulations, determine the maximum percentage of mobile nodes that can be localized using iCCA-MAP as iCCA-MAP is not suited for localizing an all-mobile WSN. This is because in iCCA-MAP, the mobile node is first removed from the global map and then its local map is patched into the global stationary map. If all the nodes become mobile the global map would seize to exist as they would all have to be removed from the global map.

- Compare iCCA-MAP to variants of MCL and Dual MCL where all non-mobile nodes are used as pseudo anchor nodes. The pseudo anchor nodes' locations will have to be estimated using a localization algorithm such as CCA-MAP or MCL. It would be interesting to see how the localization error results would compare to the results were obtained in Section 4.4, especially since the non-mobile sensor nodes which will be acting as anchor nodes have a localization error and are being used to localize the mobile node. On the other hand, if all stationary nodes are either real or pseudo anchors, the MCL sampling process would improve as more observations could be made. However, the sampling process of Dual MCL could potentially slow down as more constraints have to be met.

- Compare iCCA-MAP to other newly introduced mobile node localization algorithms proposed for WSN, such as Improved MCL (IMCL) [45].

References

[1] I.F. Akyildiz, Y. Sankarasubramaniam W. Su, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 104–112, August 2002.

[2] M. Tubaishat and S. Madria, "Sensor Networks: An Overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20-23, April 2003.

[3] M. Chawla, N. Goel, K. Kalaichelvan, A. Nayak, and I. Stojmenovic, "Beaconless Position Based Routing with Guaranteed Delivery for Wireless Ad-Hoc and Sensor Networks," in *Proceedings of 19th International Federation for Information Processing World Computer Congress*, August 2006, pp. 61-70.

[4] T. Clouser, M. Miyashita, and and M. Nesterenko, "Fast Geometric Routing with Concurrent Face Traversal," in *Proceedings of 12th International Conference on Principles of Distributed Systems (OPODIS'08)*, December 2008, pp. 346 – 362.

[5] H. Kalosha, A. Nayak, S. Rührup, and and I. Stojmenovic, "Select-and-Protest-Based Beaconless Georouting with Guaranteed Delivery in Wireless Sensor Networks," in *Proceedings of 27th Annual IEEE Conference on Computer Communications (INFOCOM)*, April 2008, pp. 346 – 350.

[6] G. Cugola and M. Migliavacca, "A Context and Content-Based Routing Protocol for Mobile Sensor Networks," in *Proceedings of the 6th European Conference on Wireless Sensor Networks*, February 2009, pp. 69-85.

[7] M. Haque, N. Matsumoto, and N. Yoshida, "Context-Aware Cluster-Based Hierarchical Protocol for Wireless Sensor Networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 4, no. 6, pp. 379 – 386, September 2009.

[8] R. Friedman and G. Kliot, "Location Services in Wireless Ad Hoc and Hybrid Networks: A Survey," Israel Institute of Technology, Technical Report CS-2006-10 2006.

[9] Y. Xiao, S. Sethi, H.-H. Chen, and B. Sun, "Security Services and Enhancements in

the IEEE 802.15.4 Wireless Sensor Networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, November 2005, pp. 1-5.

[10] L. Doherty, K. Pister, and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," in *Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, April 2001, pp. 1655–1663.

[11] R. Stoleru, J. A. Stankovic, and S. Son, "On Composability of Localization Protocols for Wireless Sensor Networks," *IEEE Network*, vol. 22, no. 4, pp. 21-25, July 2008.

[12] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Mere Connectivity," in *Proceedings of The ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)*, June 2003, pp. 201-212.

[13] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961-974, November 2004.

[14] Y. Shang and W. Ruml, "Improved MDS-Based Localization," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, March 2004, pp. 2640–2651.

[15] X. Ji and H. Zha, "Sensor Positioning in Wireless Ad Hoc Networks Using Multidimensional Scaling," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, March 2004, pp. 2652-2661.

[16] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, March 2000, pp. 775-784.

[17] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "A Novel Lightweight Algorithm for Time-Space Localization in Wireless Sensor Networks," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, October 2007, pp. 336-343.

[18] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *Proceedings of 6th Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*, August 2000, pp. 32-43.

[19] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28-34, October 2000.

[20] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free Localization Schemes for Large Scale Sensor Networks," in *Proceedings of International Conference on Mobile Computing and Networking (ACM MOBICOM)*, September 2003, pp. 81-95.

[21] D. Niculescu and B. Nath, "DV Based Positioning in Ad hoc Networks," *Kluwer Journal of Telecommunication Systems*, vol. 22, no. 1-4, pp. 267-280, January 2003.

[22] W. Chen et al., "A Localization Algorithm Based on Discrete Imprecision Range Measurement in Wireless Sensor Networks," in *Proceedings of 2006 IEEE International Conference on Information Acquisition*, August 2006, pp. 644-648.

[23] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-Free Distributed Localization in Sensor Networks," MIT LCS, Technical Report TR-892 June 2008.

[24] O.-H. Kwon and H.-J. Song, "Localization through Map Stitching in Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed System*, vol. 19, no. 1, pp. 93-105, January 2008.

[25] L. Li and T. Kunz, "Cooperative Node Localization Using Non-linear Data Projection," *ACM Transactions on Sensor Networks*, vol. 5, no. 1, pp. 1-26, February 2009.

[26] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility Improves Coverage of Sensor Networks," in *Proceedings of The ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)*, May 2005, pp. 300-308.

[27] W. Wang, V. Srinivasan, and K.-C. Chua, "Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks," in *Proceedings of ACM Mobile Communications*, August 2005, pp. 270-283.

[28] K. F. Ssu, C. H. Ou, and H. C. Jiau, "Localization with Mobile Anchor Points in

Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1187-1197, May 2005.

[29] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," in *Proceeding of International Conference on Mobile Computing and Networking (MobiCom 2004)*, October 2004, pp. 45-57.

[30] E. Stevens-Navarro, V. Vivekanandan, and V.W.S. Wong, "Dual and Mixture Monte Carlo Localization Algorithms for Mobile Wireless Sensor Networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, March 2007, pp. 4024-4028.

[31] A.T. Ihler, J.W. III Fisher, R.L. Moses, and A.S. Willsky, "Nonparametric Belief Propagation for Self-Localization of Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809-819, April 2005.

[32] I. Stojmenovic, *Handbook of Sensor Networks: Algorithms and Architectures*, ISBN: 978-0-471-68472-5, Ed.: Wiley Interscience, 2005.

[33] C. Alippi and G. Vanini, "Wireless sensor networks and radio localization: a metrological analysis of MICA2 received signal strength indicator," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, December 2004, pp. 579-580.

[34] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, March 2005, pp. 172-183.

[35] M. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," in *Proceedings of 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, October 2004, pp. 174-183.

[36] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed Online Localization in Sensor Networks using a Moving Target," in *Proceedings of Information Processing In Sensor Networks (IPSN)*, April 2004, pp. 61-70.

[37] C.-H. Ou, "Range-Free Node Localization for Mobile Wireless Sensor Networks," in *Proceedings of 3rd International Symposium on Wireless Pervasive Computing*, May

2008, pp. 535-539.

[38] S.-H. Shee, K. Wang, and Y.-L. Hsieh, "Color-theory-based Dynamic Localization in Mobile Wireless Sensor Networks," in *Proceedings of Workshop on Wireless, Ad Hoc, Sensor Networks*, August 2005, pp. 73-78.

[39] G. Yu and F. Yu, "A Localization Algorithm for Mobile Wireless Sensor Networks," in *Proceedings of IEEE International Conference on Integration Technology*, March 2007, pp. 623-627.

[40] T.-C. Chang, K. Wang, and Y.-L. Hsieh, "Enhanced Color-theory-based Dynamic Localization in Mobile Wireless Sensor Networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, March 2007, pp. 3064-3069.

[41] S. Li, D. Zhang, Z. Yang, and N. Chang, "A Novel Algorithm for Node Localization and Motion Analysis in Wireless Sensor Networks," in *Proceedings of IEEE International Conference on Automation Science and Engineering*, 2006, pp. 574-579.

[42] P. Demartines and J. Hérault, "Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 148-154, January 1997.

[43] Y. Zhou and L. Lamont, "An Optimal Local Map Registration Technique for Wireless Sensor Network Localization Problems," in *Proceedings of 11th International Conference on Information Fusion*, June, 2008, pp. 1-8.

[44] P. Scully and R. Skehill, "Demonstration of Individual Mobility Models in MATLAB," in *Proceedings of IEE Irish Signals and Systems Conference*, 2005.

[45] J.-P. Sheu, W.-K. Hu, and J.-C. Lin, "Distributed Localization Scheme for Mobile Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 516-526, April 2010.