Telematics
Computer Systems

Freie Universität Berlin

Computer Systems and Telematics

Localization of Wireless Sensor Nodes Based on Local Network Density

Dissertation zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.) im Fachbereich
Mathematik und Informatik der Freien Universität Berlin

vorgelegt von

Freddy López Villafuerte

Datum der Disputation: 12.02.2010

Gutachter:
Prof. Dr.-Ing. Jochen H. Schiller, FU Berlin
Prof. Dr.-Ing. Dirk Timmermann, University of Rostock

Institute of Computer Science, Freie Universität Berlin, Germany

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Alle Stellen, die wörtlich oder singemäß aus veröffentlichten Schriften entnommen wurden, sind als solche gekennzeichnet. Die Zeichnungen oder Abbildungen sind vom mir selbst erstellt worden oder mit entsprechenden Quellennachweisen versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner Prüfungsbehörde eingereicht worden.

Berlin, den March 27, 2010

(Freddy López Villafuerte)

Abstract

A wireless sensor network consists of a large quantity of small, low-cost sensor nodes that are limited in terms of memory, available energy, and processing capacity. Generally, these sensor nodes are distributed in space to obtain physical parameters such as temperature, humidity, vibration or light conditions, and transmit the measured values to a central entity. The measurements are tagged with the corresponding location of the nodes in the network and the time of sampling to enable a view of the value distribution in space and time later on. Positioning of wireless sensor nodes without dedicated hardware is an open research question. Especially in the domain of embedded networked sensors, many applications rely on spatial information to relate collected data to the location of its origin. As a first step towards localization, an estimation of the distance between two nodes is often carried out to determine their positions. So far, the majority of approaches therefore explore physical properties of signals such as the strength of a received signal or its arrival time. However, this is problematic since either the complexity on the software or the hardware is not adequate for embedded systems, or the approaches lack the required accuracy. This thesis presents two novel ad-hoc localization systems called FCH (Factor Correction per Hop) and PIV (Positioning Iterative Vector)based on the DIN algorithm (Distance by Intersection of Neighborhoods). The three algorithms are designed to work as distributed ad-hoc locate-sensing methods for indoor environments based solely on the investigation of local node densities. The FCH method is divided into two phases: the first phase obtains the node-to-landmark distances from which to later derived the node positions in the second phase using multilateration. Whereas the PIV algorithm has two operation modi either as an autonomus localization approach (where the first step is acquiring the local neighboring distances to subsequently discover the node position in a iterative manner) or working as an iterative refinement phase. To evaluate the accuracy of these algorithms, extensive simulations and experiments with different testbed setups using a maximal number of 108 real sensor nodes have been conducted. Finally compared the proposed position-sensing algorithms to several distributed algorithms and two RSSI-based locate-sensing algorithms. The position accuracies achieved by applying these algorithms have reached an average position error up to 1.33 ± 0.84 meters.

Zusammenfassung

Ein drahtloses Sensor-Netzwerk besteht aus einer großen Anzahl von kleinen, Kostengünstigen Sensorknoten, die in Bezug auf Speicherplatz, verfügbare Energie und Rechnerleistung stark begrenzt sind. Typischeweise sind die Sensorknoten in einem bestimmten Raum verteilt, um Umgebungsparameter wie Temperatur, Feuchtigkeit, Erschütterungen oder Lichtverhältnisse zu messen. Die aufgenommenen Messdaten werden danach an einem zentralen Rechner per Funk übertragen. Die Messwerte werden im Netzwerk in Form von Tripeln übertragen. Die Tripel bestehen aus dem eigentlichen Messwert, dem Zeitpunkt der Messung und dem Ort an dem sich der Sensorknoten zur Messung befand. Auf diese Weise ist es möglich, die Verteilung der Messwerte relativ zu Raum und Zeit auszuwerten.

Speziell im Bereich der eingebetteten vernetzten Sensoren basieren viele Anwendungen auf räumlichen Informationen, die in Relation mit den gesammelten Sensordaten gesetzt werden müssen. Als einen ersten Schritt zur Lokalisierung wird oft eine Schützung der Distanz zwischen zwei Knoten durchgeführt, um ihre Positionen annähernd zu bestimmen. Die meisten Lokalisierungsmethoden erforschen und verwenden daher die physikalischen Eigenschaften der Funksignale, wie zum Beispiel die Stärke eines empfangenen Signals oder seine Ankunftszeit. Allerdings sind diese Methoden nur bedingt für eingebettete Systeme geeignet, da entweder die Komplexität der Software die Grenzen der Rechenleistung eines Sensorknoten sprengt oder nicht genügend Speicher zur Verfügung steht. Diese Arbeit präsentiert zwei neue Systeme zur Ad-hoc Lokalisierung, im Folgenden genannt FCH (Factor Correction pro Hop) und PIV (Positionierung Iterative Vector). Beide Algorithmen basierten auf dem DIN-Algorithmus (Distance by Intersection of Neighborhoods). Alle drei Algorithmen sind so konzipiert, dass sie die Position von Knoten in einem Ad-hoc Netzwerk innerhalb von Gebäuden nur mittels des Parameters der lokalen Knotendichten bestimmen.

Die FCH-Methode ist in zwei Phasen untergliedert: In der ersten Phase werden die Abstände zu den verschiedenen Referenzknoten anhand der lokalen Knotendichte geschützt. In der zweiten Phase werden diese Abstände für eine Multilateration verwendet, um die Position der Sensorknoten im Netzwerk zu berechnen. Der PIV Algorithmus hat zwei Betriebsmodi: Im ersten Modus wird er zur autonomen Lokalisierung eingesetzt. Hier ermittelt PIV die Abstände zur lokalen Nachbarschaft, um zu einem späteren Zeitpunkt die Position mit einem iterativen Algorithmus zu ermitteln. Im zweiten Modus wird der PIV verwendet um eine iterative Verfeinerung der berechneten Positionen vorzunehmen.

Um die Genauigkeit der Algorithmen auszuwerten, werden verschiedene Simulationen und Real-World Experimente in unterschiedlichen Testumgebungen mit einer maximalen Anzahl von 108 Sensorknoten durchgeführt. Abschließend werden die neuen Algorithmen FCH und

PIV mit mehreren verteilten Ad-hoc Methoden und zwei RSSI-basierten Lokalisierungsalgorithmen vergliechen. Unter Verwendung der oben genannten Algorithmen kann ein mittlerer Positionierungsfehler von $1.33 \pm 0,84$ m erreicht werden.

Acknowledgements

This thesis has been successfully completed thanks to the help of different people and several financial and personal factors. In particular, I would like to express my gratitude to the "Instituto Politecnico Nacional" of Mexico who supported me at the very beginning of my studies and to give me the opportunity to improve my teaching skills.

I greatfully acknowlegde the financial support of the "Deutsche Akademischer Austausch Dienst" (DAAD) A0421560 from October 2004 to Febraury 2008.

Foremost, I would like to warmly thank my doctoral adviser Prof. Dr.-Ing. habil. Jochen H. Schiller whose confidence, tolerance, significant advice, encouragement and constant support allowed me to sucessfully conclude this investigation. I would like to thank Professor Schiller for all the dedication in the proof-reading of preliminary versions of this thesis.

This work owes much to the dedicated and competent advice of my colleague Dr. Kirsten Terfloth whose has shared her time, experience and knowledge, as well as many worthwhile discussions. I have learned from Dr. Terfloth how to write, organize and present these scientific contributions. I really appreciate her invaluable help during difficult moments. Working with her has been a highly educational experience and truly a personal pleasure. Thank you very much for your support and friendship.

I wish to especially express my gratitude to Mr. Tomasz Naumowicz, with whom I shared the office more than once. His advice and constant encouragement helped me enormously in completing all the required tasks. Furthermore, he was always available to share his expertise and deep knowledge of the scatterweb platforms, at the same time offering his unconditional friendship. He has been a strong motivation to continously improve every goal in my work environment.

The realization of this work was not possible without the help of my Mexican colleagues Dr. Ernesto Tapia and Mr. Marte Ramirez. Thank you so much for sharing your knowledge with me and helping me in different stages of this thesis.

I must make a special mention to Dr.-Ing. Achim Liers, Mr. Enrico Köppe, and Mr. Marco Hutta for helping me to understand the RSSI behaviour using the scatterweb nodes on different testbeds achieved at the Freie Universität.

In particular I would like to thank my colleague Georg Wittemburg and Dr. Thomas Zahn for their guidance and support by programming the `ns-2` environment. Especially to Mr. Wittenburg, thank you very much for your kindness and sense of humor. I really appreciate the good moments speaking my mother language with you making me comfortable in Berlin.

I would like to thank my master students Konstatin Clemens, Joaquin González, and Jie

Chien. I have learned so much from them and I have really enjoyed all our discussions about the localization topic.

I want to give an especial mention for Dr. Heriberto Zavala for his advice and help with the Latex editor and to be an exceptionally good friend. I would like to acknowledge the advice of Sabine de Günther who taught me a lot about the German culture. Zavala and de Günther never let me down in the difficult times being a real support throughout my thesis writing.

I deeply appreciate the help of my friends such as Aldo Mirabal, Azucena Arango, Martín Henze, Amable Mendoza, Salvador Rodriguez, Lourdes Rivera, David Jiménez, Angelica Martínez, Ralf and Susanne Digel, and Irma Martínez for their generous encouragement and support, which allowed me to finish this work.

My most valuable inspiration and force while writing have been my parents, my sister and my brother. I have learned from them that the best way to realize your dreams is through constant work every day. I want to thank their unreserved support by dedicating this work to my parents Mr. Luis López Moreno and Evelia López Villafuerte.

Dedicated to Evelia López Villafuerte and Luis López Moreno, my parents, who introduced me to the fascinating adventure of discovering the nature of things. With gratitude, admiration and love.

"...I realized I had to revolutionize my self, learning new things to keep up. I realized and I rebelled."

Jaime Sabines

Mexican writer and poet

Contents

List of Figures

List of Tables

List of Symbols

A_i	Intersection area of the overlapping transmission ranges
A_t	Complete deployed area
A_u	Union area of the overlapping transmission ranges
K_i	Number of nodes in the intersection area
K_u	Number of nodes in the union area
DGPS	Differential Global Positioning System
DIN	Distance by Intersection of Neighborhoods
DV Hop	Distance Vector Hop
ExDIN	Extension of the DIN algorithm
FC	Factor Correction
FCH	Factor Correction per Hop
GPS	Global Positioning System
IR	Infrared signal
LAN	Local Area Network
LBS	Location-Based Services
LMI	Linear Matrix Inequalities
LORAN	Long Range Navigation System
LOS	Line of Sight
MDS	Multidimensional Scaling
MSB	Modular Sensor Boards
PDA	Personal Digital Assistant
PIV	Positioning Iterative Vector

RF	Radio Frequency
RFID	Radio Frequency IDentification
RND	Real Neighbor Distance
RSSI	Received Signal Strength Indicator
RTMM	Rosum TV Measurement Module
SDP	Semidefinite Program
SPOT	Smart Personal Object Technology
SRR	Space-Range Ratio
TDOA	Time Difference of Arrival
TOA	Time of Arrival
UWB	Ultra Wide Band
WDNI	Weighted Density Node Intersection
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Networks
WSN	Wireless Sensor Network

CHAPTER 1

Introduction

Wireless Sensor Networks (WSN) [2] store and partially process sensed data either within the same sensor nodes which take the local samples or transmit the sensed data to a remote central computer where the data will receive a bigger and more complex handling process. Generally, the sensor nodes are distributed in space to obtain physical parameters at that location such as temperature, humidity, vibration or light conditions.

To have a record of the place of study, it is very important to correlate the collected measurements sensed by the nodes to a specific location. Furthermore, the position of the nodes opens up new ways to detect special events, tracking an object of interest, and improve the network coordination by executing geographic routing algorithms.

The estimation of the position of every node in the network is still an open question. The location problem is especially crucial in WSN due to the intrinsic properties of the sensor nodes which have limited compute capability, slow store memory, and scarce energy. For these kind of devices, it is necessary to find methods that work in an ad-hoc fashion, without additional specialized hardware to save their scarce resources.

1.1 Motivation

Perhaps, the most well known technique to find location is the Global Positioning System (GPS). However, the implementation of GPS chips on wireless sensor nodes is difficult because very low power chips adequate for WSN's do not exist and the GPS receivers have the strong disadvantage that they do not work indoors. In addition, the named "urban canyons" where dense blocks of structure, especially skyscrapers, contribute to lost GPS signals. Therefore, the necessity to locate wireless sensor nodes is still a topic of interest for the research community.

Several proposals to find the position of wireless sensor nodes have been developed in recent years. The first step in this direction is to estimate the distance between nodes. To obtain this information, there is a variety of techniques that exploit physical phenomena such as the arrival time of sound signals [76], the time difference of arrival between radio and ultrasonic signals [59, 9], the use of interferometry [43], radio signal strength indicators [40], or the use

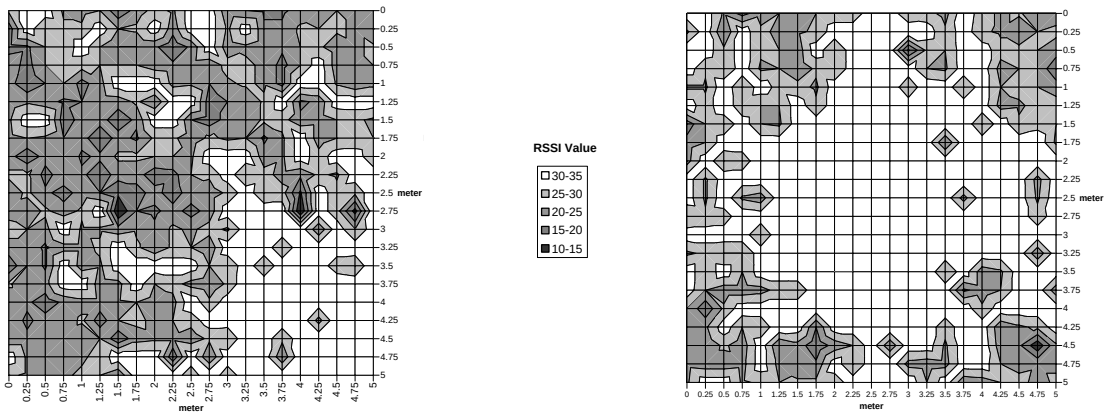
of camera pictures with a previous scene analysis [46]. These techniques and some others will be explained in the next chapter of this work.

1.2 Problem Statement

The determination of the distance between sensor nodes that are close to one another (within the range of tens of centimeters up to a few meters) is usually carried out with the help of Time of Arrival (TOA) or Time Difference of Arrival (TDOA) systems. The accuracy of these systems comes at the cost of a high synchronization overhead, thus high energy expenses at runtime and the need for dedicated hardware on the sensor nodes [59].

In contrast, range-free algorithms rely solely on the conventional hardware of sensor nodes, with the preferred technique to conclude the distance of the receiving node from the sender by means of mapping the measured RSSI value to a distance. This mapping has to be justified by previous measurements, but has the advantage of no additional cost on a node since it is provided by the transceiver practically for free.

To understand the distribution of RSSI values in an indoor setup, we measured these values with our Modular Sensor Nodes (MSB), (see section 4.3.1) at regular points and created maps, two of which are depicted in Figures 1.1 a and 1.1 b.



(a) (b)

Figure 1.1: (a) Received signal strength of a sending node placed at the lower right corner within an indoor testbed, (b) Received signal strength of a sending node placed in the middle of the network within an indoor testbed

These maps visualize very well the problem that arises when utilizing a simple mapping: As can be seen in map 1.1 a the transmission range is far from being regular, nodes may be far away from the sender and still receive a high RSSI value, while others are closer and are exposed to lower values, and thus will miscalculate their distance.

Fluctuation of the received signal imposes a major challenge on current range-free algorithms. Also, Figure 1.1 b indicates that the determination of a small distance in the range

of tens of centimeters is not possible, since the resolution of RSSI does not allow for such an accuracy. Even worse, the distribution of RSSI values is influenced by spatial, temporal and environmental parameters, the orientation of the antenna and the choice of transceiver, making the calibration an almost unaccomplishable challenge.

1.3 Objectives

The main goal of this work is to locate the position of sensor nodes using solely the node's radio communication chip to conserve the scarce energy in the WSN devices. Another desirable characteristic in this area is the use of algorithms which creates cooperative localization, that is to say, sensor nodes should work together to make measurements with the help of local data transfer and then form a map of the network where the nodes locate themselves, despite the wrong data produced by the hostile environment.

Various application requirements such as scalability, energy efficiency, and accuracy have to be considered to design a robust and ad hoc localization algorithm. The indoor environments are specially a big challenge because of the problems with the radio signal due to interferences, reflections, multi paths, signal attenuations, shadowing, and scatter effects that appear in a random manner with a severe fluctuation. In this thesis, two novel methods to estimate the location of wireless sensor nodes indoors based on local network density are proposed.

In order to obtain competitive position accuracies, first we suggest a method to improve the case of a multihop environment using the unknown node to reference node distances and computing node positions by multilateration. As a second schema, we combine the internode distances produced in a given local area of the network to later (and conjunction with the whole network information) find the estimation of different spatial points that minimize the global and local node distances errors. Both methods are compared with recent approaches not only by extensive simulations, but tested with real hardware as well. For the first time, the comparison and implementation of different positioning methods is in conjuntion with our novel way to alleviate the problem of radio fluctuation to produce more reliable inter node distances.

1.4 Structure of this thesis

The structure of the remaining thesis is as follows: First, I present basics of WSNs in terms of typical parameters to classify different methods.

In chapter 3, we show the classification of different positioning algorithms, practical applications of trading platforms, and prototypes. Furthermore, we discuss several approaches which relate node density in a network to node distances as a first step to find position estimations.

A mathematical model that relates node distances with the local node density is introduced in chapter 4. This will serve as a foundation of the proposed DIN algorithm. Making use of the `ns-2` simulator, we probe the quality of DIN in uniform and near-uniform node distributions with different node densities. There, the necessary steps to put into practice

the DIN algorithm on real sensor nodes are presented. Furthermore, a comparison against a RSSI-based distance estimation is yielded to verify the validity of the DIN algorithm on different network settings.

In Chapter 5, the DIN algorithm is applied to the localization field through novel different localization approaches. Once again, the impact of different network configurations with different nodes densities is probed with the help of the `ns-2` simulator. The extensive simulations using the new position-sensing algorithms to locate nodes are compared to different localization approaches and finally, the practical evaluation of these new algorithms with the ScatterWeb Platform is shown.

In the same chapter, the mathematical foundation of a new algorithm, called PIV, is also presented. The quality of the PIV algorithm is verified by multiple simulations over different network settings and densities. We investigate the efficiency of the PIV algorithm as a pure positioning algorithm, as well as a refinement algorithm, comparing the position accuracies with different schemes. A distributed localization setup based on the PIV algorithm structured in three phases and implemented on real sensor nodes is discussed. The summary of our findings are found in chapter 6 and finally the future work of this investigation is found in chapter 7.

CHAPTER 2

Basics of Wireless Sensor Networks

A Wireless Sensor Network (WSN) is formed by dozens of small, low-cost nodes which have limitations in memory, energy, and processing capacity [2]. In this type of network, one of the main problems is to locate each node.

The vision of many researchers is to create smart environments, controlled through planned or ad-hoc deployment of a potentially large set of sensor nodes, each with transceivers for wireless, short-range communication, each capable of detecting environmental conditions such as temperature, movement, light, acoustic events or the presence of certain objects.

WSNs will enable a very close observation and control of the physical world. The future of sensor networks appears to be in large numbers of unattended autonomous nodes which operate in a dynamic environment which will be able to organize itself. It will be aware of its physical position and will carry out dynamic tasks in a distributed form, very frequently confronting changes in the network topology and failures in the network nodes due to the lack of power, physical damage or environmental interferences.

These nodes will be able to measure and report on environmental characteristics such as temperature, pressure, humidity, vehicular movement, noise levels, lighting conditions, the presence or absence of certain kinds of objects, acoustic events, mechanical stress levels on attached objects, and so on. In other words, one can say that localization will act as a bridge between the virtual and physical world [20].

2.1 Parameter for Localization

For each method of estimating location information, it is possible to name specific parameters to establish the similarities and differences between the various techniques. In this section, the most typical parameters to classify different approaches will be presented.

2.1.1 Accuracy and Precision

The most important parameters for localization techniques are accuracy and precision [29]. Accuracy could be define as the extent by which the estimated location deviated from the

actual position is. Precision indicates how often we expect to achieve the given accuracy at the very least. (e.g. 20 cm accuracy across 95% of the time).

2.1.2 Scalability

The number of objects to be located in the system plays an important role. Each system has its own limit in the number of objects it can find in a certain time frame depending on the infrastructure per area.

The increase in localized objects in the network creates higher transmission demands which could congest the communication channel if a given threshold is exceeded. To offset this problem, another important consideration in the location-sensing systems is the selection of the best-fit radio frequency technology. The speed by which the location system outputs the position information is known as responsiveness or sampling [20]. The responsiveness is directly related to the maximum number of nodes the network can manage.

A location-sensing system could be used to find tags, objects, people, assets or animals on the surface of the earth, in a city, buildings or in a single room. These systems can be roughly classified into two different groups of scalability, either a system which works in areas outside or those which locate within inside areas. This classification lets us identify special problems that the system will have to address such as diffraction, multipath and interference problems.

2.1.3 Self-Organization, Autonomy and System Organization

The degree of autonomy has some of the most significant consequences to the system design; this parameter is closely related to the scalability of the system. A system has high autonomy when it requires little or no human intervention to operate the system and the nodes act as completely independent entities.

Autonomy of a given system is achieved through the use of extensive and sophisticated internal processes that make their own coordination possible. The self-coordination of the network is important because it impacts the ability to extend the system.

A roughly classification of autonomy and self-organization divide WSNs into centralized or distributed systems, meaning the system may not require the help of a central entity to monitor and control the activities of the elements.

2.1.4 Cost

Computing the cost of the location sensing system requires looking at several different factors including installation time, money, effort required for computation, and energy.

The effort required for computation is a crucial parameter closely related to the location algorithm of the system. This parameter, known as computational cost as well, determines whether the location system is organized centrally or distributed.

The centralized systems control and monitor the system functions with the help of a central engine. However, in the decentralized or distributed systems, the location algorithm is

spread out into each node within the network, allowing the individual computation of the position.

The time cost requires evaluating factor such as the length of the installation process and the system administration needs. The time cost is also related with the efficiency to find the node positions in the network, this time depends on the algorithm to locate the nodes and the organization of the system. Finally but not at least the financial cost of the whole system is strong related on set up and operation, including factors such as infrastructure, the salaries of support personnel and the maintenance of the system.

Figure 2.1 shows the localization taxonomy with the definitions mentioned in these last sections.

2.2 Localization Techniques

Location discovery is based on three fundamental phases: a measurement phase that produces a set of distances measurements, presence hops, angularly or optically to/from a set of anchor points; a combining phase that brings together the measurements to produce an estimation of a position; and lastly, the optional refining phase that improves the estimated positions through an iterative procedure [38].

The automatic location-sensing systems can be classified into three groups:

1. Proximity-based approaches
2. Lateration
3. Angulation
4. Analysis of Prior Survey Location

This classification is based on the information that a node uses about a node's neighborhood (proximity-based approaches), the exploiting geometric properties of a given scenario (lateration), or the characteristic analysis of the position of a node in comparison with pre-measured properties (analysis of prior survey location). Most recent papers describe location-sensing systems which employ one or more techniques of this classification, becoming hybrid systems.

2.2.1 Lateration

Lateration, an important concept in relationship to the localization area, computes the position of an object by measuring its distance from multiple reference positions.

In a plane, the position of a node can be derived by the distance measure from 3 non-collinear points. Nodes which know their exact position a priori or through the help of specialized hardware (e.g. GPS) are called anchors, landmarks, seeds, reference nodes or beacons [33]. For lateration, the extension to three dimensions requires 4 non-coplanar distances to the beacons to estimate the position.

Lateration uses a method known as “least squares" to estimate a particular position from a set of linearized equations in the form of $Ax = b$. The least squares method is efficient

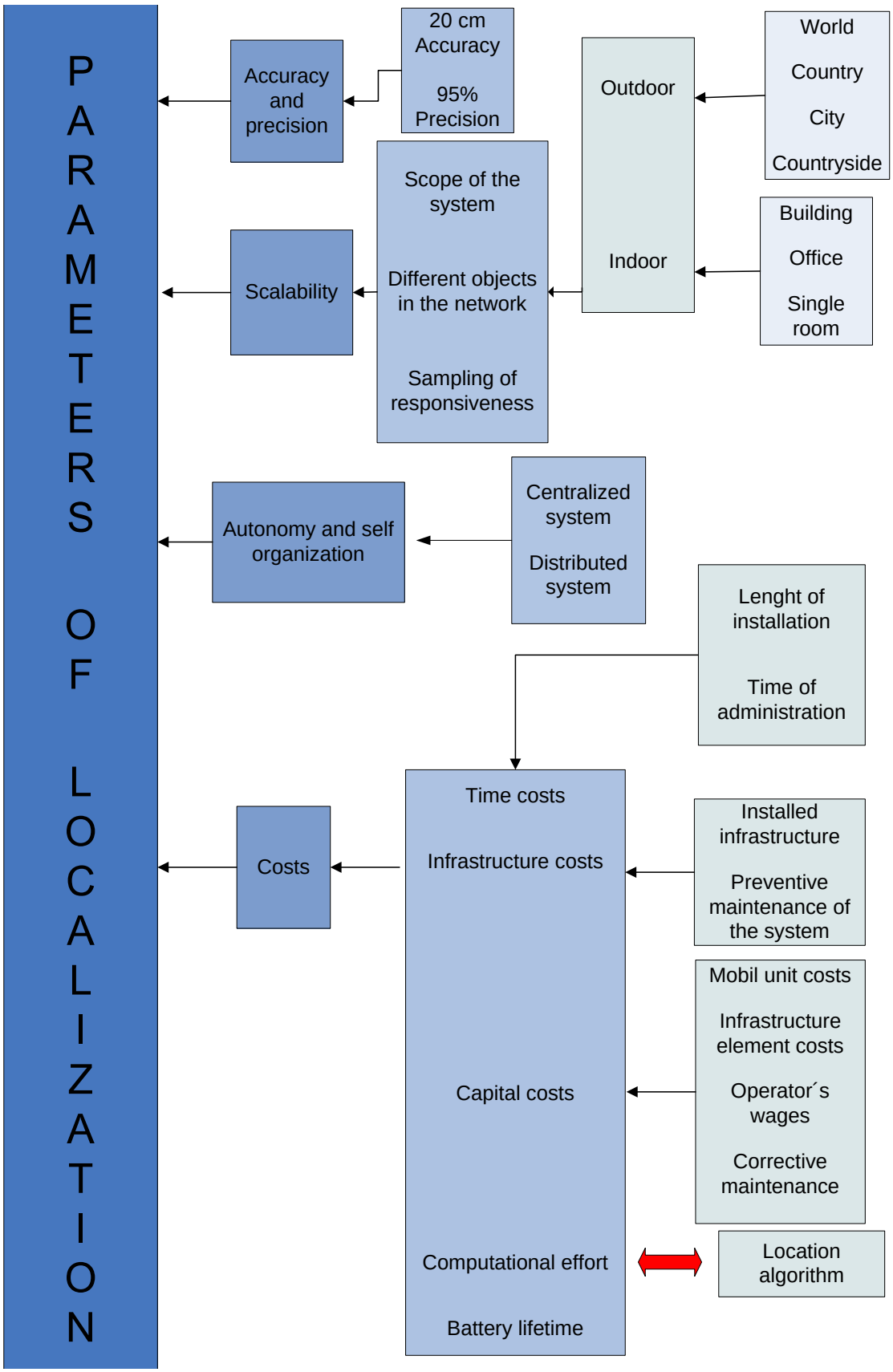


Figure 2.1: Localization Taxonomy

because it minimizes possible range estimation errors accumulated along the propagation path. Appendix A takes a closer look at the mathematics behind it.

2.2.2 Angulation

This method estimates the positions of nodes using angular measurements from the reference nodes instead of distance measurements. The estimation of a spatial point in a plane can be determined with only two reference points and for tridimensional spaces with just three angular measurements. The system proposed in [50] uses RF transmission from rotating directional antennas to determine the location of wireless sensor nodes by applying this technique. I will discuss direction-finding techniques in section 2.4.

2.2.3 Proximity

The simplest technique involves finding how “close" an object is to its known location. The node senses the object by using a physical phenomenon with limited range, like a radio signal, magnetic field, or IR light. If the node detects and recognizes the object within the range of the emitter then the node is able to run an algorithm to determine the approximate position of his neighbor. The proximity technique is also known as the range-free approach [10].

2.2.4 Analysis of Prior Survey

The analysis of prior survey location techniques uses features of a scene or area observed from a particular vantage point to draw conclusions about the location of the observer or objects in the scene. In static scene analysis, observed features are looked up in a predefined dataset that maps them to object locations.

In contrast, differential scene analysis tracks the difference between successive scenes to estimate location. Differences in the scenes will correspond to movements of the observer. If the specific positions of the features are known, the observer can compute his own position relative to them [74].

Nowadays there are systems that work with a hybrid analysis of prior survey systems. Such systems work a priori in the covered area and extract samples of different points by (multi)lateration or proximity with the purpose of constructing a reference map of the area. Later the devices find their own location by comparing the map samples obtained off-line in the area with the actual readings by the devices [40]. The major drawback of this method is that it requires substantial computational effort and sometimes demands a large quantity of physical memory.

2.3 Distance Estimations

A system based on lateration, multilateration or hybrid analysis of prior survey location requires estimation of distances between the beacons and the nodes deployed to discover the position of the unknown nodes. The most effective approaches estimating this distance between nodes or between nodes and landmarks are: Signal Attenuation, Time of Arrival

(TOA), Time Difference of Arrival (TDOA) and Interferometry, which will be introduced in this subsection.

2.3.1 Signal Attenuation

This method can be implemented in any system able to achieve radio communication. This method is also known as Received Signal Strength Indicator (RSSI). RSSI operates on the general principle of a law of physics related to the loss of signal strength in space where the signal strength loss is proportional to $1/d^2$ where d is the distance between sender and receiver. In real scenarios, the distance relation could be $1/d^3$, $1/d^{3.5}$, $1/d^4$, etc. depending on the environment where the nodes are located. There are some experimental results where the application of this approach is developed in an indirect way, similar to [53], where O'Dell et. al. use the loss of sent packets to determine distances between sensor nodes.

The main disadvantage is that RSSI values oscillate heavily due to signal propagation issues such as reflection, refraction and multipaths. These circumstances can be found even when sender and receiver do not move.

2.3.2 Time of Arrival

The Time of Arrival method (TOA), also called "Time of Flight", exploits the relationship between the beacon-node distance and the transmission time of the signal between sender and landmark.

Assuming that the sender and receptor know the starting time of a transmission, if the signal velocity is known, then the arrival of the signal indicates the landmark-node distance. This distance can then be computed by every node in the network as presented in [8].

The two main disadvantages of this method are: first, it is necessary to have a synchronized sender and receiver; second, depending on the transmission medium that is used, a high clock resolution is required to produce results of acceptable accuracy. For example, for acoustic waves, this requirement is modest (about microseconds) but for radio, a very high resolution is necessary (about nanoseconds).

2.3.3 Time Difference of Arrival

The Time Difference of Arrival approach (TDOA) can be utilized if two transmission media of very different propagation speeds can be accessed. For example, radio waves propagating at the speed of light and ultrasound.

To estimate distances, the sender must simultaneously transmit signals of different speeds. The receiver can use the arrival of the first faster signal to start measuring the difference in arrival time of the second slower signal, safely ignoring the propagation time of the first signal. The time registered by the receiver is proportional to the distance between sender and receiver [80].

The necessity of two different types of receptors and senders in every node, the limited range of certain signals like ultrasound waves and the synchronization requirement in every device however, are disadvantages to this approach.

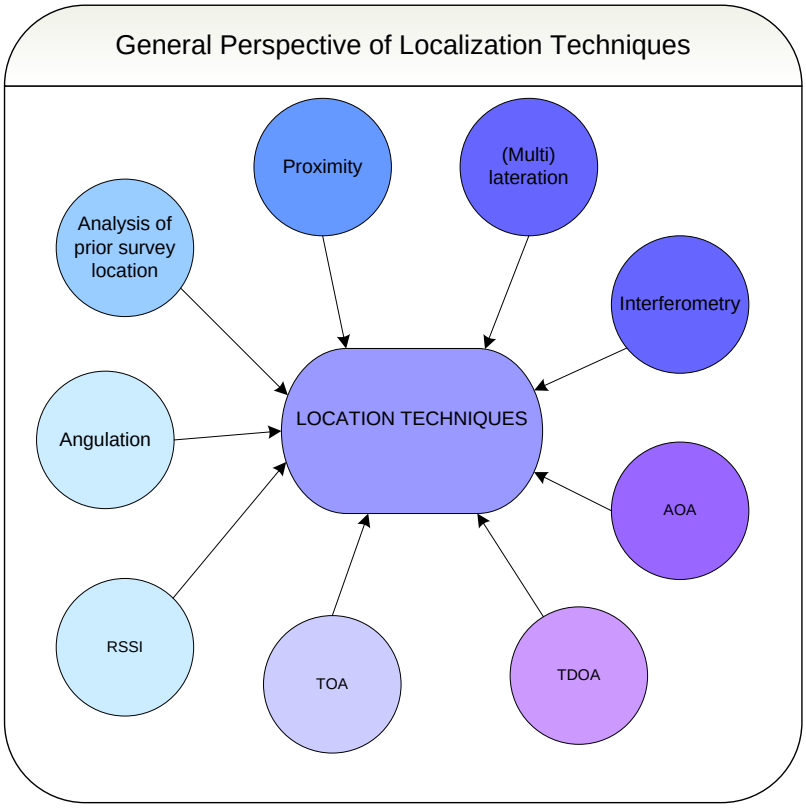


Figure 2.2: Scheme of Localization Techniques

2.3.4 Interferometry

This approach [43] is useful if the system working environment of the nodes has low mobility and is free of physical obstacles. The method uses the superposition of the fields of two separate radio transmissions to create interference. Two senders and two receivers use the radio interference to obtain distance relationships between each other. The nodes create and solve the equation sets to find their relative positions. The solution has good accuracy (approximately 10 cm), but at the cost of relative high synchronization effort, Line of Sight (LOS), and high computational effort requirements.

2.4 Angle Estimations

This technique can be separated into two different categories as mentioned in [17]:

1. Anisotropy by receiver's amplitude response
2. Anisotropy by receiver's phase response

The first schema makes use of the the reception beam pattern of directional antennas, locating the direction of the transmitter by the maximal signal strength while the beam antenna pattern is been rotated mechanically or electronically. Some strong disadvantages

of directional estimations, however, are spatial and temporal fluctuations of the transmit power of the propagated signal, the high-rate of pulses to be intercepted by the antennas, and the fact that the receiver might have to be held wide open to maximize signal reception probability.

To reduce the varying signal strength, the use of a minimum of two (but typically at least four) stationary antennas with known, anisotropic antenna patterns is proposed in [17]. Overlapping the stationary antennas pattern and comparing the signal strength received from each antenna at the same time yields the transmitter direction, even when the signal strength changes. Once again, the problem of this technique is the energy consumption and the big form factor of the devices.

Römer in [65] proves that angle data can also be obtained using optical communication which partially alleviates the problem of energy consumption and big form factor, but with the strong disadvantage of LOS requirement.

The second technique uses the phase differences of a wave front from a signal transmitted. To compute the AoA, such systems require mounting an antenna array composed of N antenna elements. The adjacent antenna elements are separated by a uniform distance S. The distance between a transmitter agent and the ith antenna element can be approximated by

$$d_i \approx d_1 - iS\cos\theta \tag{2.1}$$

where d_i is the distance between the transmitter and the i-th antenna element, and θ is the relation on the transmitter with respect to the antenna array (see Figure 2.3). The phase difference in every antenna can be determinated by $2\pi\frac{S\cos\theta}{\lambda}$, which allows us to obtain the direction of the beam from the transmitter through the measurement of the phase difference.

Although this technique works well even when confronted with a high Signal-to-Noise Radio communication media, the presence of multipath and/or strong co-channel interference have a very negative impact on the approach [63].

A third classification that could be consider a combination of the two techniques explained above is found in [23]. This subclassification is the so called subspace based algorithms. This technique uses a vector space formulation, which takes advantage of the underlying parametric data model for the sensor array problem. The technique can be explained as follows: first, all the signal measurements are considered as vectors which form M arrays elements. These arrays are visualized as vectors in M dimensional space. Using an eigen-decomposition of the correlation matrix, the vector space is separated into signal and noise subspaces. The estimations of the emitter direction are finally obtained looking for zeros in the magnitude squared of the projection of the direction vector onto the noise subspace [73, 7, 34].

Other approaches such as [66] and [67] based the angle estimations on rotational invariance techniques using two displaced subarrays of matched sensor doublets to later exploit an underlying rotational invariance among signal subspaces. Examples of experimental testbeds using subspace based algorithm can be found in [58]. We refer the readers to [72] where a comprenhensive and detail discussion about angle estimation techniques are also available.

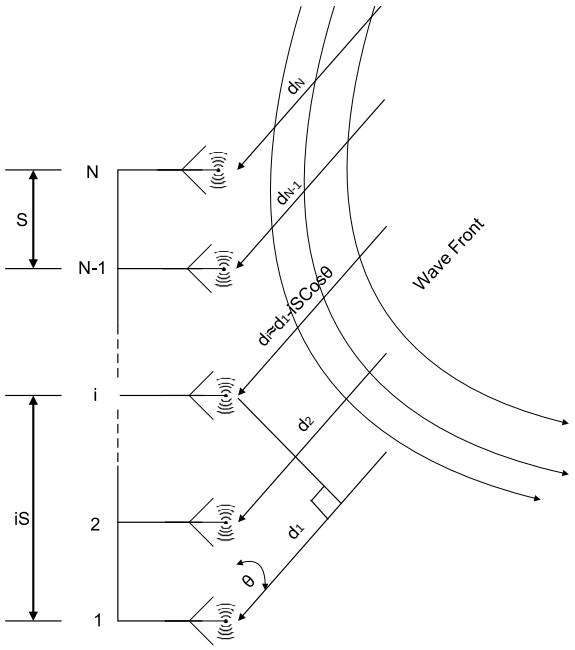


Figure 2.3: Scheme of an Antenna Array

To give a general perspective of the different distance and angles estimation techniques and the different localization approaches explained in the last sections, we resume them in a graphical scheme of Figure 2.2.

2.5 Signal based Technologies

The following section gives an overview of signal based technologies used in WSNs. Some advantages and disadvantages of the most typical signal based technologies, their usage in different environments, and some examples of applications will be touched upon in this section, but a more detailed study will be presented in chapter 3.

Signal based technologies could be divided into three different categories: Technologies that use mechanical waves signals, systems that use different electromagnetic radiations and, finally hybrid signal based technologies.

The most well-known example of mechanical waves is sound. Another option to estimate distances and location in WSNs is using ultrasound signals. These mechanical waves have frecuencies greater than 20 Kilohertz, which is the upper limit of human hearing. Some applications like [9, 60, 59] and [76] report accuracies in the order of centimeters, but at the cost of working outdoors in semi-controlled environment with a high degree of time stamp calibration and direct LOS requirement.

The most popular signal based technologies used by the majority of location-sensing systems is electromagnetic radiations. Depending on the frequency used, they be categorized as optical and radio frequency signals.

Optical signals-based systems such as [65] use a laser emitting visible coherent light $650nm$.

These kind of systems are inexpensive and have nodes with a compact form factor. The typical range of such systems is up to 5 meters. The main disadvantage, however, is that the system does not work in direct sunlight and participating nodes must be in the LOS.

Another optical schema is proposed in The Active Badge System [79]. This system uses a blink of an Infrared signal (IR) to detect the presence of an object. On one hand, the system is attractive due to inexpensive deployment (being cheaper than the systems based on laser emitters [65]), the compact form factor of the tags and the low energy consumption. On the other hand the system is restricted to the LOS conditions with low responsiveness.

Systems based on radio frequency signals like [83] works with electromagnetic waves centered on 2.4 GHz and use several pulses to confirm the presence of an object. The range of this system is about 5 to 6 meters, but it has a large form factor, an expensive infrastructure focused on a given application.

Systems such as [3] and [13] use the so called Radio Frequency IDentification (RFID). . The RFID is an automatic identification method that uses devices with miniaturized oscillator and encoder so as to generate a programmable radio modulated pulse signal. These devices called RFID tags typically transmitt radio frecuencie pulses between 840 and 960 MHz. In chapter 3 we will describe better this kind of systems that use RFID to store and remotely retrieve environment data.

Wi-Fi is a brand originally licensed by Wi-Fi Alliance to describe the underlying technology of Wireless Local Area Networks (WLAN) based on the IEEE 802.11 specifications. Systems such as [82, 31] and [40] use this technology to implement localization.

Bluetooth is an industrial specification for Wireless Personal Area Networks (WPAN) based on the IEEE802.15.1 standard. Systems like [56] and [64] use this technology to locate devices.

Ultra Wide Band (UWB) is a technology for transmitting information spread over a large bandwidth for WPAN. [77] and [39] are systems that use this technology to discover the position of the nodes.

Spread Spectrum for ANSI 371.1 is a technique that is used in [83] for localization. Others technologies based on Radio Frequency signals (RF) are Zigbee (IEEE802.15.4), DC-Electromagnetic Pulses applied in [47], Digital Television signals used in systems such as [61], Commercial Radio FM Stations signals proposed by [35] and [87], wide area cellular [36] and satellite based [32].

Table 2.1 describes some of the drawbacks and advantages of different localization signal based technologies. A detailed discussion of all this signal based technologies and some systems that use them to locate devices will be presented in chapter 3.

2.6 Conclusion

The localization parameters accuracy, the scalability, the cost, and degree of autonomy of a given locate-sensing system can be used to determine which platform is advisable to install for a given application. The different localization techniques explained above show varying levels of complexity, cost and position estimation errors. However, we can realize that

Table 2.1: Taxonomy of Localization Technologies.

Signal Based Technology			Advantages	Drawbacks
Mechanical Waves		**Ultrasound**	Accuracy in the order of cm Low Clock Rates System Inexpensive and Simple	Necessity of Calibration and Synchronization Temperature, humidity and pressure dependence Typical Range is between 3–15 m
Electromagnetic Radiation	**Optical**	**Visible Light**	Cheap System Small Form Factor Low Energy Consumption	Line of Sight Requirement Sensible to Fluorescent and Sunlight Typical Range of 5 m
		Infrared	Inexpensive Small Form Factor Low Energy Consumption	Line of Sight Requirement Low Responsiveness Sensible to Fluorescent and Sunlight Typical range of 5 m
	Radio Frequency	**RFID**	Cheap Tags Long Battery Life 3–4 m Accuracy	Very Low Power Consumption Centralized System Typical Range 1–10 m
		Wi-Fi	Potential Reuse of Deployed Network No Line of Sight requirement	High Power Consumption Typical Range Wi-Fi. 50–180 m Sensible at Interference
		Bluetooth	Low Power Consumption Low Cost Hardware	Typical Range Bluetooth 10–120 m Interference with Wi-Fi. signals
		UWB	Low Power Consumption Less Affected by Multipath Accuracy < 1 m	UWB is limited at Bluetooth range Expensive
		Spread Spectrum	Low Power Consumption Cheap Transceivers	Spread Spectrum for ANSI 371.1 Typical Range 10–300 m
		Zigbee	Low Power Consumption No Line of Sight Requirement	Typical Range Zigbee 10–100 m No Proper Propagation Model Exist
		FM Radio	Commercial FM Radio Range is City Very Cheap receptors Low Form Factor	low precision No licence-free bands
		Digital Television Radio Signal	Radio signal Range is City	Expensive No license-free band
		Cellular System	Typical Range is 100–150 m	Low accuracy Expensive Infrastructure
		GPS	Range of Satellite Based System is World Wide	It does not Work Indoors and in Urban Canyons Expensive Infrastructure Affected by Multipath and Ionosphere Propagation Delay
		DC-Electromagnetic Pulses	High Accuracy in the order of mm High Responsiveness	Typical Range of up to 5 m Sensible to metallic objects Expensive

there is a relationship between the system cost, the localization techniques and the node's distance and/or angles alternatives. For example, the RSSI based approaches are regularly simpler and do not require any additional hardware compared with complexed techniques such as Angulation techniques or multilateration using ToA or TDoA. The simpler the base technology and the algorithm used to locate the nodes is, the lower power consumption, communication overhead and system cost but with the inconvenient of lower accuracy.

Current research is strongly influenced using multilateration and time based techniques, which improve node position accuracies, but only in a localized area (as we will see in Chapter 5). Furthermore, the implementation conditions such as calibration and sinchronization of the nodes, LOS requirement, time delays of the transmitted acoustic signal, and the physical misalignment, as well as diferences between the time stamp and the actual transmission time [57] are strong disadvantages of these methods.

Researchers are still busy inventing new localization algorithms that work together with different signal based technologies in ad-hoc and global manners. These new localization algorithms have to be robust against the measurement errors produced by the environment and the intrinsic signal based technologies. At the same time, the new algorithm designs have to take into consideration the limited capabilities of the sensor nodes producing competitive position accuracies and saving scarce resources.

In chapter 3, different proposed algorithms and some prototypes developed in the last years will be described. A comparison of several practical and commercial applications will be presented using the localization parameters explained in this chapter.

CHAPTER 3

Localization Algorithms and Related Work

As mentioned in the last chapter, sensor networks are typically quite resource-starved. The sensor nodes have a rather weak processor, making heavy computation unfeasible. Therefore, the localization algorithms which run on the nodes of the system are a crucial parameter for this kind of networks. In this chapter the classification of different positioning algorithms, practical applications of trade platforms and prototypes in the localization area are shown. Finally, approaches which relate node density in a network to node distances and node position estimations are discussed.

There are two tendencies in the categorization of localization algorithms. The first refers to the computational organization and classifies the localization algorithms into centralized or distributed algorithms.

Centralized algorithms are designed to run on a central machine with a large amount of computational power. Here, the sensor nodes obtain environmental data and pass it back to a base station for analysis. The central computer processes the data and sends the estimated position back into the network. Centralized algorithms evade the problem of a node's computational power by accepting the communication cost of moving data back to the base station.

The second tendency as described in [44] and [71] is to divide localization algorithms into range-based and range-free algorithms.

Localization systems for WSNs that use connectivity-based algorithms among nodes for calculating new positions are known as range-free localization algorithms. Those systems that use methods which deal with distance measurements are called range-based localization algorithms.

Taking a closer look at different localization algorithms, one notices that many of them are sensitive to the network parameters such as node density, edge Network area, environment obstacles and terrain irregularities.

3.1 Centralized Algorithms

Central localization algorithms permit the employment of much more complex mathematics than distributed algorithms allow. The type of process performed at the base station helps to distinguish and to classify them.

The most representative algorithms in our analysis are: Semidefinite Program and multi-dimensional scaling.

3.1.1 Semidefinite Program

In [18], Doherty et al. use a Linear Program (LP) and a Semidefinite Program (SDP) to solve the positioning problem. In this algorithm, they represent geometric constraints between nodes as Linear Matrix Inequalities (LMI). Once all constraints in the network are expressed in this form, the LMIs can be combined to form a single semidefinite program. This semidefinite program produces a bounding region for each node, which is simplified into a bounding box in [75]. In Figure 3.1 the green node represents a system beacon and the white circle with a question mark is the node whose position is constrained by some sensing form from the beacon.

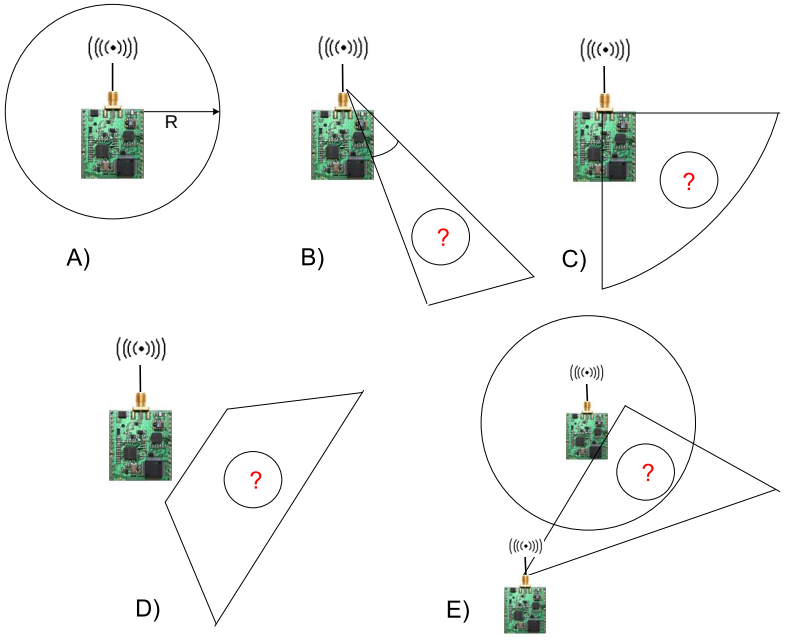


Figure	Name	Example
A	Radial constraint	Radio connectivity
B	Triangular constraint	Angle of arrival data
C	Quadrant constraint	A combination of radio range and angle of arrival
D	Trapezoid constraint	Variable radio connectivity and angle of arrival
E	Hybrid constraint	Derived from an intersection of two constraints

Figure 3.1: Possible constraint forms to work with SDP

The simplicity of modeling theoretical measurements approaches as distance or angle estimations makes methods based on SDP highly advantageous. Furthermore, it works in a range-free modus since it uses simple connectivity. Unfortunately, not all constraints can be expressed as LMIs. One cannot conveniently depict radio range data since transmission rings cannot be expressed as convex constraints.

Solving a linear or semidefinite program must be done centrally. The relevant operation is $O(k^2)$ for angle measurement, and $O(k^3)$ when radial (e.g. hop count) data is included, where k is the number of convex constraints needed to describe the network. The accuracy of this method depends on the number of nodes and the number of neighbors. Furthermore, the algorithm's run-time is a severe weakness because it is highly incremented for large and dense networks which might be harder to solve. The authors of [18] have shown that node locations can be estimated between $0.64R$ and $0.72R$ (where R is the radio transmission range) at a density of 5.6 neighbors per node.

3.1.2 Multidimensional Scaling

Shang et al. in [75] propose a centralized algorithm called MDS-MAP that uses only connectivity to estimate the node's positions in a network with or without landmarks. This method can be considered range-free, since the algorithm uses a mathematical technique called Multidimensional Scaling (MDS).

MDS can be described as follows: Suppose there are n points in a given volume where their positions are unknown, but it is possible to know the distance between each pair of points. From here, multidimensional scaling uses the Law of Cosines and Linear Algebra to reconstruct the relative positions of the points based on the pairwise distances.

MDS-MAP is almost a direct application of the "classical metric MDS." It is named classical because it uses only one matrix of distance information or "dissimilarity" and it is metric because the dissimilarity information is quantitative (e.g. distance measurements), as opposed to ordinal. The algorithm has three phases, which are as follows:

1. Collect the shortest paths between all pairs of nodes in the network, or zero if range was not gathered and produce a complete matrix of internode distances D.
2. Apply the classical metric MDS on D to find estimated node positions, retaining the two largest eigenvalues and eigenvectors in order to construct a two dimensional map.
3. Transform the solution into global coordinates using some number of fixed anchor nodes.

The advantage of MDS-MAP is that it can be used either in range-free mode, since it ignores anchor data until the last stage, or in range-based mode, obtaining distance measurements to produce both absolute and relative position.

3.2 Distributed Algorithms

These algorithms extrapolate unknown node positions from beacon positions which know their location in the network. Thus, the algorithms localize nodes in the jurisdiction of a landmark's area. Relevant computation is done on the sensor nodes themselves.

The most representative algorithms in our analysis are: Diffusion or Centroid algorithm, AhLOS algorithms, APS, Bounding Box algorithm, N-hop multilateration, Amorphous Localization and APIT.

3.2.1 Diffusion or Centroid Approach

In [10], Nirupama Bulusu et. al. evaluate the effectiveness of a very simple connectivity-metric method for location in outdoor environments that makes use of the inherent RF communication capabilities of these devices.

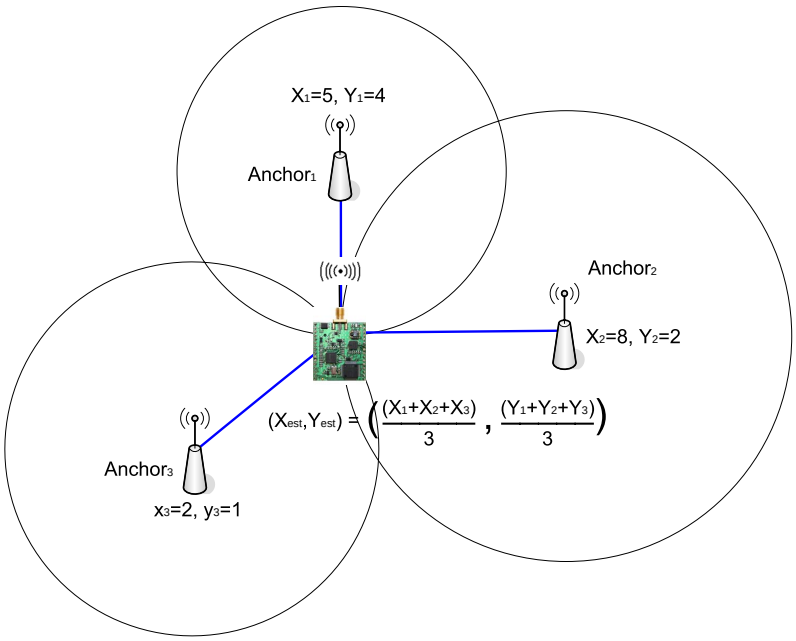


Figure 3.2: Concept of the centroid algorithm with three Anchors

Bulusu et. al. fixed numbers of reference points in the network with overlapping regions of coverage. These reference points transmit periodic signals as beacons or landmarks. The unknown nodes use the simple connectivity metric, which is more robust to environmental incertitudes, to infer proximity to a given subset of these reference points. Nodes locate themselves to the centroid of their proximate reference points by using Equation 3.1:

$$(X_{est}, Y_{est}) = \left(\frac{X_{i1} + ... + X_{iK}}{K}, \frac{Y_{i1} + ... + Y_{iK}}{K} \right) \tag{3.1}$$

where (X_{est}, Y_{est}) is the estimated location of the receiver and K is the total number of anchor nodes. The assumptions that Nirupama Bulusu et al. consider in the project are:

1. Perfect spherical radio propagation.
2. Identical transmission range (power) for all radios.

This scheme is easy to carry out on the nodes but the precision depends on the numbers of anchors that exist in the system and the intrinsic environmental problems due to the real radio propagation. This technique can be improved by making an analysis of the right placement of the system anchors. Figure 3.2 shows the scheme of the algorithm using three landmarks. The main disadvantage of the proposed algorithm is that it is necessary to have a large number of reference nodes to obtain a comprehensive coverage for large-scale networks.

3.2.2 The Bounding Box

The bounding box algorithm [65] proposed by Römer et.al. localizes nodes with low computational complexity. This algorithm creates boxes for each anchor by adding and subtracting the estimated distance from the anchor position. For example, the bounding box of Anchor 1 shown in Figure 3.3 has the following mathematical expression:

$$[X_1 - r_1, Y_1 - r_1] \times [X_1 + r_1, Y_1 + r_1] \tag{3.2}$$

where the position of the anchor is denoted by (X_1, Y_1) and the estimated distance between anchor 1 and the unknown node is r_1. The intersection of each anchor bounding box is determined by taking the maximum of all coordinate minimums and the minimum of all maximums.

$$[\max(X_i - r_i), \max(Y_i - r_i)] \times [\min(X_i + r_i), \min(Y_i + r_i)] \tag{3.3}$$

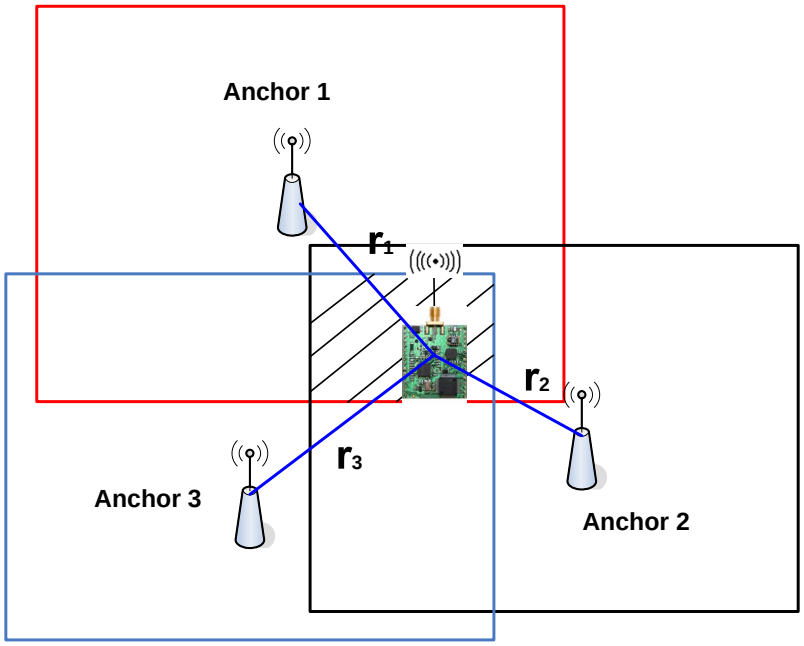


Figure 3.3: Concept of the Bounding Box algorithm

For example, the shadowing area in Figure 3.3 denotes the intersection of three anchor bounding boxes. The estimated position of the unknown node is set to be the center of this intersection:

$$x = \left[\max(X_i - r_i) + \frac{|\min(X_i - r_i) - \max(X_i - r_i)|}{2}\right] \tag{3.4}$$

$$y = \left[\max(Y_i - r_i) + \frac{|\min(Y_i + r_i) - \max(Y_i - r_i)|}{2}\right] \tag{3.5}$$

In the design of Calamary [80], a positioning system that uses TDOA, Whitehouse develops a distributed version of this algorithm. However, he mentions the main drawback of the algorithm is itï£¡s sensitivity to noisy range estimates. The accuracy of the bounding box approach is improved when the nodesï£¡ positions are close to the anchor positions. The most attractive characteristic of this method is that it can be implemented in sensor nodes with very little computational power and low memory capacities.

3.2.3 The Ad-hoc Localization System

The Ad-hoc Localization system [69] provides multiple ways to implement multilateration: atomic, iterative and collaborative. In atomic multilateration (see Figure 3.4 a), the reference nodes' density is high enough for a node with unknown position to estimate its location with basic trilateration. When the node has at least three distances to three reference points (on the bidimensional case), then the node is able to locate itself.

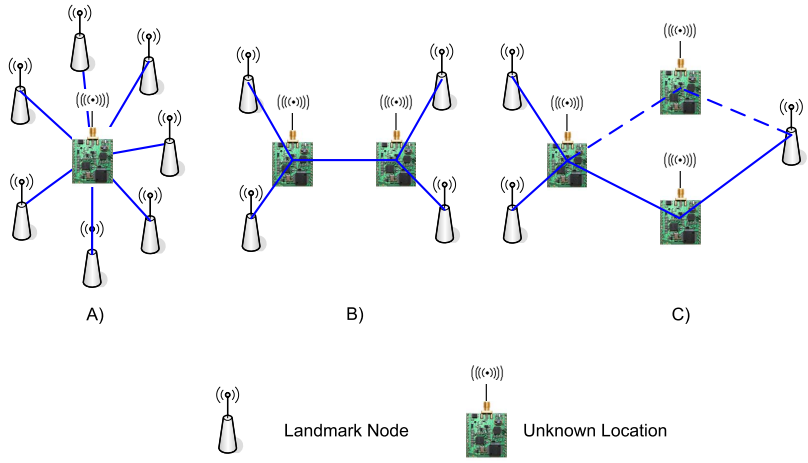


Figure 3.4: Multilateration with the AhLOS algorithm

The iterative multilateration uses other nodes in the system which are not beacons, but have already found their location within the system. Those nodes act as anchors to assist in the positioning of the unknown nodes. The cost of this version, however, is reduced accuracy. But even after applying these two methods there may be nodes in the system which are unable to estimate their position (see Figure 3.4 b). In this case, the problem must be

considered in a collaborative manner because the nodes build and solve a non-linear system using an equation for each edge in the graph.

The numbers of the equation and of unknown nodes are not always good indicators (see Figure 3.4 c). Even if four equations are available to solve four unknowns, node x is not able to resolve the ambiguity. In this case, the node is delayed to compute its position in a later iteration when more proximity information is available. The main contribution of the AhLOS method is to identify problematic situations mentioned above. The main disadvantage of the algorithm is the necessity of a high number of beacons to achieve a high percentage to resolve node positions.

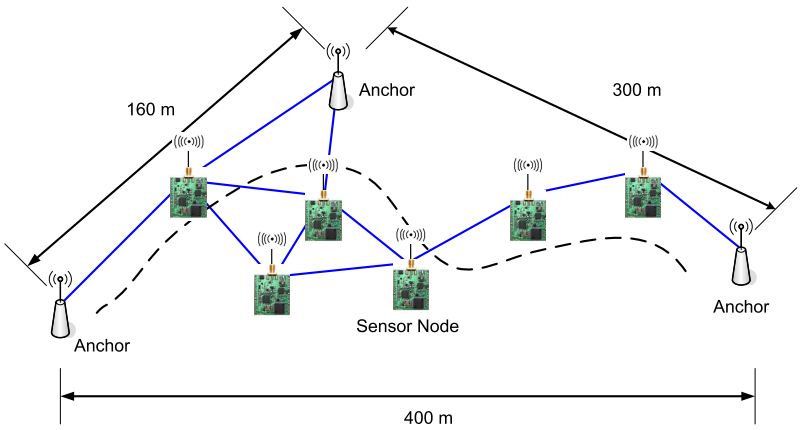


Figure 3.5: Obtainig the average hop by two Anchors with the APS algorithm

3.2.4 Ad-Hoc Positioning System

D. Niculescu and B. Nath in [51] propose a method called Distance Vector Hop (DV Hop). This approach uses a mechanism that is similar to classical distance vector routing. This algorithm computes the distances among anchor nodes by using both, the number of hops between these nodes and the average hop distance. Each receiving node maintains the minimum counter value per anchor of all beacons it receives and ignores those beacons with higher hop-count values. Thus, every node in the network finds the shortest path in hops to the corresponding node reference or landmark. To obtain this distance, the beacon's position is transmitted with hop-count values incremented at every intermediate hop. Through this mechanism, all nodes in the network (including other anchors) get the shortest distance in hops.

In order to convert hop count into physical distance, the system estimates the average distance per hop without range-based techniques. Landmarks perform this task by obtaining location and hop count information for all other anchors inside the network as shown in Figure 3.5.

The average single hop distance is then estimated by anchor i using the following equation:

$$Hopsize_i = \frac{\sum \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum h_j} \tag{3.6}$$

Where $(X_i - X_j)$ and $(Y_i - Y_j)$ are the positions of node i and node j respectively , and h_j is the number of hops between both nodes. Once calculated, anchors propagate the estimated HopSize information to the nearby nodes. Once a node can calculate the distance estimate to more than 3 beacons in the plane, it uses triangulation (or multilateration) to estimate its location. APS is an algorithm designed for ad-hoc networks and it has the advantage that uses solely the radio communication receivers included on the sensor nodes saving the node's scarce power. Another benefit is that has a low communication overhead and it is one of the ad-hoc multihops algorithms with low computational effort.

The APS's drawbacks are that the algorithms does not work for slow and high node densities in the network, the accuracies in the lowest hops produce coarse position estimations due to the poor precision on the node-node distance estimations, lack of locate-sensing accuracy if the network has not more a uniform distribution.

Due to the good properties of this schema, the comparison of our simulation results are contrasted with the efficiency of APS algorithm. In chapter 5 we will compare the performance of this algorithm using real sensor node with different node densities and configurations.

3.2.5 Amorphous Localization

The Amorphous Localization algorithm [48], proposed independently from DV-Hop, uses a similar algorithm to estimate position. First, like DV Hop, each node obtains the hop distance to distributed anchors through beacon propagation. Once anchor estimates are collected, the hop distance estimation is obtained through local averaging. When every node in the network obtains the hop distance of the neighbor nodes, then the node computes an average of all its neighbor values. Half of the radio range is then deducted from this average to compensate for error caused by low resolution.

The Amorphous Localization algorithm takes a different approach from the DV Hop algorithm to estimate the average distance of a single hop. This work assumes that the density of the network is known a priori, so that HopSize can be calculated offline. The expected distance covered per communication hop, d_{hop}, is the physical distance between a pair of sensor nodes divided by the expected number of hops in the shortest communication path. In this paper, they use the Kleinrock and Silvester Formula to obtain d_{hop}, as shown in the following equation:

$$d_{hop} = r\left(1 + e^{n_{Local}} - \int_{-1}^{1} e^{-\frac{n_{Local}}{\pi}\left(\arccos t - t\sqrt{1-t^2}\right)} dt\right) \tag{3.7}$$

where n_{Local} is the number of neighbor nodes and r is the maximum range of their radios. Finally, after obtaining the estimated distances to three anchors, triangulation is used to estimate a node's location. Experimentally, R. Nagpal et. al. show in a recent paper [49] that Equation 3.7 is quite accurate when n_{Local} grows above 5. However, when $n_{Local} > 15$, d_{hop}

approaches r, Equation 3.7 becomes less useful. They also assert that even better than hop counting distance estimates can be computed by averaging distances with neighbours. This benefit does not begin to appear until the number of local nodes rises above 15. However, it can reduce hop count error on 0.2r.

Different from the APS, the Amorphous Localization can estimate better the node-node distances improving the efficiency of the algorithm but at the cost of some requirement such as the requisite to know the density of the network previously and the necessity of a certain number of neighboring nodes to get competitive position estimations. These requirements limit the performance of the approach in some particular node densities and some network distributions.

3.2.6 N-Hops Multilateration Primitive

In [70], A. Savvides et al. develop an attractive algorithm which enables the sensor nodes to perform localization without the requirement of LOS. To produce this goal, the nodes use the location information and distance measurements over multiple hops away from the beacons. The authors consider a node uniquely positionable if it is in direct communication with at least three nodes which have to being positioned in a non-collinear manner. First, the nodes will get at least three landmark-to-node distance estimations using the cumulative node-to-node travelling distances (similar to the APS algorithm) obtained during successive network flooding. Using these distances and the known anchor location, the position of an unknown node is confined inside a bounding box centered at the respective anchor node. Using the intersection of three landmark's bounding boxes, the unknow node position is determined at the center of the bounding boxes. The final node position estimations are obtained by solving a global non-linear optimization problem through an iterative least squares process. The node's locations that were classified as no uniquely positionables are estimated with the help of its uniquely positionable neighbors.

This approach has similarities with the other counting hops methods, but in particular, Savvides et. al. recommend placing some beacons around the edges of the sensor networks field to improve the accuracy. The N-Hops Multilateration Primitive try to solve the lack of reference nodes problem and the line of sight of the beacons through the mutual collaboration of the nodes to estimate their locations. once again, the main disadvantage is that the algorithm need harware specialized on the node to node distance estimations. Another drawbacks of the technique is the requirement of some number of nodes in the network to deliver good postion accuracies.

3.2.7 Approximate Point-In-Triangulation Algorithm

This algorithm requires the separation of the environment into triangular regions between landmark nodes to perform location estimation (Figure 3.6).

In [27], Tian He et. al. mention that a node's presence inside or outside these triangular regions allows a node to narrow down the area in which it can potentially reside.

One of the drawbacks of the technique is the assumption that every node in the network is able to hear a large number of beacons. At the very least the algorithm does not assume that nodes can range to these beacons.

The APIT algorithm, every node decides whether it is inside or outside a given triangle by comparing signal strength measurements with its neighbor nodes. When this algorithm stage is finished, the node locates itself in the center of gravity of the intersection region of the triangles that contain it.

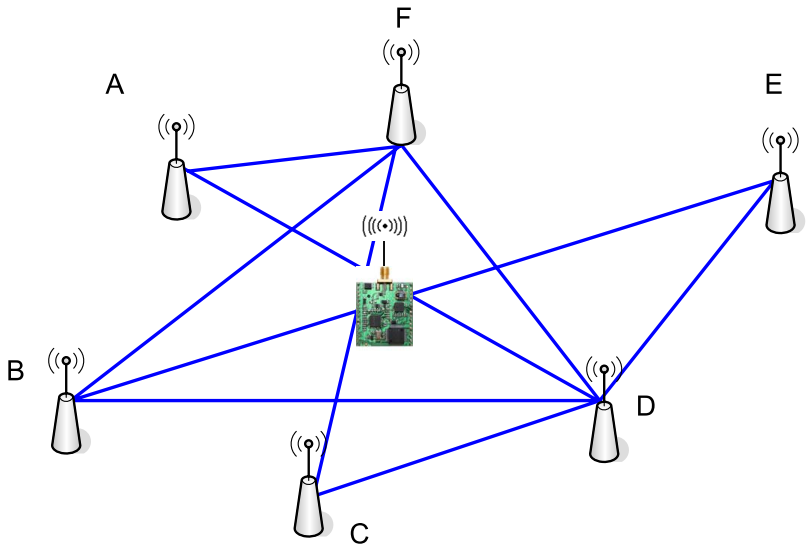


Figure 3.6: Position estimation using overlapping triangles

In Figure 3.6, every triangle corner represents an anchor node and the intersection of all the triangles defines the position of the node trying to locate itself.

The APIT algorithm uses the geometric-based Point-In-Triangulation (PIT) test. For a given triangle with A, B and C corner points, a given point M is outside the triangle ABC, if there exists such a direction that a point adjacent to M is further from or closer to all points A, B and C. On the other hand, if M's shifted position is nearer (further from) at least one anchor A, B or C, then point M is inside the triangle.

Assuming that sensor nodes do not typically move through the network, Tian He et. al. define an Approximate PIT (APIT) test which is based on the relative high node density of these network to emulate the node movement. If no neighbour of M is further from/closer to all three landmarks A, B and C simultaneously, M assumes that it is inside triangle ABC. Otherwise, M assumes it resides outsides this triangle.

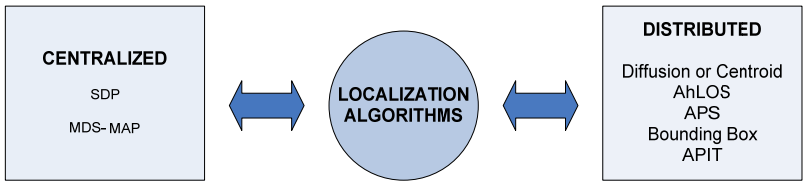


Figure 3.7: Taxonomy of the Localization Algorithms

Table 3.1: Highlights of location techniques

Localization Techniques										
Method	**Distributed System**	**Memory Requirement**	**Communication Overhead**	**Form Factor**	**Accuracy**	**Energy Consumption**	**Computational Effort**	**Scalability**	**Price**	**Observations**
Proximity	++	##	O	##	-	##	##	++	##	Range-Free Can be implemented with different signal based technologies
Lateration	+	O	#	#	+	O	O	+	O	Range-based Combination of electromagnetic waves and acoustic signals
Prior Survey Location	--	**	##	*	++	*	**	-	**	Scalability depend on the number of central engines
Angulation	O	*	#	**	+	**	*	-	*	Accuracy depend on the signal based technology
Distance and Angle Estimations										
Technique	**LOS Requirement**	**Repeatability**	**Synchronization requirement**	**Form Factor**	**Accuracy**	**Energy Consumption**	**Computational Effort**	**Scalability**	**Price**	**Observations**
RSSI	##	--	##	##	-	##	##	++	##	Heavy signal oscillation inclusive without movement
TOA	#	+	**	#	O	O	#	-	*	High clock resolution requirement Relative Robust at obstruction
TDOA	**	++	*	*	+	*	O	-	*	Necessity of two different receptors and senders
AOA	*	+	O	*	++	*	*	--	*	Obstructions correlate poor Angle Estimations
Interferometry	**	+	**	**	++	**	*	--	**	Do not work properly with obstructions

This algorithm is classified as a range-free algorithm because it can use RSSI range measurements which have to be monotonic and calibrated to be comparable, but do not have to be converted into node distances.

The drawbacks of the APIT algorithm are that it requires a relative high ratio of anchors to nodes, big foot print anchor signal, and it is sensitive to errors in RSSI measurements. On the other hand, the low complexity of the algorithm makes improvement possible on typical limited sensor nodes and it requires less communication than other anchor based algorithms.

Different algorithms used in localization are summarized in Figure 3.7. In order to better describe the practical location applications on the next section, some concepts and the most important characteristics of localization topcis, which have been seen in previuos sections, are highlighted on Table 3.1. The symbol used to compare the different techniques have the next interpretations: The symbol "++" for "very good", "+" for "good", "**o**" for "satisfactory", "–" for "sufficient", "– –" for "not sufficient", "∗∗" for "very high", "∗" for "high", "#" for "low" and "##" for "very low".

3.3 Practical Systems

Nowadays there are systems with large scale unattended computers such as automated factories, which contain hundreds of unsupervised sensors which were deployed with very careful planning and react to external events. In WSNs there are applications that work in a similar way, requiring some planning effort to determine where the nodes will be working in order for them to reach the highest grade of localization. On the other hand, there are

also systems which try to develop localization in an ad hoc manner.

Every approach has its advantages and drawbacks, depending on the way it handles the topic. Different location systems which attack a variable point of view of the localization problem will be seen briefly.

3.3.1 Global Positioning System and LORAN System

Perhaps the most well-known location sensing system is the Global Positioning System (GPS) (see [32]). The first experimental Block-I GPS satellite was launched 1978 and was originally intended for military applications but in 1980 the government of the USA made the system available for civilian use.

GPS is based on the NAVSTAR satellite constellation (24 satellites) placed into orbit by the USA Department of Defense. GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's location. Essentially, the GPS receiver compares the signal transmission time by a satellite with the time it was received. The difference in time tells the GPS receiver the distance to the satellite. With distance measurements from three or more satellites, the receiver can determine the user's position and display it on the unit's electronic map.

GPS signals were designed for tracking and locating in open outdoor areas. Therefore, the technology is not reliable indoors and in urban areas. The principal problem of GPS is the lost of the signal due to atmospherical conditions or the multipaths of the weak signal in the cities. Assisted GPS (A-GPS) helps the GPS receiver by providing an alternate source to the fragile navigation message and helping the receiver average for extended periods of time. Another alternative using GPS is the called Differential GPS (DGPS). Here, a fixed ground base station broadcasts the difference between its real position and the estimated position based on the satellite signals, periodically. All the GPS receivers that track the same satellites can use this information as correction when estimating their own position. However, even A-GPS and DGPS are unreliable indoors due to the fundamental physics of GPS satellite signals.

The Long Range Navigation System (LORAN) [24] is a terrestrial navigation system that uses low frequency radio transmitters operating in the low frequency 90 to 110 kHz band. The transmitters use the time interval between radio signals received from three or more stations to determine the position. Actually LORAN operates in a similar way to GPS but uses ground based beacons instead of satellites.

3.3.2 Aeroscout System

AeroScout applications [3] use the standard IEEE 802.11 protocol (Local Wireless Networks or Wi-Fi.) to locate Standard Wi-Fi devices and/or Wi-Fi based Active RFID tags which have a special receptor in the 125 KHz band. These devices send a signal at a regular interval which is received by standard wireless access points or by special location receivers called Aeroscout Location receivers. These receivers work with 2.4 GHz or 125 KHz signals.

The system uses signal strength and/or time of arrival information which is sent to a location

processing server. The engine compares the location data with data previously obtained from the work environment and processes it to show the location on a virtual map. The average accuracy is up to 1 meter rms. Figure 3.8 shows a node and a landmark of the aeroscout system.

(a) (b)

Figure 3.8: Node and Landmark of the aeroscout system (a) Wi-Fi. Based Active RFID (b) Location Receiver. Both pictures have been derived from [1]

3.3.3 Ekahau System

The Ekahau System [40] tracks wireless laptops, PDAs, VoIP phones, Wi-Fi tags and other 802.11 enabled devices. These devices use Wi-Fi. signal as signal base technologie to obtain the position of the unknown devices. The technique used to locate them is the prior site survey. The Ekahau schema includes a special software called Ekahau Position Engine based on signal strength calibration which supports standard 802.11a/b/g Wi-Fi. access points.

The Ekahau Client inside the tag or other Wi-Fi device only collects RSSI values from surrounding access points and sends the data to the server. On the server, the data is processed and compared with the measured RSSI data to determine an estimated location.

For system calibration, the site survey software is installed on a laptop or tablet PC with a floor plan map of the area. The systemï£¡s engineer walks through the facility carrying the laptop and records the RSSI signal strength values in those areas where location information is needed by clicking the map using the mouse or a pointer. Due to these characteristics, the Ekahau project is classified as a centralized system with a Prior Survey Location technique. Once again, the disadvantage of the system is the requirement to do a previous survey of the place where the devices are deployed. The System demand a considerable computational effort, thus the centralized nature of the system design. On the other hand, an advantage is that Ekahau can locate and tracking all devices uses the standard 802.11 protocol. Figure 3.9 shows the nodes used by the Ekahau system.

Figure 3.9: Wi-Fi. Tag T201 of Ekahau System. Picture has been derived from [40]

3.3.4 Rosum System

This system provides indoor and outdoor localization. The structure integrates signals from the existing commercial TV digital broadcast for indoor and urban areas location with the performance of the signals from GPS satellites to support positioning outdoor.

The position calculation according to [61] may be implemented at the enabling device or at the server. They use the signal structure for Digital Television (DTV) specified by the American Television Standard Committee (ATSC). However, it could be developed in NTSC Analog TV Broadcast which in recent years has also included synchronization signals like DVB in Europe and ISDB-T in Japan.

The first step to estimate location consists of extracting timing from the digital (or analog) TV signal through a correlation process where the correlation peak for the field synchronization segment is found. The correlation peak is translated into a timestamp or pseudorange, which represents the arrival time of the field synch at the receiving antenna.

Unlike the case of the satellite-based positioning technique [32], the location of the transmitter is unchanged and does not need to be continually updated. Therefore, the TV transmitter location data may be stored at the Rosum TV Measurement Module (RTMM) inside a given mobile device or at the server.

In order to compute the location of the RTMM, the precise timing of the TV synchronization code transmissions must be known. To develop this task, the system uses Monitor Units. These Monitor Units are small devices that are deployed in a fixed location in the region in which RTMMs (and their host location devices) will be deployed. The Monitor Units use their antenna to monitor TV signals in the region, to analyze the stability and timing of these signals, and to report this information back to the Location Server. Finally the location server computes the position of the RTMM and sends it back to the device. Figure 3.10 depicts the main components used by Rosum TV-GPS. An strong disadvantage of the system is the usage of private license radio bands which makes expensive the localization of devices. The system works either in a centralized mode or in a semi-distributed mode. The scalability and accuracy of the system depend on the number of Monitor Units deployed on the deployment area.

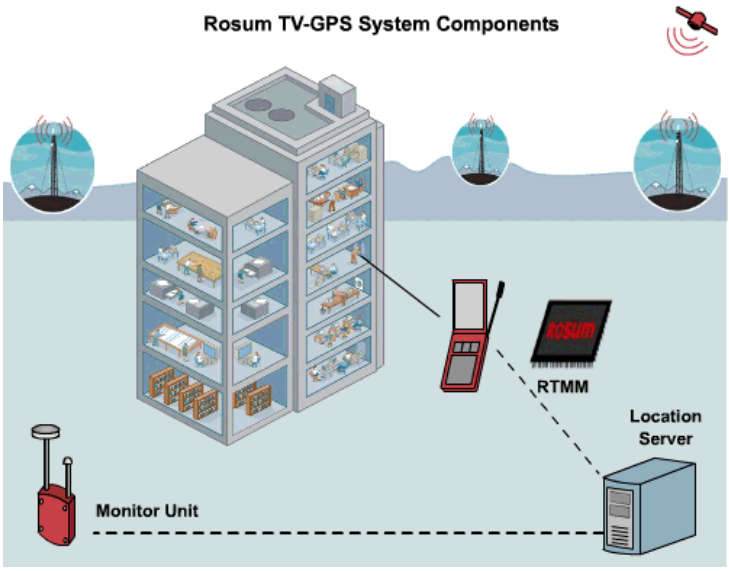


Figure 3.10: Rosum TV-GPS components. Picture has been derived from [16].

3.3.5 Skyhook System

The Skyhook system [82] uses the ubiquitous cloud of radio signals of both public and private Wi-Fi access points networks to estimate location. The server skyhook software is a compilation of a nationwide database of known Wi-Fi access points in the main metropolitan areas of the USA. The service includes location for laptops, tablet portable computers, Personal Digital Assistants (PDA) and smart-phones that have Wi-Fi capability.

To produce location, the WPS location client scans the airwaves for 802.11 signals and obtains some RSSI measurements from Wi-Fi. sites in range. The client finds its position by comparing observed access points against the Skyhook Software database which contains the graphically known locations.

There are two modalities for the Skyhook database; it is either located in a central server or runs on the device. In the case of the device model the database and client are installed on an individual device, the position is calculated on the device and then communicated to other applications such as mapping of the area.

With the network model, the location database resides on the skyhook wireless server which computes the position device and then sends the location back to the client over the network connection or another server based application.

This System also belongs to the Prior Survey Location group. This Wi-Fi. Positioning System (WPS) has the advantage that it requires no specialized hardware and works indoors or outdoors. The high computational effort and the large of data memory on the device, however, are two drawbacks to this system.

3.3.6 BLIP System

The BLIP System [77] is based on Bluetooth signals which operate in a globally available 2.4 GHz radio frequency band. Bluetooth ensures reliable communication within a range of 100 meters if the device is class 1 or 30 meters if it belongs to class 2.

The system is managed by special server software called BlipManager. The BlipManager creates BlipZones, which groups a number of BlipNodes. The BlipNode is a Bluetooth Access Point which controls and monitors end terminals such as mobiles phones and PDAs. The mobile phones and PDAs that are connected to a Blipnode acquire all services that offers a Local Area Network (LAN). The BlipNodes connect to the BlipServer through Internet, Ethernet or Bluetooth as can be seen in Figure 3.11.

The BlipServer is formed by modules, one of these modules being the Positioning Module, which is configured to use graphical user interface. To estimate the position of a Bluetooth device, the server uses the BlipNodes to confirm the presence in the area and register the RSSI, making triangulation possible when more than two BlipNodes detect the wireless device.

Accuracy depends on the distance between BlipNodes, for example, 10 meters between BlipNodes gives approx. 2 meters accuracy. To increase the position accuracy, the grid of BlipNodes is extended with other devices called BlipNode-micros which have wireless connection with the BlipNodes. The main drawback of the system is the necessity to deploy a lot infrastructure on the network area to have an acceptable aware-location or to track a special endterminal.

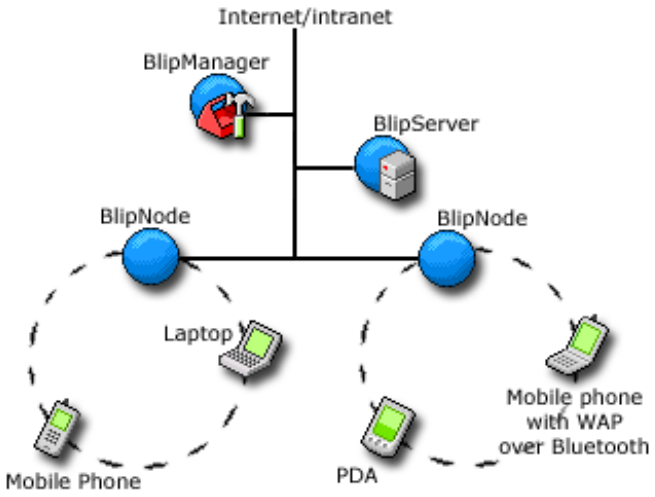


Figure 3.11: Blip system network diagram. Picture has been derived from [78].

3.3.7 Place Lab System

Place Lab [31] is a software which allows clients like notebooks, PDAs, and cell phones to locate themselves by listening for radio beacons such as 802.11 acces points, GSM cell

phones towers, and fixed Bluetooth devices that already exist in large numbers around us in the environment. Place Lab is released under an open source license. Binary and source releases for many platforms, as well as sample radio traces can be found in [55].

This system is similar to the Skyhook system since the clients estimate the position by by identifying the IDs of the beacons and comparing the associated beacon position in a locally cached map.

The three fundamental parts of the Place Lab are: radio beacons in the environment, databases that hold the information of the beacons' position, and the Place Lab client which uses this data to estimate its own position.

An advantage of this locate-sensing system is that it works by listening to the transmission of wireless network sources or radio source beacons (802.11 access points, fixes Bluetooth devices and GSM cell towers). This way, the clients do not need to transmit data to know their own locations, nor listen to other user's data transmission. In [37], Anthony Lamarca et al. obtain a 20-30 meter median accuracy with nearly 100% of coverage.

3.3.8 Right Spot System

John Krumm et al. in [35] present an algorithm to obtain location based on the RSSI measurements of the FM radio stations. The hardware platform for this project was a small low-power device named Smart Personal Object Technology (SPOT). This device was designed to listen for digitally encoded data transmitted on frequency sidebands leased from the commercial FM radio stations.

The standard SPOT device has to measure the signal strength of arbitrary frequencies in the FM band to locate itself. This is one of the attractions of this approach, unlike GPS, because the FM radio signals have wide coverage (indoors and outdoors). The SPOT device obtains a vector of measured signal strength from the different scanned list of FM radio frequencies. Through a correlation of these vectors and a Hash code, the SPOT device estimates its own position.

This project was augmented by Youssef et al. in [87]. Instead of manually training locations as a function of signal strength as before, Youssef et al. used simulated signal strength maps to eliminate the need of physical location visits and to measure signal strengths for training. Using smoothed histograms of rank hash codes, they can infer a device's location down to an accuracy of about 80-50 metres.

3.3.9 Ubisense System

This system develops localization based on UltraWideBand signals (UWB). The network works by creating sensor cells, each of which is typically composed of four to seven fixed sensors to deploy location.

The fixed sensors, called Ubisensors, are devices which work like anchors or landmarks. The Ubisensors are fixed in a known position around the area that has to be covered and are networked using standard Ethernet. Each Ubisensor has an RF transceiver and an array of four UWB transceivers.

The objects whose position has to be discovered are small tags called ubitags. The ubitags have an RF transceiver and an UWB transmitter. When an ubitag is active, it sends out a conventional RF message containing its identity. At the same time, the ubitag transmits a UWB pulse sequence that is used by the Ubisensors to calculate the TDOA and the AOA to estimate the tag's location.

To determine the ubitagï£¡s location in three dimensions, it is necessary to have at least two Ubisensor readings. As mentioned before, the system works in cells which contain one Ubisensor that functions as a master as can be seen in Figure 3.12. This master Ubisensor coordinates the transmission of every ubitag into the cell dividing the conventional RF channel into time slots and then allocating slots to multiple calls.

If the application is focused on tracking a special Ubitag, then it can be scheduled more frequently than the others, conserving the UWB channel and the Ubitag power supply.

In [77], Steggles et al. mention that outdoors, the system can register an accuracy of 15 cm across 95% of the readings without obstacles. Although this system has a good accuracy in the location, the main disadvantage is the need of a lot of expensive infrastructure per area unit.

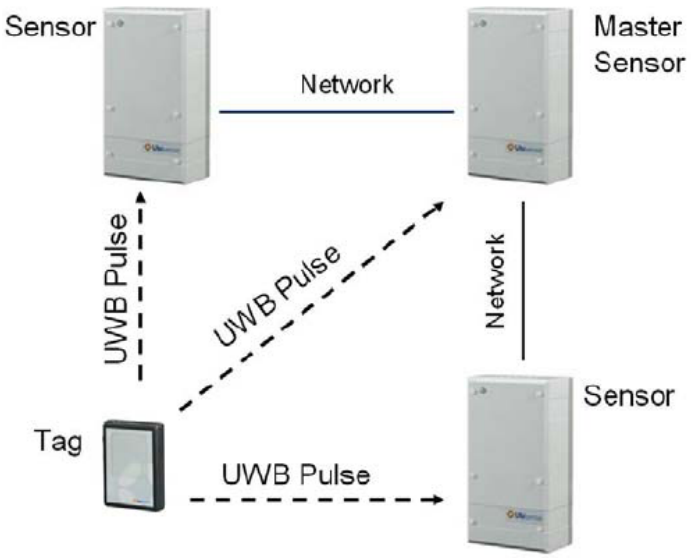


Figure 3.12: Ubisense hardware components. Picture has been derived from [15].

3.3.10 BlueLon System

This system, which shares some similar characteristics with the Blip system, works with devices which are able to use the Bluetooth protocol.

This structure [64] is administered by a server and special software called BlueLon iTooth Acces Control. This software manages the Bluetooth access points that cover the indoor or outdoor area. The server could have a wired internet, or Bluetooth connection with the Bluetooth access points called BlueAccess Bal-100. The Bluetooth access point is capable of

leading with 200 bluetooth devices. To obtain the position of every client, it is necessary to register its ID in the system, and then the server computes the position through the RSSI readings of the different access points by triangulation or multilateration.

For every Bluetooth tag, the battery power lasts approximately 14 days. Once again, the system is centralized and the location engine requires a considerable amount of infrastructure to have an acceptable accuracy. The cost of the hardware is not an active reason to put the system into practice.

3.3.11 WhereNet System

This system locates devices which work based on the standard ANSI 371.1. This protocol is defined at 2.4 GHz and uses spread spectrum technology. This technology specifies nominal location accuracy between two and three meters. The system requires a distance between the tag and the reader of 300 meters outdoors and a distance of 100 meters for indoors.

The WhereNet system [83] works in a centralized form. A server controls and monitors the visibility of the tags in the covered area. The whereTags transmit spread spectrum signals which are recognized by the whereNet beacons. These beacons calculate the position of the WhereTag using differential time of arrival and multilateration.

The battery life of every WhereTag is about 7 years; the time depends also on the frequency of the communication between the beacons and the whereTags. The main drawback of the system is the need to deploy large amount of reference on the usage area, the tags have specialized hardware that increment the tag unit price.

3.3.12 The Radar System

The RADAR system [6] uses the RF signal strength measurements in the 2.4 GHz license-free ISM (Industrial, Scientific and Medical) band.

To estimate the location of the user, the RADAR system takes the RSSI readings from three fixed base stations in two phases. In the first phase, the schema constructs a set of received signal strength maps. To reach this goal it is necessary to collect the RSSI values of the base stations from different points of the area where the system will be used. This process is called the data collection phase or off-line phase.

The second phase is an on-line phase during which the location can be obtained by observing the received signal strength from the user and matching that with the readings from the off-line phase. Through this approach, one can obtain a median accuracy between 2 and 3 meters.

The weaknesses of the system is that the object to be tracked must support also wireless LAN, which makes it impractical for the small and power constraint wireless node devices, the requirement of a a prior survey over the usage area and the relative high amount of data access points on the deployed area to send the signals collected by the user devices to the central engine.

3.3.13 The Cricket System

The Cricket location system [60] was designed to locate devices indoors as well as outdoors. It provides location using ultrasound and RF signals. The Cricket nodes consist of small hardware platforms which have Radio Frequency (RF) transceivers on 433 MHz unlicensed band, a microcontroller, and other associated hardware to generate and receive ultrasonic signals.

Priyantha et al. in [60] mention that the Cricket nodes could be classified into two categories: beacons and listeners. A beacon, as mentioned above, is hardware that acts as a location-aware fixed reference point. These beacons are typically attached to the ceiling and the walls of the work area. Unlike the beacons, the listeners are mobile objects that seek to determine their location.

The beacons periodically transmit a message containing specific information such as the ID of the beacon, its coordinates, the physical space associated with the beacon, etc.

The listeners listen to beacon transmission (the RF and Ultrasound signal) and compute the distance to nearby beacons by Time Difference of Arrival (TDOA). Each listener uses the beacon distances to estimate its position using multilateration. If the listener has multiple ultrasonic sensors, it is able to estimate its orientation.

The accuracy obtained with this system was in a region of 30x30 centimeters, a distance of 5 centimeters in free space is even reported in [59].

The biggest weakness of the system is the use of ultrasound signals because they are very sensitive to obstacles. It also has a very limited scope (approx. 5 meters) and high precision distances can only be achieved through a direct LOS as well as fine alignment.

3.3.14 The Bat System

In the Bat system, Harter et al. in [26] develop a matrix of receiver elements which are placed on the ceiling of the building similar to the Cricket system, but the receivers are connected together by a serial wire cable network to form the receiver matrix.

The receivers have an ultrasonic receptor and a transceiver. The transmitters are small units called bats. These bats are attached to equipment or are carried by personnel.

Bats consist of a radio transceiver, controlling logic, and an ultrasonic transducer. Each bat has a unique ID associated with it. The network is controlled by a central computer which periodically sends a radio message with a single ID, causing the corresponding bat to emit a short uncoded pulse of ultrasound. The network receivers detect this ultrasound pulse and send out the information to the central computer which does all the data analysis for tracking the transmitters.

The central computer uses the time of arrival of the ultrasound signal, which it records from the different receivers which listen to the signal. Then, this time is converted into the corresponding bat-receiver distances and the position of the bat can then be deduced through multilateration. The system use small devices which reduce the infrastucture cost, but the cable planning and the centralization of data render the system inflexible and not appropriate for different scenarios.

3.3.15 The Pinpoint 3D-ID System

This system [13] uses Radio Frequency Identification (RFID). The readers in this system extract the tag ID and determine the tag's distance from the antenna by measuring the time of flight round-trip. Since the reader generates the signal, there is no need to calibrate the tag's clock.

The distance of each reader is determined independently. Consequently, there is no need to synchronize the clocks on the various readers.

The system is organized into cells within a building. Each cell is handled by a cell controller, which is attached to 16 antennas by means of coaxial cables.

To estimate location, the tag signal is received at a time that is the sum of the following: transmission time by the cell controller and the transmission of data to the central engine. The central machine leads with the known fixed delays in the system and computes the distance tag multiplying the time of flight through the air and the speed of light. Since the signal travels to and from the tag, the distance to the tag is half of the air distance traveled.

On the one hand, a limitation of this system is that it requires its own indoor antenna infrastructure. On the other hand, it offers a relative ad-hoc deployment of tags. The approach acquires distance measurements between 1 and 3 meters of accuracy. The wiring nature of the central machine, therefore, makes the system less attractive.

3.3.16 The Calamary Project

The Calamary project [80] works with ultrasound signals as well as the Bat and the Cricket systems. In this project, the acoustic time of flight is calculated by transmitting a radio message of 433 MHz radio and ultrasonic pulse of 25 KHz simultaneously.

The nodes measure the time difference of arrival of the sent signals and find their own position on the network when they collect three or more measures from the reference nodes.

The disadvantage of the system is that receivers and anchors have to be synchronized. After the synchronized phase, the nodes in the network are able to estimate their location. Whithouse et al. report accurate measurements within approximately 10 cm of error when the acoustic transducers are pointed at each other.

A special contribution put forth in this paper is the usage of a reflective cone over the ultrasonic transducer (see Figure 3.13) which minimizes the requirement of having an strict alignment between nodes to achieve more precise distance measurements. Althought the nodes are cheap and ad-hoc devices, the scalability of the system is limited by the maximum range of the ultrasonic signal (about 5 meters).

3.3.17 The Pushpin System

This system also belongs to the ultrasonic group as the Calamary project does. Broxton et al. in [9] also use ultrasound signals for this project. But instead of using RF signals to coordinate the time difference of arrive protocol, they use Infrared transceivers in every node of the network. Most importantly, the paper proposes the estimation of the node's location

(a) (b)

Figure 3.13: The Calamary nodes (a) Assembled reflective cone on nodes (b) Calamary node and the reflective cone from [81]

using a special global external stimulus. This external stimulus is an external device which generates an acoustic and IR pulse which produces the beacons of the system.

The reference nodes on the network are not special nodes; however, every node in the network could be a landmark node. The process to elect a node to be a reference depends on the three external stimuli proximity for bidimensional localization (or four in the case of three-dimensional location-sensing). In other words, the external stimulus finds the seeds of the system to develop an ad-hoc localization system with relative coordinates. Once the beacons are chosen through the network, they help the other nodes to find their location by multilateration.

In contrast to the other applications, the Pushpin system works as a distributed network with which localization is achieved in an ad-hoc manner with simple sensor nodes.

The fact that the external stimulus has to be activated in a remote central engine and also that the system uses ultrasonic plus infrared light makes for some strong drawbacks. These two signals technologies have the disadvantage of operating solely in direct LOS. The smallest obstacle to the direct sighting between nodes would misplace the corresponding nodes.

3.3.18 The SpotON System

J.Hightower et al. in [30] document the creation of the SpotON system. This system tracks people indoors. It uses tagging technology for three dimensional location based on radio signal strength analysis. The system uses 916.5 MHz for communication and works in a centralized manner.

In order to locate the node, it is necessary to do an offline previous place survey taking RSSI measurements. These measurements are stored in a remote central computer. The Spot ON System computes the position of the tags by multilateration. Altough the use of

the radio signal allows the location of tags without direct LOS, the previous analysis on the workplace makes it a system not appropriated for wireless sensor networks.

The battery time life is approximately of 10 hours and the total hardware component cost is about $30-$40.

3.3.19 Easy Living System

The Easy Living project at Microsoft Research [74] was developed for building intelligent environments. An intelligent environment is a place, where people live and work that contains computing devices which work together with the common goal of providing information and services to mobile users.

Here the system attempts to manage an emerging area with stereo vision technology. The vision group at Microsoft Research uses the computer vision to locate objects. The localization of objects could be very precise but at the cost of high computational effort, prior survey of the working place, and a considerable amount of infrastructure installation. In other words, the only way to achieve localization with this technique is through a centralized network.

3.3.20 E911

Maybe the main motivator to implement localization was the 1996 Federal Communications Commission (FCC) second-phase mandate for E911 services [14]. This mandate was the product of a new industry of location-aware applications, formally named location-based services (LBS) .

The E911 is not a specific location-sensing system but is important due to the fact that it was the first try for localization. To meet the FCC requirements for positioning, it must be accurate within 150 meters for 95 percent of calls with receiver-based handset solutions such as GPS, or to within 300 meters with network-transmitter-based approaches.

3.3.21 Motion Wireless

This enterprise [47] offers position and tracking to support virtual reality and motion capture for computer animation. This tracking system generates axial DC magnetic-field pulses from a transmitted antenna in a fixed location.

The central engine monitors and controls the system sensors. This engine computes the position and orientation of the receiving antennas by measuring the response in three orthogonal axes to the transmitted field pulse. This pulse is the linear combination of the constant effect of the earth's magnetic field and the transmitted signal. The central computer has to separate this effect to obtain the feasible measurement.

With this system, it is possible to achieve very high precision and accuracy in localization (on the order of 1 mm spatial resolution) and also high responsiveness, but with a high price tag.

The range of operation of the system is in an area of 3x4.5 m. Another strong drawback is the extreme sensitivity to metallic objects. Other example that uses a similar technique

in the virtual area is [8] the usage of a CDMA radio which enables the tracking of body sensors.

In Table 3.2 we can compare each location-sensing system that we have discussed in this section in a quantitative form with the parameters previously seen on the last sections.

3.4 Algorithms which work with Node Density

As we saw in the last sections, there are many commercial applications and a considerable amount of algorithm proposals to sensing location. Most of the practical applications, however, have the strong disadvantage of offering the location service at a very high cost. Either the infrastructure and special devices on the nodes and tags are necessary or the computer complexity is so hard that the system can not be flexible or ad-hoc, but strongly centralized.

We set out in this thesis to design a distributed ad-hoc system for WSNs solely using the radio communication, the RSSI measurements, and the knowledge of local network density. In the development of this system, we took into account the scarce resources of single and cheap nodes such as limitations of memory, energy, and processing capacity.

Most of the localization algorithms designed in recent years are based on multilateration without directly confronting the poor ranging measurement of dynamic and noisy environments. This section presents the state of algorithms, including an analysis of the density of nodes in wireless networks. We focus on the most significant papers derived from the analysis of the radio signal strength indicators, connectivity, and prototypes which do not depend on global infrastructure.

A plethora of algorithms have been proposed to estimate distances between sensor nodes. For example, Bulusu et al. [11] use a node density classification to find the best location for beacon placement in a network. Y. Ohta et al. [54], present a data collection technique in dense networks. They relate local node densities to the number of nodes necessary to obtain a given accuracy for distance estimation to a moving target.

The more interesting approaches are those which use node densities to approximate the distance between nodes. In [84], Wong et al. analyze neighborhood node densities to improve algorithms which rely on hop-counting to estimate positions. They classify node densities into three categories: low, medium, and high densities. The motivation behind this classification is that the less dense a network is, the less optimal a path between two nodes becomes. Varying densities are a characteristic problem in non-uniformly distributed networks. The authors utilize these categories to produce range ratios which represent the ratio of expected hop-distance associated with the transmission range for a particular local density.

The follow-up paper [85] discusses the minimization of the cumulative error in distance-hop estimations for long hop-count propagation paths. They use the local density per node to weight every hop accordingly and compare the results with DV-Hop [52]. While the approaches above rely on three discrete categories of densities as parameters to weigh hop distances, DIN allows for a continuous mapping, thus resulting in a better resolution of distance estimation.

Table 3.2: Applications in Location-Aware Systems

Application name	Classification parameters								
	Work frequency	Battery time life	Range		Location Algorithm	System	Accuracy and precision	Cost	Observations
			Indoor	Outdoor					
GPS Pocket size	L band Frecuencies 1575.42 MHz and 1227.60 MHz	≈2 days	No reliable	Earth	Multilateration TOA	Decentralized	15 m (95 to 99%)	Expensive Infrastructure $100 receivers	Not Indoor and in urban areas. Vulnerable to weather conditions, multipath, line of sight.
Aeroscout	Wi.Fi. 2.4 GHz and 125 Khz	≈4 years	60 m	200 m	Multilateration TDOA RSSI	Centralized	5 to 10 mts (≈100%)	≈$55 each RFID. Software and reader ≈$3000	Previous map RSSI planning. Use Location receiver and Wi.Fi
Ekahau	Wi.Fi. 2.4 GHz	≈5 years	40 to 60 m	100 to 150 m	Multilateration RSSI Analysis	Centralized	3 to 5 mts	$100 Wi.Fi. Tags. Administration Cost	Prior RSSI site survey Building and periphery
Rosum	300–750 MHz	Depend of the host device	Building	25 to 75 Km	Multilateration Pseudoranges analysis	Centralized	-----	Administration cost, relative cheap tags	Previous signal survey outside and inside
Skyhook	Wi.Fi. 2.4 GHz	Depend of device	Covered for the Wi. Fi. Network	City	Multilateration RSSI analysis	Centralized	Aprox. 20 mts outdoor	Administration cost	No Hardware. Signals private and public
BLIP	Bluetooth 2.4 GHz	Depend of the host device	100 m	120 m	Presence and RSSI	Centralized	Approx 2m (90%)	Administration Cost	Make a Personal Area Network, use presence/RSSI
Place Lab	Wi.Fi. GSM cell phone towers Fixed Bluetooth devices	Depend of the host device	Wide Network	City	RSSI analysis in Database	Decentralized	Approx. 20-30 m (≈100%)	Administration Cost	The software compare the IDs of the beacons to estimate location
RightSPOT	FM radio signals	Between 2–4 days	-----	City	RSSI analysis compare to simulated RSSI maps	Decentralized	Approx. 8 Km (81%)	$70–$75 per Unit	No constitute a bilateral network
RADAR	2.4 GHz radio signals	Several months	200 m	-----	RSSI analysis triangulation	Centralized	2–4 m (50%)	≈$100	Considerable Preplanning effort
Cricket	433 MHz 40 KHz Ultrasonic wave	5–6 weeks	≈5 m due Ultrasound	-----	TDOA analysis Multilateration	Decentralized	0.30 m (≈100%)	≈$10	Beacons per room. Receiver computation
BAT	40 KHz ultrasonic wave	Several months	≈5 m due Ultrasound	-----	TOA Multilateration	Centralized	10 cm (95%)	Administration Cost, cheap tags and bases	Wait to loss reverberation in every tag location
PinPoint 3D-ID System	RF 40 MHz 802.11	Several months	Building	-----	Lateration	Centralized	1–3 m	Several bases stations, Infrastructure, Installation expensive	Interference 802.11 Installation planning
Calamary Project	RF 433 MHz 25 KHz ultrasonicave	Several months	≈5 m due Ultrasound	-----	Lateration TDOA	Decentralized	10 cm (95%)	Cheap Nodes low infrastructure	Line of sight necessary, temperature dependence, relative indoor environment
Pushpin System	IR transceivers 40 KHz Ultrasound	N / D	1–2 m	-----	Lateration	Decentralized	5 cm (≈ 90%)	Cheap Nodes external exciter is needed	Relative position, external stimulus needed
Ubisense System	UWB 928 MHz, 433 MHz, 870 MHz	12 months	60 m	250 m	Multilateration TDOA and AOA	Centralized	15 cm (95%) outdoors	Relative cheaps tags, expensive Infrastructure	Robust against multipath, scatter and interference
BlueLon System	Bluetooth	14 days	50 m	100 m	Triangulation presence	Centralized	2–3 m	Administration Cost. ≈$100 pro tag	Expensive Infrastructure limited scalability discoverable after 3 min
WhereNet System	ANSI 371.1 2.4 GHz	7 years	100 m	300 m	Multilateration RSSI	Centralized	2–3 m	Administration Cost ≈$100 Wi.Fi. Tags	Robust against interference and multipath. No Interference with Wi.Fi.
SpotON System	RF 916.5 MHz	10 Hours	Depend on the cluster size	-----	Multilateration RSSI Analysis	Centralized	2–3 m (95%)	Administration Cost $30–$40 Hardware Component	Cluster Administration
Easy living	Computational Vision	N / D	Single Room	N / D	Triangulation Pixel Analysis	Centralized	Variable	Cheap cameras Expensive infrastructure	High Computational effort
E911	Cellular System	Depend of the device	Building, office	City	Triangulation Presence	Centralized	150–300 m (95%)	Expensive Infrastructure. Cost-Unit depend of the cellular phone	Available only where cell coverage exist Presence
Motion Start	Magnetic Field Pulse	N / D	3 m	-----	Multilateration	Centralized	Milimeters (100%)	Expensive Infrastructure	High Computational effort, good accuracy but high cost

N / D Information not available
----- The system was not designed for this purpose

The publication having the most similarity with our work is described in [12]. Here, Buschmann et al. present an algorithm called NIDES to infer distances between nodes. The algorithm functions based on the observation that distant nodes have fewer neighbors in common than close ones. They first use a mathematical model to derive the overlap of two unit disc graphs representing the transmission range, and relate this intersection area to a number of shared neighbors. Depending on this number and different radio models, a look up table to determine the distance between two nodes can be constructed.

The exploitation of the received signal strength for distance estimation has been examined from several points of view. Lymberopoulos et. al in [42] carried out extensive measurements in both obstacle-free and indoor environments of signal strength properties for a 2.4 GHz CC2420 radio to provide a detailed analysis of parameters influencing the RSSI value. Conclusions on how to overcome intrinsic problems are left undiscussed and open for future research. Besides proving that RSSI values are closely correlated to environmental parameters, Zhao et. al [88] explore the feasibility of concluding the distance of a transmitting Berkeley Mica Mote operating at 433 MHz from the reception rate of packets. They aim to set up a merit figure based on RSSI to describe and quantify the corresponding reception rate. Furthermore, research in the area of range-free positioning and localization methods include studies on optimizing antennas to provide less fluctuating signals [86] or the development of more sophisticated algorithms relying on ordering and ranking sequences of measurements to predefined reference points to identify unique regions within the localization space. Although the distance estimation method based on RSSI values that we utilized for our comparison is rather simplistic, we based the chosen algorithm on intense calibrations. With regard to the instability of RSSI values identified, we tried to provide the fairest comparison possible to DIN.

Table 3.3 displays a general comparison between range-free algorithms and useful parameters to engage in a state-of-the-art trial. We are interested in truly distributed algorithms that can be employed on large-scale ad-hoc sensor networks. These algorithmns have to be robust which means they need to be tolerant of node failures and range errors, energy efficiency with little computation effort, and self-organazing (i.e., independent of external infrastructure). We try to evaluate different algorithms in a qualitative manner estimating the advantage and drawbacks of algorithms and prototypes with the following pattern of symbols: The symbol "++" for "very good", "+" for "good", "**o**" for "satisfactory", "–" for "sufficient", "– –" for "not sufficient", "**" for "very high", "*" for "high", "#" for "low", "##" for "very low" and "NA" for "not applicable".

3.5 Conclusion

The area of localization in ad hoc wireless sensor networks constitutes an emerging research field with a variety of open topics to be addressed. In this chapter, I described different algorithms and practical location-sensing systems. The parameter explained in section 2.1 help us to compare the weaknesses and advantages of the several algorithms and location-sensing systems in a qualitative manner.

In reference to the different localization systems, we identified the relationship between the signal based technologies used and the scalability of the whole network, as well as the energy

Table 3.3: Comparison table for different localization algorithms and prototypes in WSN

Parameters	Centroid	Bounding Box	Ad-hoc Localization System	Ad-hoc Positioning System	Easy Living System	N-Hope Multilateration	Approximate Point-in-Triangulation	Pushpin System	NIDES	Cricket System
Scalability	--	+	+	++	-	+	-	--	++	-
Computational Effort	##	##	#	#	**	O	##	O	#	#
Self-Organizing	O	O	+	++	--	+	O	-	+	+
Distance Estimation	NA	O	+	+	++	+	NA	+	+	+
Anchor fraction	**	*	O	O	NA	O	*	*	NA	O
Energy Efficient	++	+	O	+	--	-	+	O	O	-
Real Hardware Implementation	+	NA	+	NA	++	+	NA	++	NA	++
Cost Implementation	##	NA	O	NA	**	O	NA	*	NA	*
Position Accuracy	-	O	+	+	++	+	+	+	NA	+
Refinement	NA	NA	+	NA	NA	+	NA	NA	NA	+

consumption on the devices. For example, some signal based technologies like the acoustic signals in combination with electromagnetic waves present a very good position accuracy, but at the cost of a limited operation range which is later translated on an increment of the number of beacons to cover a determined area.

The commercial locate-sensing systems based on Wi-Fi signals partially solve the usage of specialized hardware to locate wireless devices, but with the inconvenience of a centralized work mode or the employ of relativly large amount of memory making it inappropiate for WSNs.

Focusing on the several localization algorithms of the last section, we found the centralized algorithms have a good performance and can be implemented using different distance or angle measurement techniques. The variety of measurement techniques can improve the position accuracy of the system due to better quality on the distance or angle estimations, but the nature of the algorithms do not allow to program the algorithms on a single node.

Another disadvantage of the centralized algorithms is that the scalability of the system is proportional to the number of central engines on the deployed area. In contrast, the computational complexity for distributed algorithms generally increases with the number of deployed sensor nodes in the network following a exponential relationship [19, 4, 5].

We can observe in table 3.3 the more complex the algorithm is, the less energy demanded of the unknown device. Some of the distributed algorithms have a refinement phase where the position estimation can be improved, but at the cost of a extra communication overhead. In general, the distribuited algorithms are more appropiated for WSNs than centralized ones due to their low complexity, high scalability value, and better self-organization of the system.

An interesting point discovered in our survey was that the majority of the scientific studies do not provide information regarding bias and variance metrics to compare their results with other algorithms using real hardware. Few studies such as [62] have been validated their algorithms through implementation in an experimental testbed. We realize that real-scenario performance results, such as pracitcal comparison of several algorithms, have only begun to be reported. In particular, we have identified two research topics as key future challenges requiring further scientific attention: first is the development of a common framework for analytical characterization and effective performance comparison of localization algorithms

for wireless sensor networks; second is an experimental evaluation of localization schemes via real-scenario testbed deployment. In the next chapters of this thesis we will try to compare our innovative algorithms with different distributed algorithms in simulation and using real hardware.

CHAPTER 4

Distance by Intersection of Neighborhoods

WSNs [2] store and partially process the sensed data either within the same sensor nodes which take the local samples or transmit the sensed data to a remote central computer where the data will receive a bigger and more complex handling process.

In order to have a record of the place of study, it is very important to correlate the collected measurements by the nodes to a specific location. Furthermore, the position of the nodes opens up new ways to detect special events, track an object of interest, and improve the network coordination by executing geographic routing algorithms. The location problem is especially crucial in WSNs, because it is necessary to find methods that work in an ad-hoc fashion and without additional specialized hardware to save scarce resources due to the fact that indoor positioning is not possible with GPS.

The first step into this direction is the estimation of the distance between nodes. To obtain this information, there is a variety of techniques that exploit physical phenomena such as the time of arrival of sound signals [76], the time difference of arrival between radio and ultrasonic signals [59, 9], the use of interferometry [43], RSSI [40], or the use of camera pictures with a previous scene analysis [46].

This thesis focus on the problem of non GPS based, ad-hoc and low cost localization for WSNs. It proposes a method to estimate distances based only on the analysis of local node densities called Distance by Intersection of Neighborhoods (DIN)[41]. This algorithm estimates distances between nodes which share a communication link using the number of nodes that are positioned in the union and intersection area of their communication ranges. We evaluated our algorithm for indoor usage using simulations and real hardware experiments.

First at all, we implemented an RSSI-based distance estimation experiment to obtain an idea of its performance with real hardware using the Scatterweb nodes described in the next subsection. The main purpose is to know the quality of this range free technique for indoor environments using WSNs.

4.1 The DIN Algorithm

The DIN algorithmï£¡s main objective is to determine the distance between nodes in ad-hoc manner, relying solely on the investigation of local node densities. In this section, it will be described this algorithm as well as the precursor algorithm to DIN to get a better idea of the functionality of the contribution.

4.1.1 Weighted Density Node Intersection

The first approach to the DIN algorithm was the Weighted Density Node Intersection (WDNI) [21]. This algorithm determines distances between adjacent nodes using knowledge about local node densities. The first step required finding a mathematical expression of the distance between two nodes in terms of the intersection area of their communication ranges. This distance requires taking into consideration two parameters, the first one being an approximation of the distance obtained by evaluating the different sized, uniformly distributed networks. Second, one much use a function which relates this distance to the number of nodes in the union of their communication ranges. The mathematical foundations are described in Appendix B.

4.1.2 The DIN algorithm as the improved version of WDNI

In this section, the DIN algorithm as a proposal to increment the accuracy and flexibility of the range-free techniques such as the RSSï£¡-based distance estimation is introduced. The essence of the approach is to determine distances between nodes through the analysis of the local density which the nodes find between each other in an ad-hoc manner.

Unlike to the WDNI algorithm, the DIN algorithm does not need the help of weighted functions previously evaluated by means of different network simulations. To obtain a distance from the local density surveys, the DIN algorithm relates the distance between two nodes in terms of the union and intersection areas of their communication radii. The mathematical model is verified using the help of the `ns-2` simulator with different nodes distribution. The DNI algorithm as the improvedment of WDNI is also tested with real sensor nodes in uniformly and near-uniformly distributed networks.

4.1.3 Relating Distance to Radio Intersection and Union Area

An important consideration in our mathematical foundations is that we based the DIN algorithm on an idealized two dimensional radio model. Although we are aware that assumption is not valid in reality, as we have shown in section 1.2, we use it because it was simple and easy to reason out mathematically. We take into consideration the next three main assumptions:

1. Unit disc graph radio transmission range
2. Identical transmission ranges for all the nodes in the network
3. Uniform distribution of nodes in the network

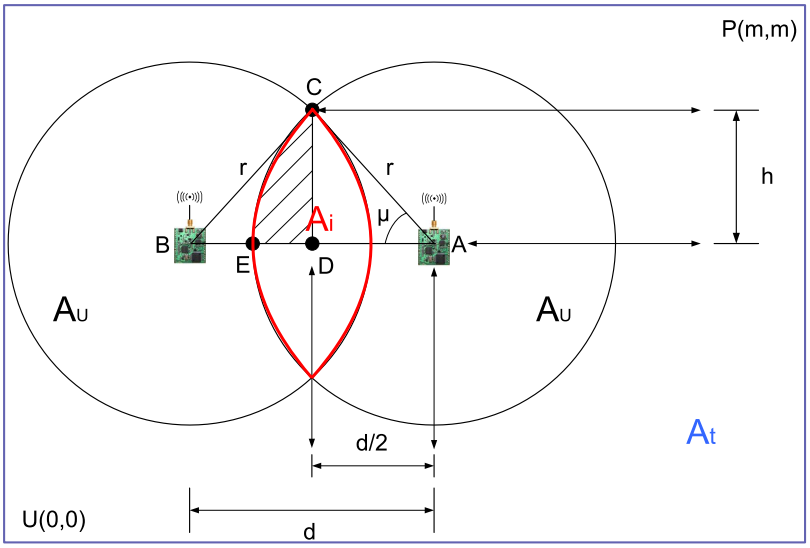


Figure 4.1: Geometric analysis of the intersection and union area of overlapping transmission ranges

If we consider a bidimensional square with diagonal coordinates $U(0,0)$ and $P(m,m)$, where the nodes could be deployed as shown in Figure 4.1, we can find a point x which represents a sensor node, into the area as $x \in [U(0,0), P(m,m)]$.

When we take into consideration two neighboring nodes which share a radio link communication (as it is marked in red in Figure 4.1), we can obtain the intersection area A_i of the overlapping transmission ranges, which are despicted as circles by geometrical analysis. As we can see in the same figure, we define the union area with (A_u) and the complete deployed area as A_t.

We denote the probability of finding x in the union and intersection area in terms of the named superficial area as:

$$P(x \in A_i) = \frac{A_i}{A_t} \tag{4.1}$$

$$P(x \in A_u) = \frac{A_u}{A_t} \tag{4.2}$$

The conditional probability to find x in the intersection area under the condition that x is already in the union area is denoted as:

$$P(x) = P(x \in A_i | x \in A_u) = \frac{\frac{A_i}{A_t}}{\frac{A_u}{A_t}} = \frac{A_i}{A_u} \tag{4.3}$$

To find the intersection area between the two circles, we rely first on the mathematical development to estimate the fourth of A_i (shadowed area in Figure 4.1)

$$A(\overline{CDE}) = A(\overline{CEA}) - A(\overline{CDA}) = \frac{\alpha}{2} \cdot r^2 - \frac{h \cdot \frac{d}{2}}{2} = r^2 \left[\frac{\alpha}{2} - \frac{d_n}{8} \sqrt{4 - d_n^2} \right] \tag{4.4}$$

where d_n is the distance normalized by the radio transmission range ($d_n = \frac{d}{r}$). Finally, we take four times the equation 4.4 to complete the formula

$$A_i = 4A(\overline{CDE}) = r^2 \left[2\cos^{-1}\left(\frac{d_n}{2}\right) - \frac{d_n}{2}\sqrt{4 - d_n^2} \right] \tag{4.5}$$

Combining 4.1, 4.2 and 4.3 with 4.5, one obtains

$$P(x \in A_i) = \frac{A_i}{A_t} = \frac{r^2 \left[2\cos^{-1}\left(\frac{d_n}{2}\right) - \frac{d_n}{2}\sqrt{4 - d_n^2} \right]}{m^2} \tag{4.6}$$

$$P(x \in A_u) = \frac{A_u}{A_t} = \frac{r^2 \left[2\pi - 2\cos^{-1}\left(\frac{d_n}{2}\right) + \frac{d_n}{2}\sqrt{4 - d_n^2} \right]}{m^2} \tag{4.7}$$

$$P(x \in A_i | x \in A_u) = \frac{A_i}{2\pi r^2 - A_i} = \frac{4\cos^{-1}(\frac{d_n}{2}) - d_n\sqrt{4 - d_n^2}}{4\pi - 4\cos^{-1}(\frac{d_n}{2}) + d_n\sqrt{4 - d_n^2}} \tag{4.8}$$

4.1.4 Relating Distance to the Local Node Density

Due to the restricted resources available on real sensor nodes, it is not suitable to design an algorithm that requires in-situ complex mathematical computations. In turn, the idea here is to make this problem a more light-weight one by demanding the sensor nodes to conclude their distances from the number of their neighboring nodes. Under the assumption of uniformly distributed networks, the number of nodes in the intersection area A_i is proportional to this area. We refer to this quantity of nodes as K_i.

Although using the equation 4.6 and 4.7, we can obtain the information of the distance between adjacent nodes, we believe that this estimation can be further minimized taking into account the local node densities of participating nodes.

The approximation can be smoothed out by weighing it with the number of nodes that are in the union of the transmission ranges A_u. We denote as K_u the number of nodes in the area A_u. When the number of nodes in the intersection area A_i and in the union area A_u are proportional, we complete the mathematical steps of DIN defining the function $H(d_n)$ as following:

$$H(d_n) = \frac{A_i}{A_u} = \frac{4\cos^{-1}(\frac{d_n}{2}) - d_n\sqrt{4 - d_n^2}}{4\pi - 4\cos^{-1}(\frac{d_n}{2}) + d_n\sqrt{4 - d_n^2}} \approx \frac{K_i}{K_u} \tag{4.9}$$

Since we are interested in finding an expression for the distance between two neighbor nodes, we have to solve equation 4.9 for d_n. Using MatLab, we obtained a polynomial approximation of degree three to determine the normalized distance between nodes.

$$d_n \approx \begin{cases} -2.73H_n^3 + 5.66H_n^2 - 4.88H_n + 1.88 & k_i \neq k_i \\ \frac{1}{k_i - 1} & k_i = k_i \end{cases} \tag{4.10}$$

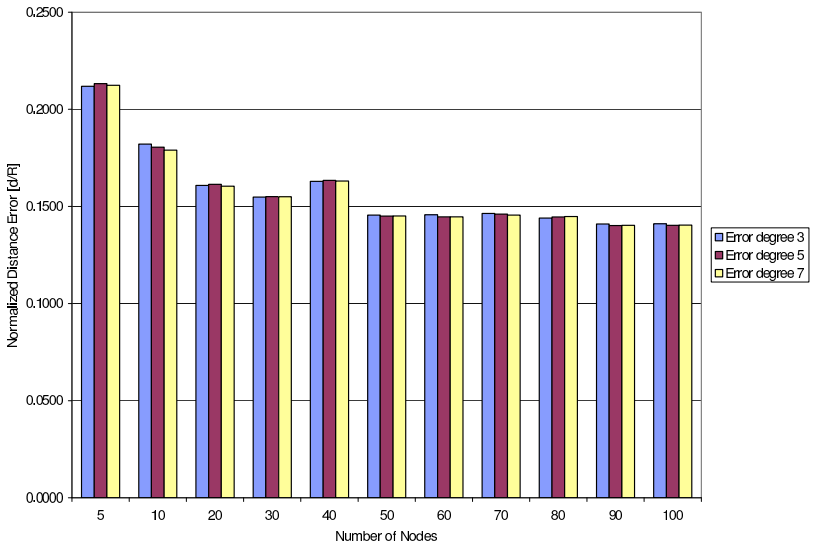


Figure 4.2: Comparison of error resulting from different degrees of the approximation polynomials obtained with ns-2.

Equation 4.10 is limited by $H(d_n)$ values between 1 and $\frac{4\cos^{-1}(\frac{1}{2})-\sqrt{3}}{4\pi-4\cos^{-1}(\frac{1}{2})+\sqrt{3}}$. Those values assure a shared link communication between two adjacent nodes. With this mathematical expression of 4.10, we finish the analysis to find the distance between adjacent nodes. Now every node in the network can estimate its distance between neighbouring nodes in ad-hoc manner solely based on the investigation of the local node density.

4.1.5 Impact of the Degree of the Approximation on the Error

The approximation of the function $H(d_n)$ can be expressed with polynomials of different degrees. To measure the impact of the degree, thus the accuracy of the approximation on the distance estimation, we rerun the simulations under the same conditions using those different polynomials. The estimated distance of every pair of adjacent nodes in the network has been computed using the DIN algorithm. This calculated distance is then compared to their real Euclidean distances. The average error values for the polynomial curves of degree 3, 5 and 7 are shown with the help of interquartile bars (see Figure 4.2).

We chose to use interquartile diagrams since they allow the judgment of the value dispersions of the distance errors. Due to the fact that the data presented in Figure 4.2 did not yield a critical difference in the average error computation, we can safely use the polynomial approximation of degree three without significant loss in accuracy.

4.2 Simulating the DIN Algorithm with ns-2

An important issue to examine with the help of ns-2 has been to determine the behavior of DIN under variable network settings. The main parameters used in our simulations are

condensed in Table 4.1. We obtained the results in the previous section with a fixed network size while varying the number of nodes. In contrast, subsequent simulations feature a fixed number of 100 nodes. The network size was increased until the density of nodes became too sparse, thus the network disconnected. The same effect can be obtained when changing the transmission range of the nodes accordingly. Every node in the network computes its relative distances to those nodes that are in its transmission range by using DIN.

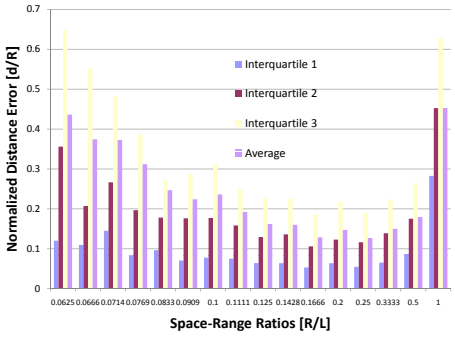
Table 4.1: Simulation parameters

Simulation parameters	**Range**
Wireless Network	802.11, 11MBit/s
Radio Propagation Model	Free Space
Transmission Range	250m
Routing Algorithm	AODV

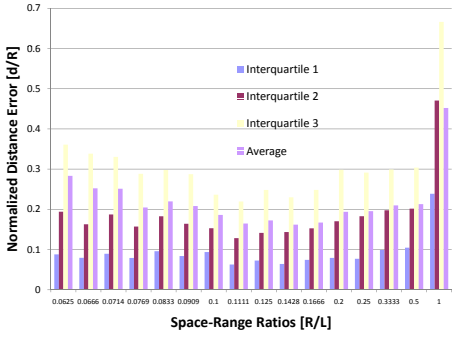
The corresponding absolute error of the estimation is simply the absolute value of the difference of the actual distance between nodes and the calculated distance. In the following, the normalized error is therefore calculated by dividing the absolute error by the radius of the transmission range of a node.

In order to also be able to compare the error independent of variable radio communication ranges and network sizes, we define the Space-Range Ratio (SRR) depicted on the x-axis of Figure 4.3 a and b. This is simply the radio communication scope of the idealized node over the length of one of the sides of the deployment area. A value of 1 for SRR therefore equals a transmission range covering the complete network.

While DIN has been developed under the assumption of uniformly distributed networks, we want to measure the accuracy of distance estimations for near-uniform distributions in a second step.



(a)



(b)

Figure 4.3: Absolute normalized errors in distance estimation versus covered radio range in a uniformly distributed network (figure (a)) and Absolute normalized errors in distance estimation versus Space-Range Ratios in a horseshoe setup (figure(b))

4.2.1 Uniform Network Distribution

In Figure 4.3, the absolute normalized errors in distance estimations of ten simulation runs are shown. Unlike [12], we take into account the error of all nodes including border nodes, although these nodes contribute to the error with higher values since their calculation depends on less available information. This way, the error of the complete network instead of a central area is measured. Nodes that can not derive their distances to other ones, due to the lack of neighboring nodes within their range, contribute with a value of normalized error distance equal to 1, thus incrementing the error average in the interquartile statistics.

Figure 4.3 a shows that the DIN algorithm produces the best performance with an SRR value of 0.1666, where the 25% of the estimations (first interquartile)have normalized error values less than $0.0532R$. Compared to our previous work, we discovered that the DIN algorithm yields better performance than WDNI [21] algorithm, where the smallest normalized error reported was a value of $0.16R$. Although the absolute normalized distance error in Figure 4.3 a shows the trend to decrease with increasing node densities.

Unlike WDNI, the DIN algorithm uses solely the number of local nodes without the help of a weighting function, thus it is not compensated for high node densities. We can denote that the duty zone of DIN is between SRR values of 0.0769 and 0.5. Those values represent deployed spaces with L values from $2R$ to $13R$ respectively. In this interval, we can see that the normalized distance error for the 75% of the estimations is less than $0.39R$ (see interquartile 3).

For bigger deployed areas than SRR values of 0.0769, the average normalized distance error increases smoothly. This is due to fact that the connections in the network start to break so the interquartiles begin to reach the maximum normalized error.

4.2.2 Near-Uniform Network Distribution

A different network setup to test DIN, such as a near-uniform distribution of nodes, is an interesting subject to evaluate the flexibility of its usage. Therefore, the nodes were positioned in a way that the network takes the form of a horseshoe. This means that the center and one side were left empty, and all nodes are spread out on the remaining three sides. This allows for testing the response of DIN to high densities both close and far away from a node with few nodes in a medium distance.

In Figure 4.3 b, we can observe that for SRR values between 0.0625 and 0.5, the normalized average error and the 75% of the error values in every case is lower than $0.37R$. For this configuration, DIN has better performance than in the uniform distribution. That means for values of SRR smaller than 0.0769 in Figure 4.3 b, it continues displays lower error under to $0.37R$.

Unlike the uniform distribution, the errors in a horseshoe set up increase in a smooth way. That is due to the smaller configuration in the deployed area compared to the uniform distribution, in such a way that the network remains connected longer but produces larger errors for bigger deployed spaces.

Looking at the normalized error of interquartile one, we realize that the best 25% of the errors is presented for an SRR value of 0.1111 showing error values lower than 0.0627.

Once again, we can observe an operation area in terms of SRR values. For high node densities (SRR=1) the distance estimations are less accurate. They assume to be uniformly distributed, causing high error rates in the distance estimations. On the other hand, DIN loses precision for networks with low node densities due to the lack of nodes, producing less available information.

Table 4.2: Minimal and maximal normalized error values in simulations of the uniform and horseshoe distributions

	Uniform Distribution							
Norm.	1_{st} Interquartile		2_{nd} Interquartile		3_{rd} Interquartile		Average	
Error	Value	SRR	Value	SRR	Value	SRR	Value	SRR
Min.	0.0532	0.1666	0.1061	0.1666	0.186	0.1666	0.127	0.25
Max.	0.0966	0.0833	0.1968	0.0769	0.387	0.0769	0.3118	0.0769
	Horseshoe Distribution							
Norm.	1_{st} Interquartile		2_{nd} Interquartile		3_{rd} Interquartile		Average	
Error	Value	SRR	Value	SRR	Value	SRR	Value	SRR
Min.	0.0627	0.1111	0.1283	0.1111	0.2195	0.1111	0.1618	0.1428
Max.	0.1046	0.5	0.2	0.5	0.36	0.0625	0.2832	0.0625

Table 4.2 shows the minimum and maximum values over all the experiments obtained with the different network distribution using the `ns-2` simulator. In this section, we confirm that a node which uses the DIN can estimate distances between their neighbors as long as the transmission range is set to a value that enables most of the nodes to experience a neighborhood close to a uniform distribution. The second step to test the accuracy of our algorithm is implementing DIN using real hardware; the results of this new test are presented in the next section.

In this last two sections, we describe the performance of the DIN algorithm discovering that the algorithm produces competitive values for uniform and near-uniform distribution (see table 4.2). We confirm once again that our algorithm works as long as the transmission range is set to a value that enables most of the nodes to experience a neighborhood close to a uniform distribution.

4.3 Experimental Evaluation of DIN

Simplifying assumptions about radio propagation, network coverage and node distributions are common in network research. The core idea of DIN uses a circular transmission range of nodes to find a relationship between distances and local node densities. Since the results of the simulations were very promising, we decided to implement DIN on our WSN hardware platform and measure the impact of a real environment on the performance of the algorithm.

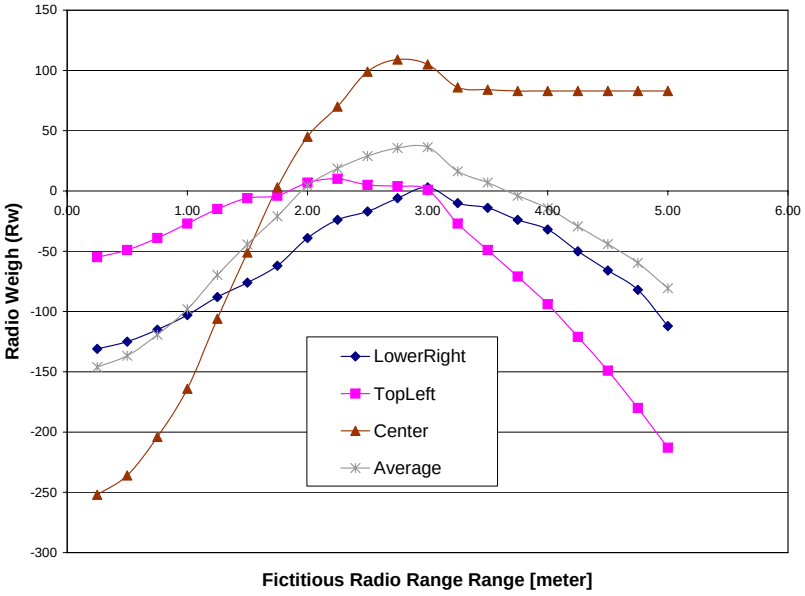


Figure 4.4: Radio Weigh curves to determine a general approximation of the radio communication range.

4.3.1 ScatterWeb Sensor Network Platform

For the set up of our testbed, we rely on the ScatterWeb Modular Sensor Boards (MSB) [22]. These nodes feature the 16-bit microcontroller MSP430F1612 from Texas Instruments equipped with 55 KB of flash memory and 5 KB RAM, and a Chipcon CC1020 transceiver using the ISM band at 869 MHz. The transceiver is able to monitor the received signal strength (RSSI). The transmitted power can be set directly in software. Additional sensors are available on a separate sensor board to be plugged onto the core board if needed.

4.3.2 Communication Range Calibration

Since our algorithm depends on a spherical radio propagation model described in section 4.1.2, we rely on mapping a transmission power setting to a known distance to calibrate DIN. This calibration was obtained by testing the behavior of the radio with different transmission power settings.

The first problem we encountered was that even with the smallest value for setting the transmit power, it was not possible to construct an experimental setup where not all nodes were within each other's transmission range indoors.

To solve this problem, two different strategies have been explored. On the one hand, we experimented with sensor nodes with disassembled physical antennas, where only the small welded copper footprint on the node itself is left for communication. On the other hand, we simply used the RSSI value to artificially limit the transmission range by filtering signals below a certain value. Although at first glance this solution may seem to be a testbed workaround, the results obtained will still be valid in a larger multi-hop environment since

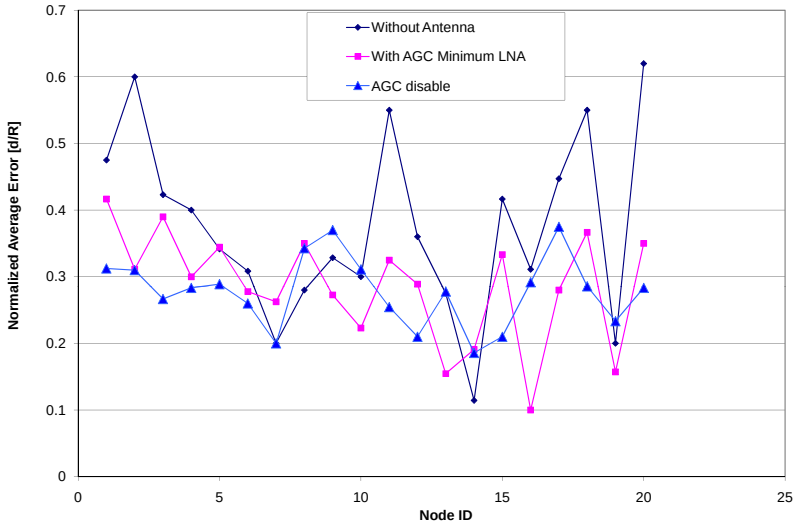


Figure 4.5: Normalized average error per node with different transceiver settings with a near-uniform node distribution

the filter will be used equally here, thus no difference in the behavior of DIN will be observed.

Another important issue we encountered is how to evaluate the quality of the transmission range in terms of temporal and spatial fluctuation of the area covered by the signal. Once again, this setting has a high impact on the performance of the algorithm since the distance estimation directly depends on the radio link. In order to choose the best configurations for the transceiver in terms of least fluctuations, the transmission range of a transmitting node was analyzed by measuring the communication link on different peripheral points, as discussed in section 1.2.

The next calibration for an approximation of a circular transmission range was obtained by mapping the radiation pattern of the MSB nodes on an indoor environment. For this purpose, a sending node was located on three different positions of a 5 mx5 m square area (upper-left corner, central position, and lower-right corner) in a seminar room of our institute. Two of the created maps are shown in the Figure 1.1 a and 1.1 b.

The RSSI measurements were taken every 25 cm from an emitter node until a complete sweep of the setup area was finished. Both nodes were positioned over cleared desk height in order to provide a good transmission scenario. As we expected, these figures have confirmed that the transmission is strongly irregular and without homogeneity. The nodes can be far away from the transmitter node and still receive high RSSI values, while others are closer and exposed to lower values.

We determined a standard radio range for the DIN algorithm by analyzing every transmission pattern map previously produced, and then evaluated the quality of the transmission range in terms of fluctuations of the RSSI values of the area covered by the signal. From the measurements, we reason that with an RSSI threshold of 33 (-42.5 dBm), an artificially limited transmission range could be implemented.

To determine a standard radio range for our system, it was necessary to analyze every trans-

mission pattern map previously produced and to evaluate the quality of the transmission ranges in terms of fluctuations of the RSSI values of the area covered by signal. Taking as a reference point the position of the sender node, we created different circular transmission range in increments of 0.25 cm until it covered the complete setup area. In order to find the best circular transmission range that fits better with the RSSI threshold, we evaluated every disc communication range with the help of the variable called radio weigh (R_W) defined as follows:

$$R_W = R_I - R_O - NR_I \tag{4.11}$$

In the mathematical expression of R_W from Equation 4.11, the number R_I is defined as the number of regular points inside the fictitious radio scope within the range of the artificially RSSI limit value (33). R_O is the variable that counts the RSSI values in range, but outside a given radio range, and finally NR_I is the number of points that are inside the disc communication range, but have an RSSI value lower than 33.

By averaging all the R_W values of all the received signal strength measurements over the different scenarios, we obtained the average curve of Figure 4.4. We can observe that the curve reaches its maximum value in 3 m. Thus, we decided to consider a radio transmission range equal to 3 m using an RSSI threshold value of 33 with a transmit power of 0x01 from the CC1020 radio transceiver. We used these values as a reference for all our DIN experiments.

The radio range fluctuation also depends on intrinsic transceiver settings; two configurations that offer few fluctuations were activating Automatic Gain Control (AGC) with minimum Low Noise Amplifier (LNA) or disabling the AGC. On average, this fluctuation has a value between 15 to 20 cm. It was obtained using twenty different sensor nodes and averaging the error.

Although this fluctuation is very critical when relying on nodes with disassembled antennas, we chose to also launch experiments with these nodes, since they offer an opportunity to test the behavior of DIN with asymmetrical communication ranges. The influence of different configurations of the transceiver on the distance error estimations are shown in Figure 4.5, and will be discussed in the next sections.

4.3.3 Adaptation of DIN for Experimental Evaluation

Due to the limited computational capability of sensor nodes, we decided to adapt DIN to avoid resource-intense computations. First, we replaced the computation of Equation 4.10 with a density-to-distance lookup table. Depending on the number of nodes in common (represented by K_i) and the local node density (depicted by K_u), a node can derive its distance from a neighboring node.

The protocol of DIN proceeds in three phases. In phase one, every node in the network broadcasts a HELLO packet to discover neighboring nodes within its communication range. It is important to take into consideration that signals received with an RSSI value below 33 will be dropped automatically to preserve the artificially constructed transmission range. To avoid collisions on the medium, we implemented a delay timer depending on the node ID. The information obtained in the first phase is a neighbor table with a single entry for

each discovered neighbor.

The second step on the DIN protocol is the exchange of neighbor tables. This process allows on to find how many nodes are in the union, as well as the intersection transmission area of two neighbor nodes in the network. With this information, every node is able to compute the distance to an adjacent node by its density-to-distance table.

The main problem in exchanging neighbor tables in the network was the communication link asymmetries. Here, a sensor node can receive signals of another node perfectly, but communication in the other direction fails. To prevent retransmission of a neighborï£¡s table's request, the expiration of an internal timer limits the overall waiting time. When a node in the network experiences an asymmetric link, the DIN algorithm implemented in th nodes sets the estimation distance to the maximum value.

In the last phase, nodes consult the density-to-distance lookup table using K_i and K_u, obtained in phase two as input parameters, to finally determine the distance between themselves and another node.

The protocol of DIN can be naturally integrated into any routing overhead. The exchange of neighborhood information and HELLO packets are subject to more routing schemes and thus may also be utilized by DIN when available. Additional information such as the local view on the network of each node can be piggybacked on regular data packets to minimize the overhead for the distance estimation. Therefore, DIN can be implemented on top of existing sensor network software at very low additional communication costs.

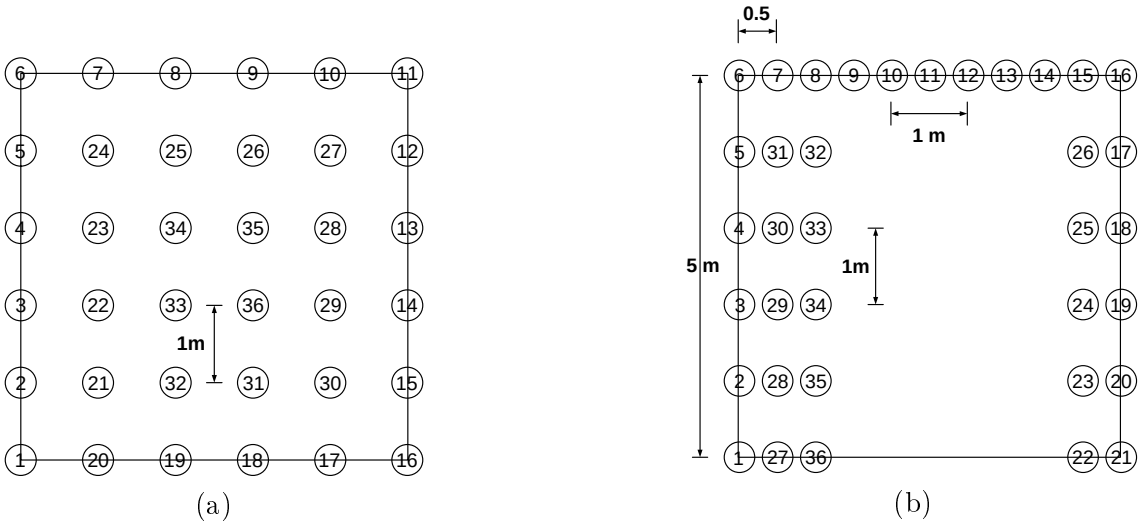


Figure 4.6: (a) Uniform node distribution, (b) Horseshoe node distribution

4.4 Testbed Set up and Experimental Results

The primary goal has been to test DIN with the best hardware configuration possible. Therefore, experimenting with different transceiver settings, as already mentioned in section 4.3.2, had to be done in-situ before obtaining the final results.

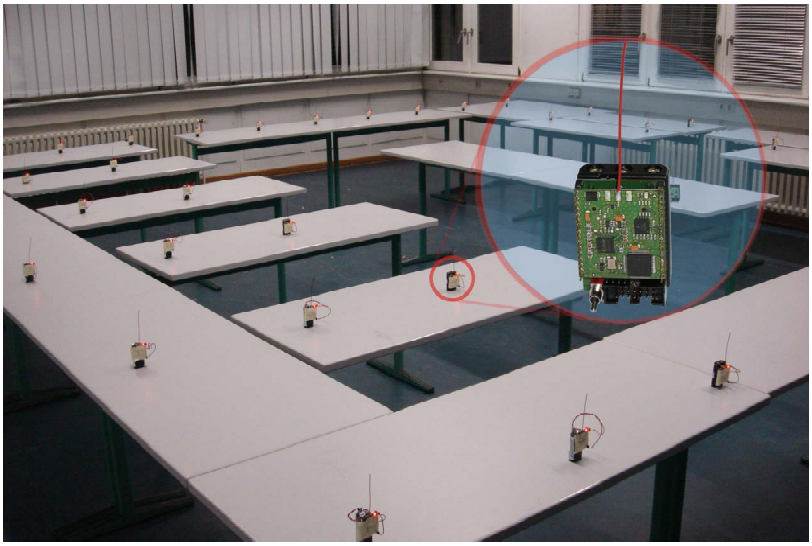


Figure 4.7: Picture of the testbed enviroment

Two different testbed layouts have been used for our experiments (see figure 4.6 a and Figure 4.6 b). The first layout is a uniform distribution of sensor nodes and the second depicts a horseshoe setup. Both testbeds have already been used in the `ns-2` simulations.

The nodes were deployed in a 5 x 5 meter square in a seminar room at our institute. We placed them on desks and the room was cleared over desk height. The room was big enough to assure a distance of at least 1.5 meters between border nodes and the walls (see Figure 4.7).

We chose the uniform layout to further analyze transceiver settings. Since we aimed for a configuration that allows for a received signal with the least variation possible, we examined the influence of the Automatic Gain Control on the complete algorithm.

Two options have been considered and tested with DIN; the activation of the AGC with a minimal setting for the low noise amplifier and disabling it. As a comparison, the test has also been conducted using sensor nodes without antenna. The results reflect our expectations (see Figure 4.5).

The error in distance estimation using no antenna suffers from high variations for the different sensor nodes due to the asymmetrical transmission ranges. AGC regulates the amplification of the received signal dependent on its amplitude and the sampling rate. Thus, the less gain is set, the less fluctuation we observe and the better the results for the distance estimation. Therefore, all following experiments have been carried out with a disabled AGC.

The main results are depicted in Figure 4.8. Here, the average normalized distance error per interquartile with the DIN is plotted, as well as the dispersion of obtained error values with the help of the interquartile diagrams. Once again, DIN works best for uniformly distributed network. The average, normalized miscalculation of the nodes of $0.325R$ in this setting equals to 0.975 m, with the best 25% of the distance calculations having an error below $0.118R$ or 0.354 m, a value that provides a good accuracy for indoor usage.

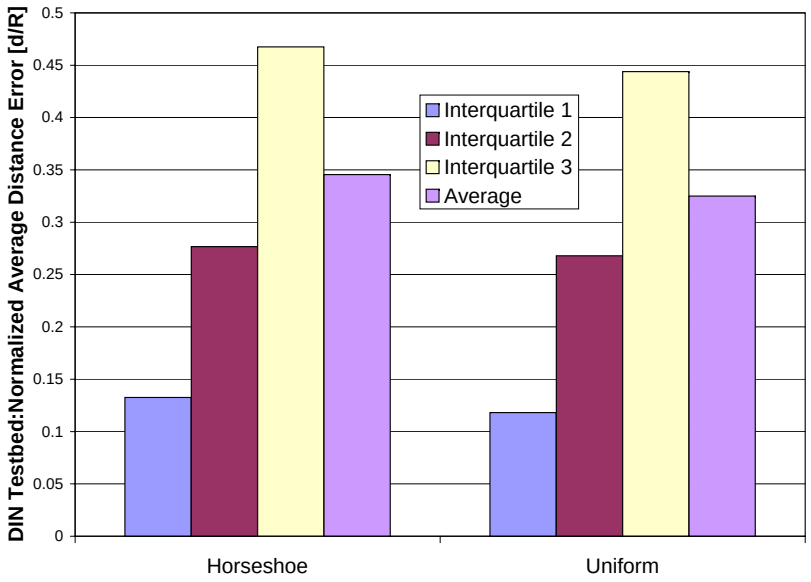


Figure 4.8: Normalized distance errors per interquartile in horseshoe and uniform distribution derived from the DIN testbed.

In 75% of all cases, the error remains at a value of $0.443R$ or a maximal offset of 1.3 m within acceptable bounds. On the other hand, the horseshoe distribution remains below a threshold of $0.15R$ which is equivalent to 0.45 m in interquartile 1, and features an average error of roughly 1 m at the most, an observation that shows the validity of applying the DIN to near-uniform network distributions despite its initial design for uniform distributions.

An interesting question that we wanted to examine was the influence of the node placement, more precisely the membership of nodes to the border or inner portion of the network on the distance estimation.

The bars of nodes 1 to 20 in Figure 4.9 represent the error of nodes placed at the border of the network, while nodes 21 to 36 denote the inner sensor nodes. In the portion of the inner nodes, the best estimation of the network is presented with a value of 0.17R. The 94% of the estimations in this section is lower than 0.30R.

The border nodes made distance estimations with a value lower than 0.4R for the 65% of the total cases. Making a closer analysis over the network, we found that the border nodes distance estimations are worse than the inner nodes due to a poor neighbor table quality. Asymmetric links and packet collisions led to a neighbor table with far less entries than usual.

Two interesting points in Figure 4.9 are the normalized average distance estimations of the nodes 6 and 11 which present distance estimations far from acceptable. Both nodes were placed on the top corners of our experimental testbed. Analyzing its information obtained during run time, we realized that they could set a communication link with nodes that were outside of the artificial radio communication scope. Thus, they underestimate the real distance to these nodes. However, there are border nodes like the node number 3 that present good distance estimations. This is because to each calculated distance estimation has been exceptionally good.

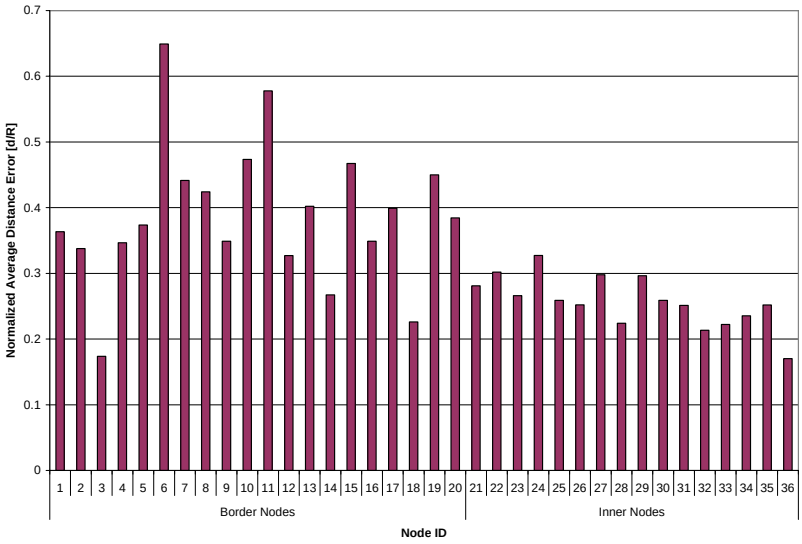


Figure 4.9: Average normalized errors in uniform node distribution for border and inner nodes in the experimental setup

Overall, we can conclude that the node placement does seem to have an influence on the distance calculation. In order to add statistically significant evidence to such a proposition, we have to evaluate more and larger scenarios.

4.5 Evaluation of DIN

In this section, we want to present details on two interesting comparisons we carried out: First, we contrasted the behavior of DIN in simulated and real environments and second, we evaluated the algorithm against other range-free distance estimation technique.

To minimize effects resulting from technological differences, we therefore implemented an RSSI-based distance estimation algorithm and compared the performance of the two approaches accordingly.

4.5.1 Comparison of Simulation and Testbed

The good results observed in the interquartile diagrams by testing the DIN algorithm on the `ns-2` simulator were confirmed by the test run conducted by the Scatterweb sensor nodes.

Looking on the interquartile 1 and 2 for uniform and horseshoe node distributions, we realized that the average normalized errors are slightly lower in a simulation environment than in an implementation on real hardware, keeping in mind that the SRR value for the experimental setup corresponds to a simulative value between 0.5 and 1 which has to be considered when comparing the overall averages of testbed and simulations results.

Taking as a reference the interquartiles with an SRR value of 0.5 for the cases of horseshoe

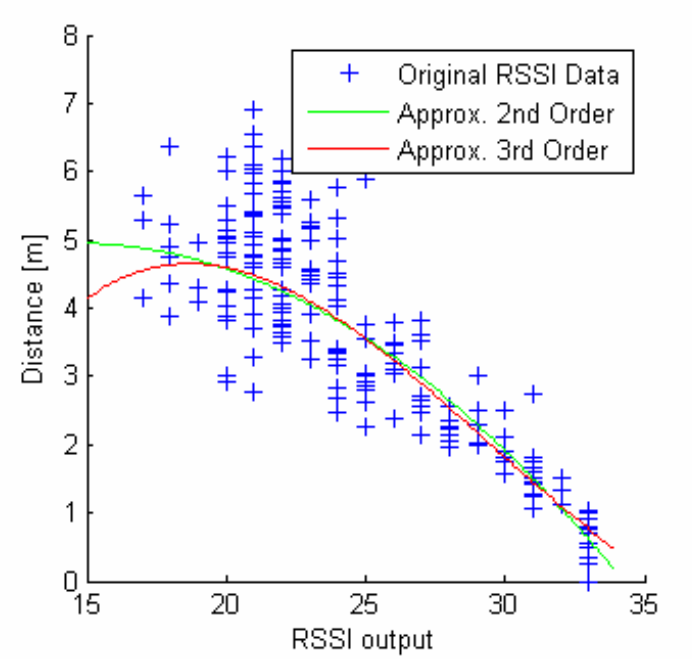


Figure 4.10: Approximation of the distance between nodes based on RSSI for indoor scenario at a transmission power of 0x01

and uniform nodes distribution on simulation diagrams, we can see that the interquartile 3 of these both distributions in our testbed add to higher values due to an increase in miscalculation. We considered that this behaviour is due to external influences such as fading, interference, or asymmetric links. However, the discrepancy on the average normalized errors between real and simulation environments is not higher than $0.14R$ which is an acceptable behavior for practical use.

4.5.2 Comparison of DIN and RSSI Distance Estimation

The evaluation of the quality of the proposed range-free distance estimation algorithm is complemented by a quantitative comparison to a similar estimation based on RSSI data. To guarantee a fair validation, the same testbed settings including the physical setup of the sensor nodes and the best transceiver settings have been used.

Furthermore, the function that maps the strength of the received signal to a specific distance as depicted in Figure 4.10 has been constructed by interweaving several RSSI measurements such as Figures 1.1 a and 1.1 b with different node positions into a suitable approximation. Note that the validity of the polynom obtained with MatLab

$$f_x = -0.0127x^2 + 0.3697x + 2.2688 \tag{4.12}$$

is constrained to the measurement area since extrapolation instantly leads to untolerable errors, a fact that once again emphasizes the need for careful calibration when relying solely

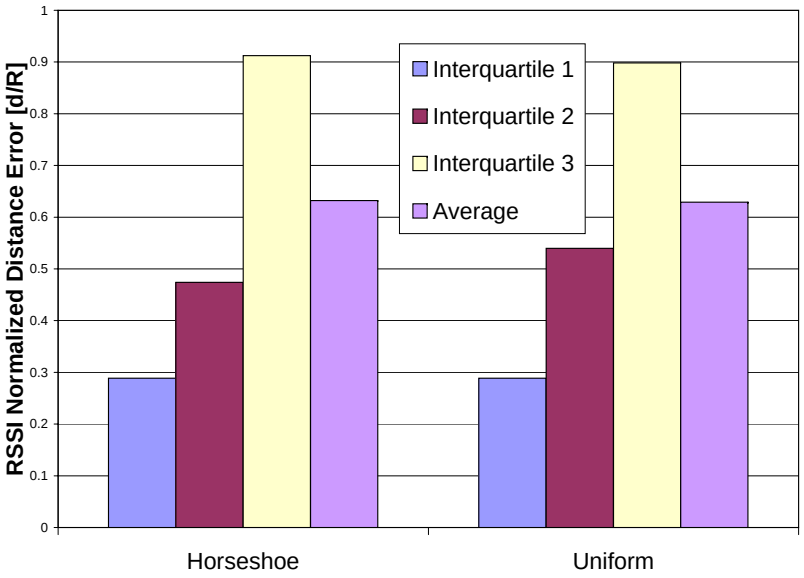


Figure 4.11: Averaged normalized errors per interquartile in a horseshoe and uniform distribution derived from RSSI distance estimation

on RSSI values.

Within the RSSI comparison testbed, the protocol used to determine distances in the network was developed as follows: Every node in the network has the opportunity to broadcast its ID. The receptor nodes register the signal strength of this packet and computes its distance to the sender nodes of the network by substituting the value of the received signal strength for x in Equation 4.12 and solving for f_x to obtain the corresponding distance.

As we already mentioned, the same 36 MSB nodes with the two different testbed (see Figure 4.6 a and 4.6 b) layouts were used to test the RSSI-based distance estimation. The result of our experiment is shown in Figure 4.11.

Although the normalization of the results is artificially applied after the collection of the data, it has been chosen for comparability reasons. Results can easily be remapped to discrete error values by multiplying them by 3 m (the fictitious radio obtained in section 4.3.2). The dispersion of the error values is once again illustrated with the help of interquartile diagrams.

The data on RSSI distance estimation as shown in Figure 4.11 reveals the weaknesses of relying solely on RSSI readings. In average, RSSI distance estimation errors are almost twice as high for all the tested scenarios than the ones provided by DIN, leading to an average misplacement of 1.88 and 1.89 m for a uniformly and near-uniform distributed network respectively. As we expected, the RSSI measurements lacked the required accuracy to determine distances between adjacent nodes.

For a view on the normalized average error per node see Figure 4.12, which nicely illustrates the superiority of DIN in all examined node distributions, a result that meets our expectations. The approximation simply lacks the required flexibility to cope with the problem of fluctuation of the received signal strength maps and their low resolution as have been

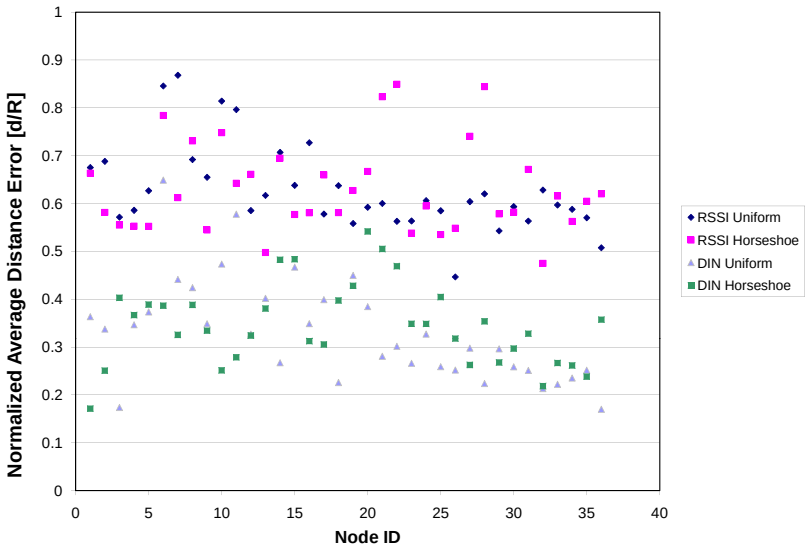


Figure 4.12: Average normalized errors per node in horseshoe and uniform distribution with the RSSI and the DIN distance estimation

Table 4.3: Testbed Comparisons

Norm.	DIN Real		RSSI		DIN Simulation	
Error	Uniform	Horseshoe	Uniform	Horseshoe	Uniform	Horseshoe
Min.	0.00047	0.003833	0.003957	0.00766	4.52E-5	9.99E-6
Max.	1.3132	1.1574	2.1666	2.089	0.602477	0.7145

discussed in section 1.2, a problem that DIN is able to at least partially solve with its exploitation of the knowledge of local node densities. The best and worst normalized error values for the distance estimations in the different scenarios and implementations are depicted in Table 4.3.

4.6 Conclusion

In this chapter, I presented the DIN algorithm which estimates distances between two adjacent nodes based solely on local neighborhood information. As a foundation, the area of intersection of two overlapping transmission ranges has been related to the number of nodes in a uniformly distributed network to determine the internodal distances.

To investigate the accuracy of the DIN algorithm, we rely both on simulations and experiments with real sensor nodes and provide an exhaustive evaluation against another range free distance estimation algorithm. One can observe on the simulations a tendency towards better results in medium and high densities networks. The duty zone of DIN is between SRR values of 0.0769 and 0.5 for uniformly distribution networks and between SRR values

of 0.0625 and 0.5 for near-uniform distribution networks with the majority of normalized error values below 0.39R. Here, the best average, normalized distance error has been 0.127R for a uniform distribution of sensor nodes.

The good results obtained putting into practice the DIN algorithm with different testbed layouts, reflect the findings of the simulations as can be observed in Figure 4.8, although we were only able to analyse a fraction of the simulation cases. Finally, we confirm the better accuracy of distance estimation of our approach comparing the real hardware tesbed results with those obtained using solely RSSI-values. The results show that up to more than the double enhancement of distance accuracies for different tesbeds (see Figure 4.12) have been achieved.

With this work, we demonstrated that DIN also works in near-uniform environments, but features slightly higher error values. The advantage of this approach is that neither the usage of specialized hardware, nor the measurements of physical properties that are inaccurate or unreliable are necessary for this estimation.

The DIN is a completely Ad-hoc algorithm that keeps the overhead in communication and calculation at a minimum. We therefore use the DIN algorithm as a ranging method to help the node discover their locations. As we will see in the next chapter, the DIN algorithm is the basis of our localization scheme because of its good performance compared to the RSSI-based ranging approach.

CHAPTER 5

Locating the Sensor Nodes

The positioning of wireless sensor nodes without dedicated hardware is an open research question. Especially in the domain of embedded networked sensors, many applications rely on spatial information to relate collected data to the location of its origin. The majority of approaches, such as those explained in chapter 3, explore physical properties of acoustic and/or radio signals such as the strength of a given received signal or its trip time. This is, however, problematic since neither the complexity on the side of the software nor the hardware is adequate enough for embedded systems.

Due to the good results obtained using the DIN algorithm to estimate distances, we were more than motivated to apply the DIN algorithm on the sensor-location area. To evaluate the accuracy of the DIN algorithm, we ran extensive simulations and we experimented with different testbed setups. So far, we have applied the DIN algorithm for localization using real sensor nodes. Finally we implemented different range-free approaches such as DV-Hop and RSSI based localization parallel to our real and simulated experiments to obtain a comparative analysis.

In this section, we will introduce a new and innovative technique to find the position of nodes deployed in a distributed network based on the DIN algorithm. This new algorithm called the Positioning Iterative Vector (PIV) could be used either as an iterative approach to determine node positions or as a refinement phase of any other location-sensing method.

5.1 Distance to the Reference Nodes

The first step for multilateration is determining the distance between unknown nodes and reference nodes. In chapter 4, the single hop distance-to-landmark problem using node local density was successfully solved using the DIN algorithm. If a node can directly communicate with several landmarks, its position can be discovered by the estimation of the node-to-landmark distance supplied by DIN algorithm and later solving the linear system by multilateration. Unfortunately, the assumption of a direct connection to the anchors rarely occurs in real scenarios of wireless sensor networks. In this section, we will introduce approaches that extend the neighboring distances provided by the DIN algorithm to the node-to-landmark distances.

To discover the position of an unknown node in a three dimensional space, it is necessary to estimate the distances between the node and at least four landmarks. In the final phase of the DIN algorithm, each node obtains the distances to its neighbors, so we have to extend these distances to the four reference nodes.

To estimate the distances (d_l) from the unknown nodes to the reference nodes (landmarks) in our simulations, we explored three possible solutions: Extension of the DIN algorithm (ExDIN/DV-Distance), Factor Correction (FC), and Factor Correction per Hop (FCH).

5.1.1 The ExDIN Algorithm or DV-Distance algorithm

The ExDIN algorithm is based on the concept of adding the distances produced by the DIN algorithm, thereby finding the shortest path between the unknown nodes and the landmarks. The ExDIN process begins when the landmarks flood its own positions throughout the network. When the first unknown nodes have received the broadcast of those landmarks, the nodes add its own DIN-distance to that shared landmark packet. After the addition of its distance value to the landmark's data, these nodes broadcast the new packet to other adjacent nodes, who will integrate to the package the neighboring distances from the nodes which have recently been heard. In this manner, the nodes construct paths following a distance-vector approach (similar than APS) where ranging information is flooded throughout the whole network in a hop by hop fashion.

A node is allowed to send the distance-to-landmark packet into the network again when its distance-to-landmark is updated. That means if an unknown node obtains a new packet data from another adjacent node with a shorter distance-to-landmark value, it will replace the old distance-to-landmark path with the new shorter one. Once the node has updated its own distance-to-landmark, it will schedule its retransmission of this new data packet into the network.

Assuming different connectivities in the network from node to node, we describe the distances between a given unknown node and a landmark l as follows:

$$\hat{d}_l = \sum_{\forall(v.u)\in p_l} DIN(v,u) \tag{5.1}$$

where p_l is the shortest path to landmark l.

The subsequent simulations feature a fixed number of 100 nodes in a square area with a network length of L meters. The node transmission range (R) is 250 meters. Once again, we increased the network length in increments until the network became too sparse and communication between nodes was no longer possible. Four landmarks have been placed on every corner of the network area, so the unknown nodes in the network could compute their distances to those landmarks using the DV-dist algorithm.

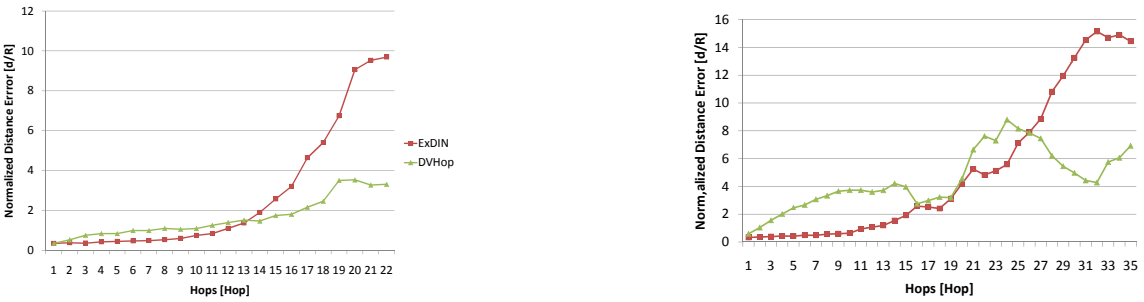
The idea of using the shortest path between nodes and landmarks have also been utilized in the Ad-Hoc Positioning System (APS) [51] of Niculescu and Nath, where node-to-landmark distances are delivered as a distance vector in hops (DV-Hop). The following simulations utilized the DV-hop technique to validate our results.

Table 5.1: Interquartile Comparisons of ExDIN and DV-Hop methods for a length square side of 1500 meters

SRR=0.1666	Interquartile 1	Interquartile 2	Interquartile 3	Average
ExDIN	0.20957	0.42007	0.69293	0.59209
DV-Hop	0.18263	0.40096	0.74889	0.55368

5.1.2 Simulation of the ExDIN Approach

Figure 5.1 shows two distance error curves produced by simulations of the ExDIN and DV-Hop algorithms with a length square side of 1750 and 2000 meters. The radio range of every node in the network was 250 m. These two interlaced plots confirm that the ExDIN algorithm can out-perform the DV-Hop algorithm under the first distance hops from the landmarks. For nodes that are more distant from the reference nodes, the estimations of distances by ExDIN are longer than those computed by the DV-Hop algorithm.



(a) (b)

Figure 5.1: Comparison of the error resulting from the simulations of the ExDIN and DV-Hop simulations for an uniformly distributed network with (a) $L = 1750$, and (b) $L = 2000$ meters

Figure 5.1 (b), the performance of the ExDIN algorithm decreased compared to the DV-Hop efficiency. This effect is due to the accumulative distance error of adding the DIN distances through the network. The more distance in hops the nodes are, the higher accumulative landmark to node distance error is obtained. Although DV-Hop also experiences this effect, the average value of the distance-hop makes up for better results.

In chapter 4, we mentioned that the best performance of the DIN algorithm was produced with a side square of 1500 meters (or 0.1666 SRR value) for a uniform distribution. With this SRR value, 25% of the estimated distances (see first interquartile of Figure ref-Fig:Ns2Interquartiles) have normalized error values less than $0.0532R$.

Extending the neighbor distances to the landmarks, we find for a similar range that 25% of the ExDIN algorithm has normalized error values less than $0.2096R$, while DV-hop has error values less than $0.1826R$, as presented in the interquartile comparison of Table 5.1.

Analyzing the curves of Figure 5.1, Table 5.1, and other curves produced with different SRR

values in Apppendix C, we discovered the best result of neighbor distances does not imply the best distance to the reference nodes. The normalized error using ExDIN increases little by little with every increment of the hop-distance.

Looking closer at the results of the different simulations, we conclude that the progressive increment of the normalized error for long node-to-landmark paths are caused by deviation of the straight line connections, i.e., the path from landmark to a given node in the network rarely plots a straight line, but rather takes more the form of a snake. We presume a connection of the shortest path is still an overestimation of the real distance between faraway nodes and landmarks (see Figure 5.2).

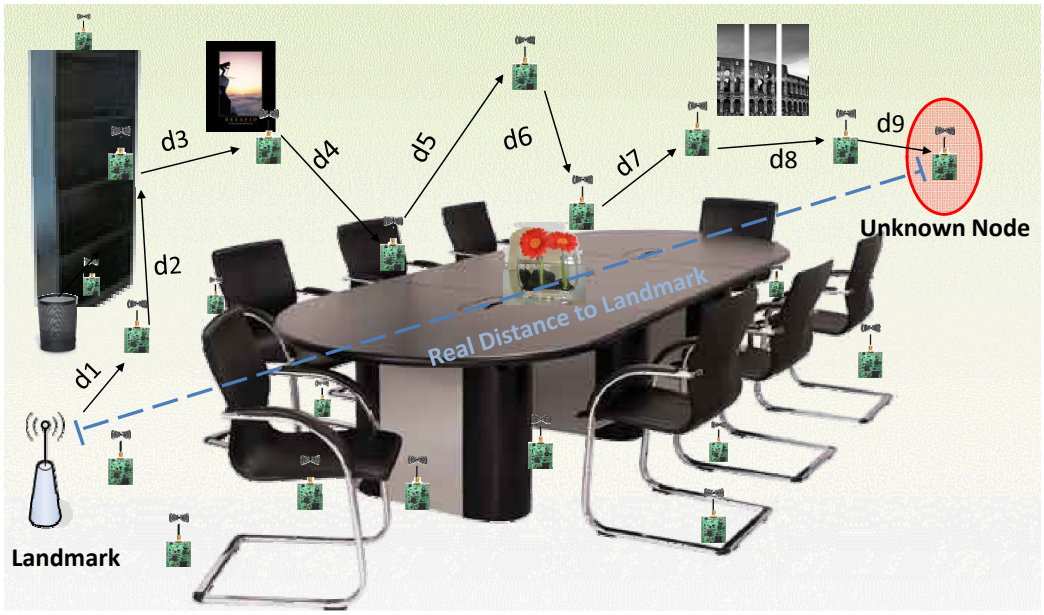


Figure 5.2: Accumulative distance error adding node to node DIN or DV-Hop distances

5.1.3 The FC and FCH Algorithm

The detour effect that provokes the overshot of normalized errors using ExDIN motivated the improvement of the extension of the distances using the DIN algorithm. Due to the cumulated errors per hop produced using ExDin and inspired by the APS method, we propose two new attempts to estimate node to landmark distances.

Working with uniformly distributed networks, we experienced over-estimation on entire paths from a reference node to an arbitrary unknown node. Trying to alleviate this phenomenon, we proposed the method called Factor Correction. The Factor Correction is defined as the average of the real distance ($d_{l_i real}$) to the estimate distance ($d_{l_i estimated}$) ratios between a landmark i and the other reference nodes in the network.

To better visualize the idea behind the Factor Correction, let us consider the situation shown in Figure 5.3 where four landmarks have been deployed in a square area similar to our simulations.

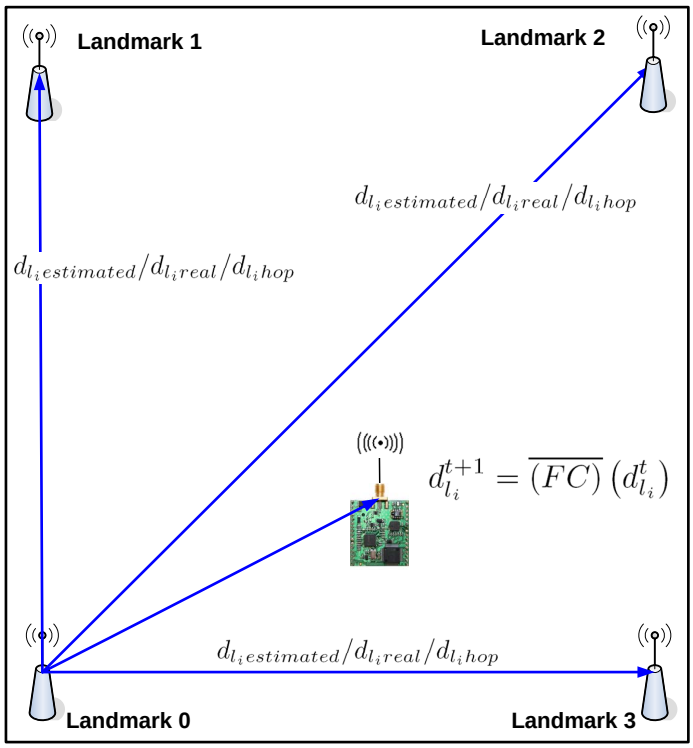


Figure 5.3: Correction of the estimated node to landmark distance

Every arrow in the figure has three different labels. The label $d_{l_i estimated}$ is the distance yielded by adding DIN-distances from landmark "0" to landmark i, the variable $d_{l_i real}$ is defined as the real distance between landmarks, and $d_{l_i hop}$ represents the distance from landmark to landmark in hops. The mathematical expression to calculate the factor correction is:

$$\overline{FC} = \frac{\overline{d_l real}}{d_l estimated} = \frac{\frac{d_{l_1} real}{d_{l_1} estimated} + \frac{d_{l_2} real}{d_{l_2} estimated} + \frac{d_{l_3} real}{d_{l_3} estimated}}{3} \tag{5.2}$$

The correction factor implemented in our simulations determines the ratios of the real distances from landmark "0" to the other landmarks labeled as 1, 2 and 3 respectively in Figure 5.3. By using Equation 5.2, a node in the network is able to improve its distance to the landmark by just multiplying the correction factor and its estimation distance to the landmark. Thus, the new approximation of the node-to-landmark distance is

$$d_{l_i}^{t+1} = \overline{(FC)}\,(d_{l_i}^{t}) \tag{5.3}$$

where $d_{l_i}^{t+1}$ is the new distance estimation, FC is the Factor Correction and $d_{l_i}^{t}$ the estimated distance produced by ExDIN.

The landmark-to-landmark and landmark-to-node lengths provided in hops in Figure 5.3 have been taken as supplementary information to shape our third attempt to improve the

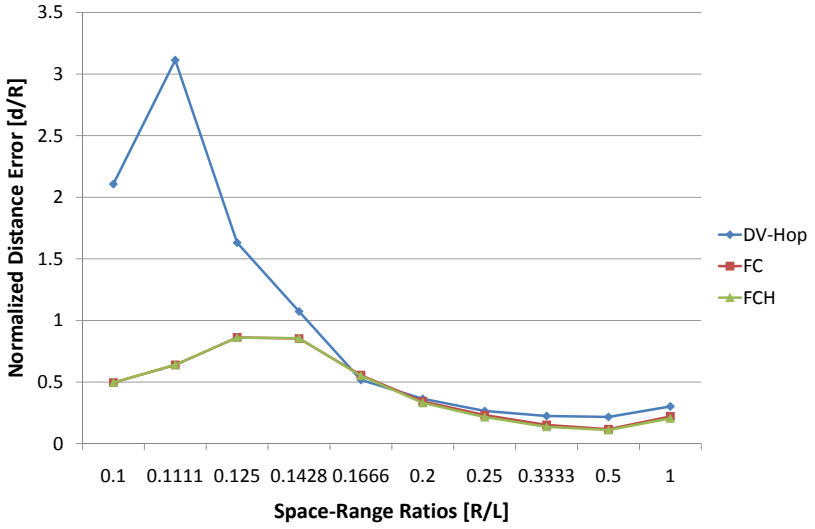


Figure 5.4: Average normalized errors in distance to the landmarks estimations versus covered radio ranging in uniformly distributed networks

distance to landmark values. This new approach has been named Factor Correction per Hop (FCH).

The FCH method improves the distances by taking into account the influence of the error estimated per hops. In contrast to the APS algorithm, the FCH algorithm employs the ExDIN algorithm to derive a first distance estimation and later figure a factor correction per hop.

First, the real distance between landmarks ($d_{l_i real}$), its distance in hops ($d_{l_i hop}$) and its ExDIN distances ($d_{l_i estimated}$) are required to produce the new Factor Correction. Adding the difference between the real distances and the ExDIN distances results in the total error rate. Finally, the average error caused by one hop is produced by the ratio between the total error rate and the addition of the hop distances between landmarks.

$$FCH = \frac{\sum_{i=1}^{n} d_{l_i real} - d_{l_i estimated}}{\sum_{i=1}^{n} d_{l_i hop}} \tag{5.4}$$

Since we are interested in finding an expression for refining the node-to-landmark distances, we have to multiply the distance-hops from a given node and the FCH value from Equation 5.4. The new approximation for the distance-to-landmark length is the addition of this product (FCH) and the estimated distance to the landmark ($d_{l_i estimated}$), computing in the node by EXDIN as described in Equation 5.5

$$d_{l_i}^{t+1} = d_{l_i estimated} + (d_{l_i hop})\,(FCH) \tag{5.5}$$

5.1.4 Simulating the FC and FCH Algorithms

The average normalized errors of distance-to-landmarks estimations of ten simulation runs using the FC, FCH and DV-Hop methods are presented in Figure 5.4. Four reference nodes are deployed in every corner of the square area where the unknown nodes are distributed. The length of the square side (L) was incremented until the nodes in the network lost the connection-hops with the landmark, meaning the nodes were not more capable of calculating the direct or indirect distance to the landmark due to the lack of connections with neighboring nodes. Every simulation takes into account the results of distance to landmark estimations of a fixed number of 100 nodes.

Figure 5.4 shows the simulations of the FC and the FCH methods overcome the DV-Hop error values. Although the absolute normalized distance error of FC and FCH for SRR values from 1 until 0.1666 are slightly lower than those generated by the DV-Hop evaluations, a significant difference can be calculated from SRR values higher than 0.1428.

Looking closer at the simulation results, we found the path to the landmarks is improved by FCH for relative low node densities because the nodes make a beeline for the landmarks, i. e., the FCH can better discover the node distances when the trajectories are free from snake paths.

The best accuracy is obtained by the FCH algorithm for a SRR value of 0.5 with an average normalized distance error of $0.11R$. Unlike the DV-hop technique, the FCH algorithm holds an average normalized distance errors less than $0.86R$ throughout the SRR values. The curves of normalized distance errors of FC, FCH, and DV-hop methods versus the number of distance hops for different network densities presented in Appendix D confirm the good performance of the FCH algorithm. The best example is Figure D.1 b, where analyzing the best 25% of the distance estimations have a normalized error less than $0.04403R$. The nodes in the following simulations will use the FCH algorithm as the input parameter to discover its own position in the network, unless specified otherwise.

5.2 DIN Based Localization: Simulation Environment using ns-2 Simulator

After obtaining the distances to the reference nodes, the unknown nodes begin to determine their positions. In order to calculate its own location, the node requires the coordinates of the reference nodes and its distance in relation to the Landmarks.

In Chapter 3, the most common distributed and centralized algorithm for localization was presented. The most popular, ad-hoc, and simplest distributed algorithms for deriving a position are multilateration and bounding box methods.

We decided to implement both methods in our experiment using the ns-2 simulator to provide the best location-aware solution for the FCH algorithm. Although multilateration is quite expensive due to the number of floating point operations, it provides very accurate positions if the estimated landmark-to-node distances are relative precise. Nevertheless, the bounding box algorithm is cheaper in terms of computation effort and robustness. But, as we mentioned in 3.2.2, this method is very sensitive to the place of anchors, especially at the edges of the deployment network area, as well as the physical topology [68].

In order to be consistent with the previous experiment, we preserved the same ns-2 simula-

tion parameters depicted on Table 4.1 to determine the behavior of those different location-aware algorithms. A fixed number of 100 nodes under variable network sizes produce the subsequent simulation results. According to the recommendation of Savvides et al. [70], we have placed four Landmarks at the edges of the simulated network to avoid the inaccuracy of the node locations at the edge. We also used different SRR values in order to compare the error independent of variable radio communication ranges and network sizes. Every node in the network calculates its actual positions through the combination of the bounding box and multilateration methods using the distances produced by DIN or DV-hop algorithms. The corresponding absolute normalized error of the position estimation is simply the distance between the real position of the node in the network and its calculated location divided by the radius of the transmission range of the a node (R) as it is described in Equation 5.6.

$$Error = \frac{\sqrt{(x_r - x)^2 + (y_r - y)^2}}{R} \tag{5.6}$$

where (x_r, y_r) denotes the real coordinates of the node and the coordinates (x, y) is the estimated node position.

Due to the nature of the DIN algorithm, the comparisons of the combination with FCH/lateration and FCH/Bounding Box have been developed under uniformly distributed networks. After obtaining the best combination result, we compared it with the APS method. Finally, the measurement of the accuracy of position estimations for near-uniform distributions are presented.

5.2.1 Simulation of Uniform Distributed Networks

To face the issue of which algorithm combination has to be used to find the node positions, we produced 30 simulation runs of the distance algorithms and two location methods. We decided to test the worst case of DIN using the ExDIN algorithm as distance estimator versus the DV-hop technique to clarify the best node location-aware method. The average normalized error in position estimation using multilateration and bounding box algorithm are shown in Figure 5.5.

Comparing the curves of the normalized errors of Figure 5.5, we can observe when the network is less dense, the accuracy produced by the multilateration method is higher than the one obtained by bounding box. The accuracy using the bounding box fails for SRR values below 0.2 with the combination of either ExDIN or DV-hop algorithms. This is due to the lack of precision of the algorithm by nature when setting the position estimation at the centroid of the square of the Min-Max square overlapping area (see section 3.2.2). Therefore, all following experiments have been carried out with the multilateration approach.

As mentioned in section 2.2.1, multilateration is sensitive to the accuracy of the distance estimations, but at the same time multilateration tries to minimize the possible estimation errors. The node-to-landmark distance estimations produced by the FCH algorithm presents more precise measurement, therefore we were motivated to use it in order to improve the position estimation. The main results are depicted in Figure 5.6: here, we plotted the average and the interquartiles of the normalized position errors yielded by DV-Hop and

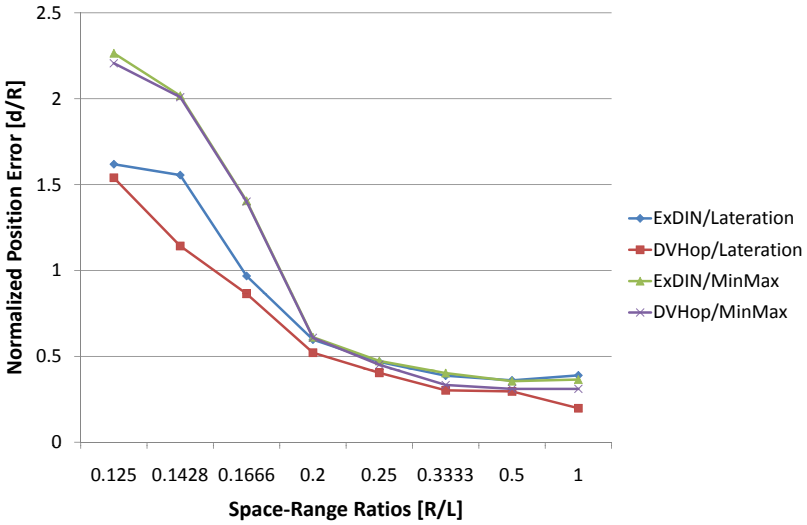


Figure 5.5: Comparison of position errors resulting from the combination of different position algorithms and distance estimation methods

FCH algorithms using multilateration for different SRR values.

As expected, the absolute normalized distance errors of DV-hop and the FCH algorithms show the trend to decrease with increasing density of the network. When we looked at the average distance error using the FCH algorithm, we recognized that the best performance is obtained for SRR values of 1,0.5 and 0.3333 with normalized average error values lower than 0.2. The best 25% (see the first interquartile) with SRR value of 0.5 has normalized error value of 0.1 using the FCH algorithm. This value represents an improvement of 59% over the same interquartile using the DV-Hop algorithm. This is due to the fact, that nodes have more information to position themself better with the help of the FCH algorithm.

Once again, when the values of SRR decrease, the connections in the network start to break. This result suggest that nodes have no further communication through other nodes which link them to the landmarks making the multilateration to locate itself impossible. This effect is appreciated observing the errors produced with a SRR value of 0.125 for both algorithms, where fewer nodes add to the overall error.

5.2.2 Simulation of Near-Uniform Distributed Networks

To evaluate the flexibility of the FCH algorithm, we have simulated implementation of different setups with near-uniform distribution. Like the structure in Chapter 4, the nodes were positioned in the horseshoe setup distribution. As in our previous simulations, the results are presented through different network densities and using Interquartiles.

Figure 5.7 shows the results of 30 simulation runs using the FCH, DV-hop and multilateration algorithms. In this network distribution, the normalized position errors are quite similar. For SRR values lower than 0.5, the FCH algorithm overcomes slightly the DV Hop performance. Only in the case of SRR values of 0.5 and 1, the DV Hop algorithm presents somewhat better results than the FCH algorithm. Because of the similarity between the in-

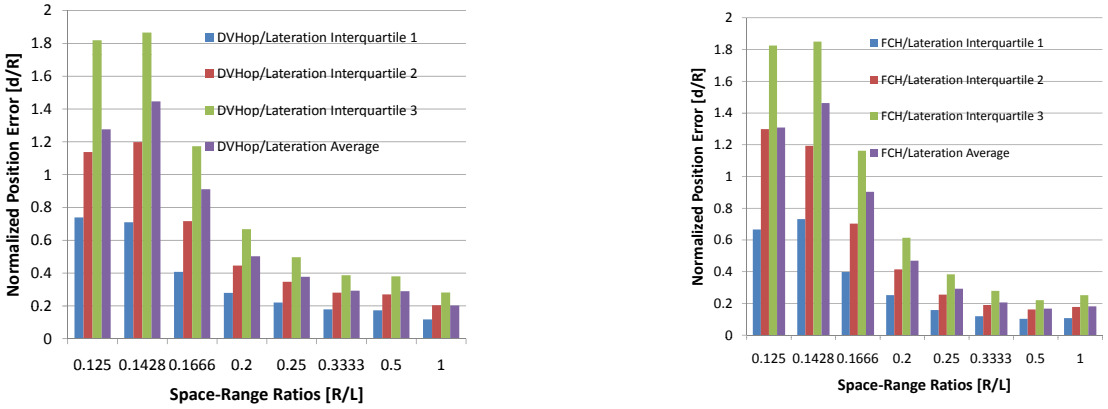


Figure 5.6: Normalized positions errors using the (a) DV-Hop and (b) FCH algorithms

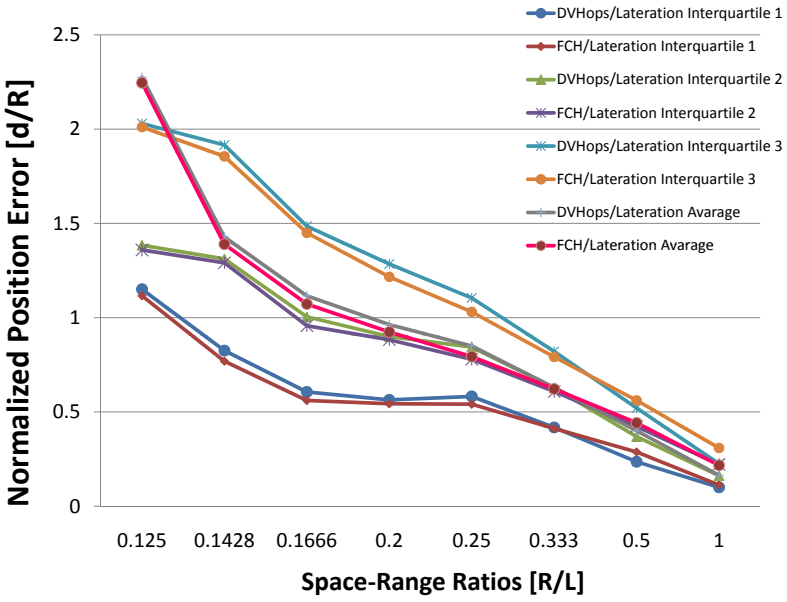


Figure 5.7: Comparison of normalized position errors produced by using the DV-Hop and FCH algorithms in a horseshoe configuration

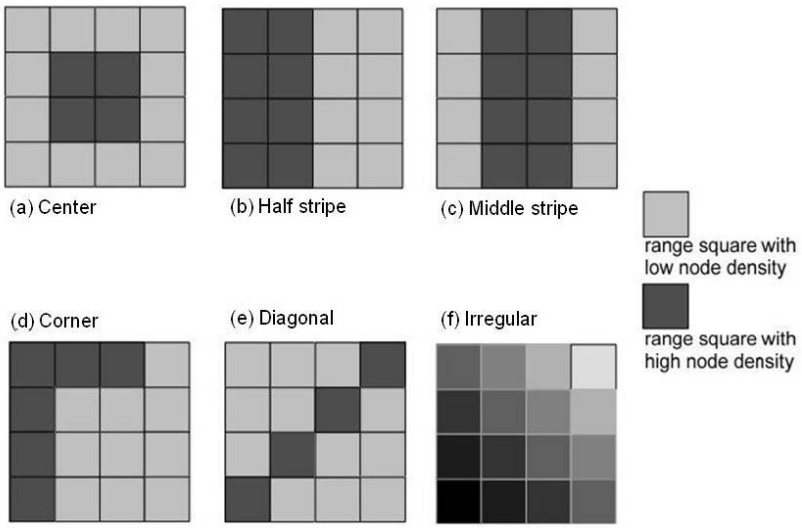


Figure 5.8: Diferent simulation testbed to measure the DV-Hop and FCH algorithms' performance

terquartiles and average curves, we decided to explore the performance of the FCH algorithm with other near-uniform distributions.

The nodes were deployed in different schemes as shown in Figure 5.8, all of them with a side length square of $4R$. To preserve the idea of a near-uniform distribution, the square deployed area was divided into 16 range squares each with a side length of $1R$.

In Figure 5.8, two kinds of range square are presented: the range square with low node density (grey squares) where a maximum number of 5 nodes were uniformly deployed and the range squares with high node density (dark squares) which contain 25 nodes uniformly distributed. We have related this manner to distribute the nodes over the low and high density range squares as sparse networks. In contrast with the sparse distributions, dense networks were populated with five extra nodes per range square. That is to say, 10 nodes for low node density range squares and 30 nodes for high node density range squares. Once again the position of the nodes over every range square in the dense network were located in a random manner.

A special case is the irregular setup (mash) where the node distribution in every range square was increased by 6 nodes more than the preceding range square on the x-axis. For every range square on the y-axis the increment of nodes respect to the preceding range square was increased by adding 7 nodes. Furthermore, for the case of sparse networks the less populated range square began with a node density of 5 nodes, while in the case of dense networks, the less dense range square had 10 nodes.

The curves presented in Figure 5.9 shows the results of 30 simulation runs in the sparse network modality previously described. The error bars in mash, middle and slash network distributions with the FCH algorithm present similar errors to the bars produced using the DV-Hop algorithm. For the remaining configurations, the comparison of position errors present high similarities on the normalized error values either using the DV-Hop or the FCH

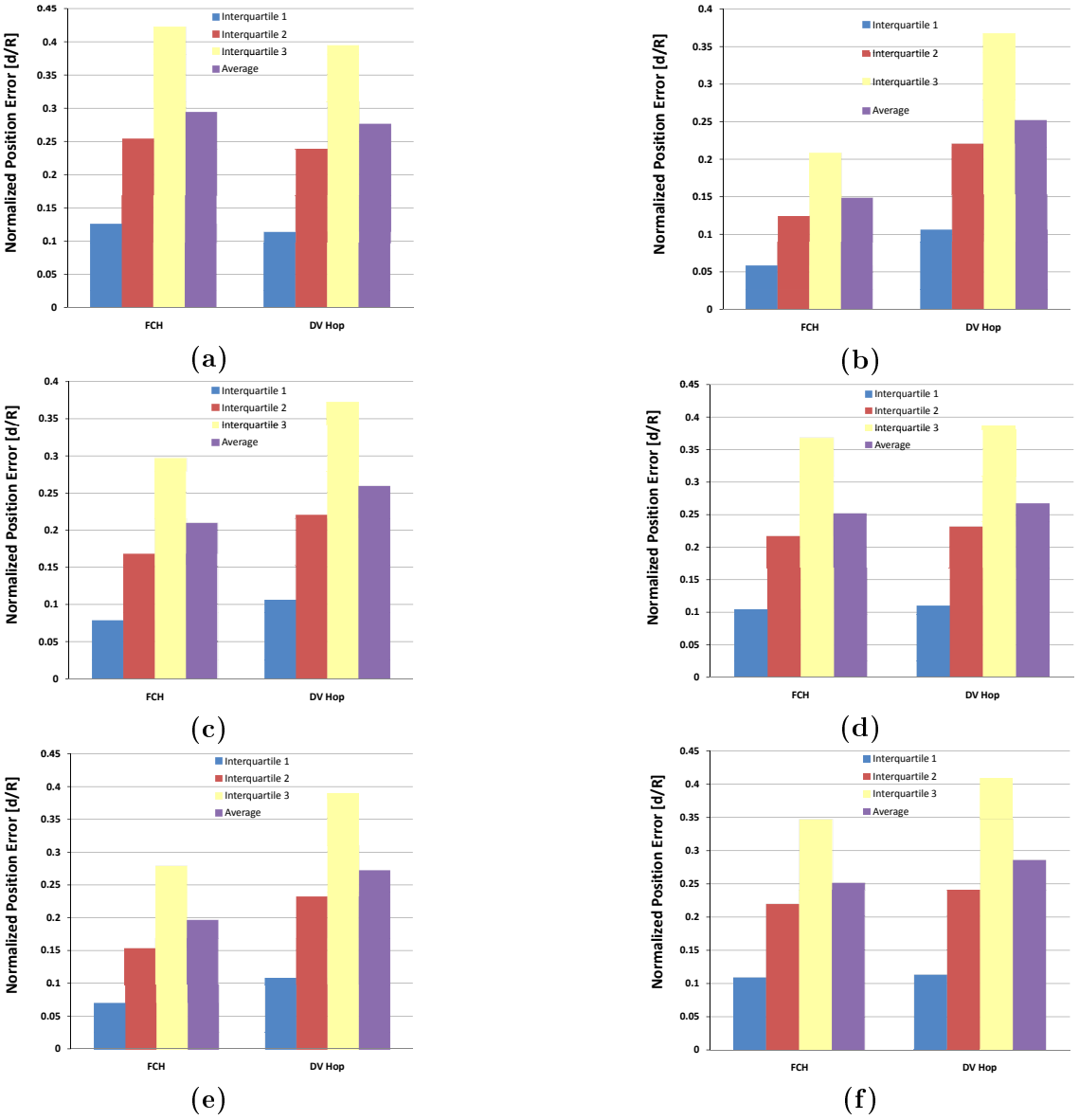


Figure 5.9: Comparison of the error resulting from the simulations of the FCH and DV-Hop algorithms in sparse networks with a) center b) mash c) corner d) middle e) half and f) slash near-uniformly distributions

algorithm. The error estimations produced in Figure 5.9 b show a considerable improvement over all the interquartiles using the FCH algorithm. The improvement on the average error to the FCH's Interquartile was 54% lower than the DV-Hop value. This effect using the irregular setup motived us to test the FCH algorithm in dense networks.

Figure 5.10 shows the interquartiles bars of the FCH and DV-Hop algorithm in dense networks. With the increment of the node densities for the different schemas, a significant decrease on the position estimation errors is computed for all the network configurations. The best normalized position errors are found in the irregular configuration. Here, the error of 75% of the position estimations is lower than 0.17 (see interquartile 3). In the same plot, we can see that the best 25% is obtained with a normalized error value of 0.049. No configuration using the DV-Hop algorithm can overcome the good results yielded by the FCH algorithm. This effect occurs due to the fact nodes in dense networks have more information through their neighborhood and this is translated into a decrease on the position errors.

In this section, when we tested the efficiency of the FCH algorithm, we realized it overcame the performance of the DV-Hop algorithm for different configurations and network densities. However, in sparse networks, the efficiency of FCH is very similar to that produced by using the DV-Hop approach. In the next section, the innovative PIV algorithm will be described as a refinement-phase algorithm, but also as an autonomous location system.

5.3 Iterative Approach to Locate Wireless Sensor Nodes

There are several approaches to improving the accuracy of the position of sensor nodes such as those mentioned in chapter 3. A common drawback in methods such as bounding box, APS or APIT algorithm is the strong impact over the accuracy of landmark-to-node distances in the location process. Although the FCH algorithm presents better position estimations compare to RSSI or DV-hop methods (as we will see in section 5.4), it also suffers from a lack of precision in the distance to anchor estimations due to the accumulative distance error per hops. This makes necessary an improved algorithm to solve this problem.

Saavides et al. have proposed in [69] an iterative algorithm to minimize the error position caused by long landmark-to-node path. In this method, an unknown node can serve as an anchor so long as it is aware of its position. One of the main requirements to use this algorithm (called AHLoS) is the presence of at least three anchors within one hop distance of a given unknown node. Thus, the unknown can compute its position by multilateration and help other nodes to find its location. The two principal disadvantages of the algorithm are, first, the need of a percentage of anchors nodes to yield multilateration and, second, they still have error accumulation using the iterative multilateration.

Another scheme which works with iterative refinement is the proposed by Savarese et al. in [68]. Here, the Hop-Terrain avoids that the range error problem similar to the FCH algorithm by counting the distance hop between anchors and nodes and multiplying that by an average-hop distance. With this correction data, the unknown nodes calculate its position by multilateration. Savarese et al. use the iterative multilateration, giving a confidence value to every single node. The confidences weigh the equation of multilateration, making stronger the impact of the iterative multilateration in nodes closer to anchors than nodes hops away

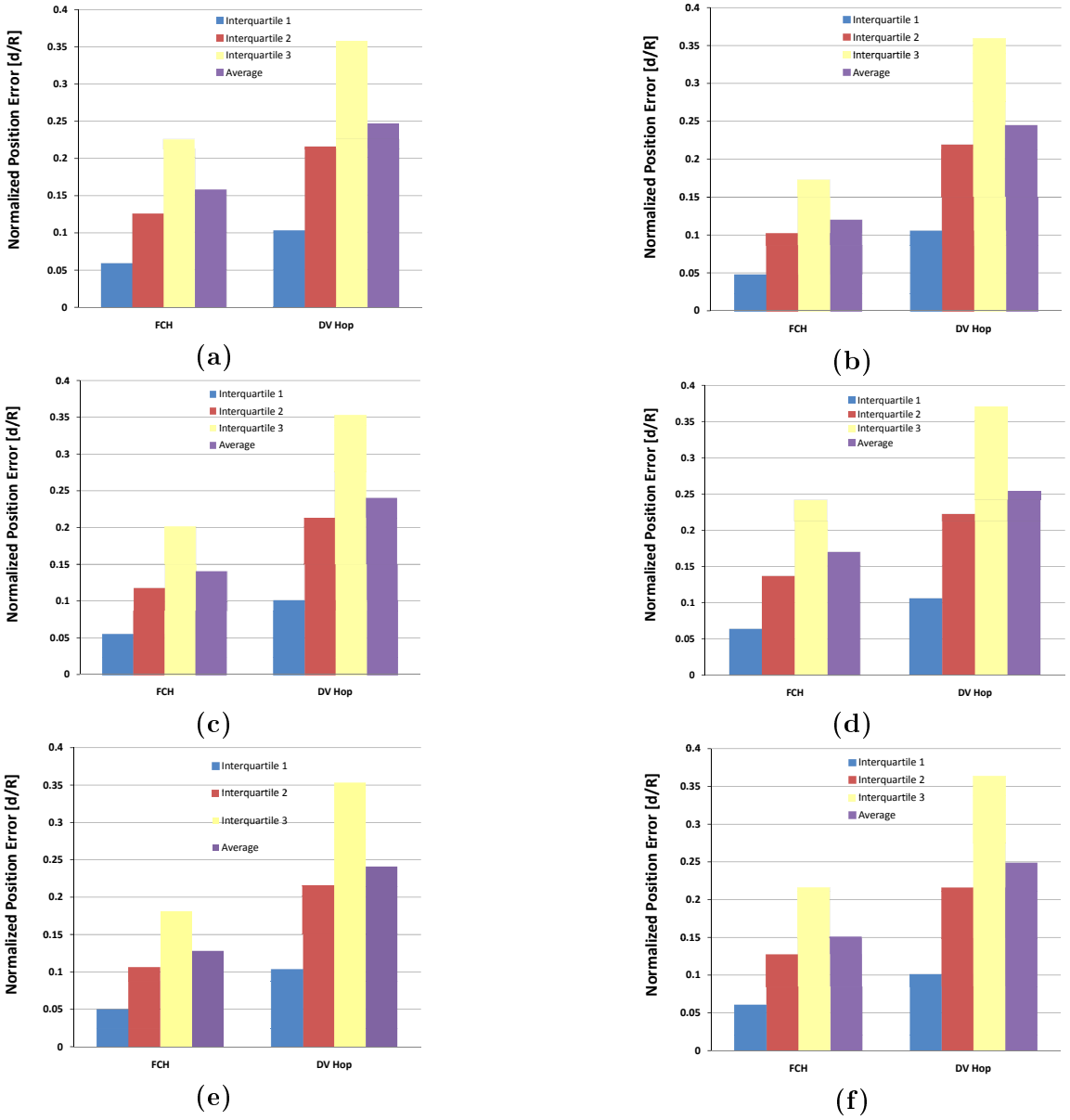


Figure 5.10: Comparison of the error resulting from the simulations of the FCH and DV Hop algorithms in dense networks with a) center b) mash c) corner d) middle e) half and f) slash near-uniformly distributions

from them. The critical drawback is that this refinement phase works reasonably well for a regular network and high connectivity but does not converge for other network topologies.

Although the Positioning Iterative Vector (PIV) algorithm proposed in this paper has been initially proposed as a new refined iterative method to locate the sensor nodes, we will investigate the feasibility to implement PIV as an autonomous distributed localization approach. The main idea is to present an algorithm which determines the position of sensor nodes relying only on local neighboring distance estimation within an ad-hoc space with sparse number of reference nodes, and at the same time, avoid the accumulative distance errors due to the snake paths. In the following sections, we will describe the mathematical origin of the PIV algorithm to later test its performance as an autonomous iterative positioning algorithm as well as refined system after a multilateration phase. Our findings are based on comprehensive simulations with uniform distributed networks as well as grid node deployment.

5.3.1 Mathematical Foundations of the PIV Algorithm

The available information that a given unknown node has to locate itself in a WSN before a locate-sensing process can be divide into three main features: first, the position of the reference nodes. This data can be obtained by direct communication with anchor or in an indirect manner by, for example, a flooding broadcast of the landmark. Second, the distance measurements between adjacent nodes obtained by the exchange of packages between neighboring nodes. Third, the superposition of these distances to find the shortest path between the reference nodes and a given unknown node as used in multilateration.

The idea behind the PIV algorithm is to work just with the first two parameters, considering the second parameter (the estimated distance between neighboring nodes) as a force vector that pulls and pushes the estimation of the node position iteratively.

At the very beginning of the localization process, every unknown node has a random position (we called this tuple as Virtual Coordinates). The nodes can calculate a virtual distance from its initial estimated position and its corresponding neighbor positions to later compare it with its measured or estimated adjacent node distances. The difference between the estimated distances and the virtual distances is used as a figure of merit to correlate the current node to its real position. To estimate the virtual distances between adjacent nodes, we used the DIN algorithm, but this distance can be obtained by other range-based methods such as RSSI, TOA or TDOA.

As can be expected, the position error is very high at the beginning of the localization process due to the random distribution. This disparity between the virtual distances and the estimated distances produces large changes in the node position estimations for the first iterations. As the PIV algorithm progresses, the changes in the node position estimations will be smaller compared to the initial iterations, and little by little the position estimation of a given node will converge to its real location.

The reference nodes in the network play an important role in the iteration process. Unlike unknown nodes, the landmarks do not change its coordinates, but they help and accelerate the process in every iteration to find the right position of the unknown nodes. In other words, ones realizes that the solution with the PIV algorithm is analogous to the system of

free masses (unknown nodes) connected by springs, where a few masses are fixed by nails (anchors or reference nodes).

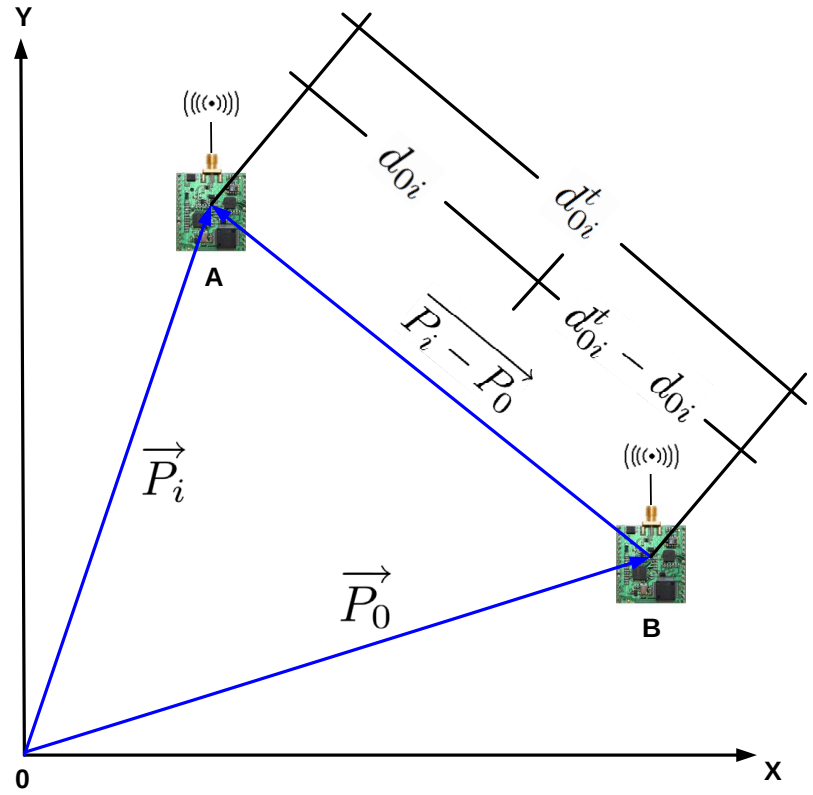


Figure 5.11: Mathematical Schema for two neighboring nodes using the PIV algorithms

To understand how the nodes move to its real position in every iterative step, we can consider two nodes A and B as shown in Figure 5.11. Here, their positions are regarded as vectors $\overrightarrow{P_0}$ and $\overrightarrow{P_i}$. Both nodes know after the process of distance measurements (in our particular case are the DIN distances) that the separation between them is the distance labeled as d_{0i}. Using the virtual coordinates resulting from the random deployment, the node B computes the euclidean distance to node A in the t-th iteration as

$$d^t_{0i} = \| P_0 - P_i \| \tag{5.7}$$

The difference between the distance d^t_{0i} and the estimated distance d_{0i} produces the distance error (ϵ) between the nodes

$$\epsilon = d^t_{0i} - d_{0i} \tag{5.8}$$

The absolute value of the normalization of Equation 5.8 in respect to the euclidean distance

d_{0i}^t is the magnitude of the error vector (α_{0i}^t) between the node A and B in the t-th iteration

$$\alpha_{0i}^t = \left| \frac{d_{0i}^t - d_{0i}}{d_{0i}^t} \right| \tag{5.9}$$

Equation 5.9 is the constraint value to move the node B to A with the direction of $\overrightarrow{P_i - P_0}$. Due to the fact that node A will move exactly the same stretch, but in opposite direction $\left(\overrightarrow{P_0 - P_i}\right)$, a factor of 1/2 is introduced in α_{0i}^t to avoid the overlapping of the distance correction (ϕ_{0i}^t). Node B will advance half of ϵ and the remaining half is covered by the node A in the same iteration as depicted in Equation 5.10.

$$\phi_{0i}^t = \frac{1}{2} \left| \frac{d_{0i}^t - d_{0i}}{d_{0i}^t} \right| \left(\overrightarrow{P_i - P_0}\right) \tag{5.10}$$

To find the position update for a given node with k neighbors, the average of all α_{0i}^t and its respective orientation vectors are taken into account. Thus, the new position of a node in the position P_0 for the $t+1$-th iteration with k adjacent nodes is

$$P_{0i}^{t+1} = P_0^t + \frac{1}{k} \sum_{i=0}^{k} \frac{d_{0i}^t - d_{0i}}{2d_{0i}^t} \left(\overrightarrow{P_i - P_0}\right) \tag{5.11}$$

Equation 5.11 is the generalization for the update of the components estimations for the X,Y, or Z axis. The process by which to calculate the update for the X-axis and Y-axis coordinates of an arbitrary node at the $t+1$-th iteration is introduced in Equation 5.12 and 5.13.

$$X_{0i}^{t+1} = X_0^t + \frac{1}{k} \sum_{i=0}^{k} \frac{d_{0i}^t - d_{0i}}{2d_{0i}^t} \left(\overrightarrow{X_i - X_0}\right) \tag{5.12}$$

$$Y_{0i}^{t+1} = Y_0^t + \frac{1}{k} \sum_{i=0}^{k} \frac{d_{0i}^t - d_{0i}}{2d_{0i}^t} \left(\overrightarrow{Y_i - Y_0}\right) \tag{5.13}$$

This completes the mathematical steps of the PIV algorithm, since every node in the network can now estimate its position iteratively with its local information by using Equation 5.12 and 5.13.

As we will see in the next sections, we have simulated the multilateration algorithms using real neighbor distances as well as the two mode of the PIV algorithm (Autonomous localization algorithm and refined algorithm). There, we have realized that the better the quality of the distance estimations does not mean necessarily the best position estimation using multilateration or the iterative location-sensing algorithms. This is mainly due to dependence of network factors such as the initial position that every node has at the very beginning of the iterative process, the number of iterations, number of hops inside the network, and some others that will be later go into detail.

5.3.2 The PIV Algorithm as Localization Algorithm

In order to discover the pure behavior of the PIV algorithm as a position-sensing algorithm under variable network settings, we performed the following simulations using C++ language. We decide to use C++ language instead of the `ns-2` simulator to present an idealized version of the network to compare purely the performance of our algorithms in this environment without communication protocol and/or the delivering of simulated packages. At the same time, we compared the simulations obtained using PIV and multilateration. The simulation runs are divided into two groups:

1. Simulations using random uniform nodes deployment
2. Simulations using grid nodes deployment

Both simulations groups take into consideration the following parameters: The simulation space where the nodes were deployed has a normalized dimension of $1x1$. The normalization of the area let us to later compare our results easily with other research works. The number of deployed nodes have been varied from values of 10 nodes until 100 nodes adding 5 nodes as discrete step for every simulation case. The normalized radio transmission range was changed from 0.1 until 1.0 with an incremental of 0.1. Every simulation case was tested with 4, 8 and 16 reference nodes. For all the possible simulation cases resulted by changing the number of deployed nodes, their radio transmission, and the number of reference nodes, we have plotted the average position error and its respective standard deviation using the next algorithms:

1. Multilateration using Real Neighbor Distances (RND)
2. Multilateration using the ExDIN algorithm
3. Multilateration using the DV-Hop algorithm
4. Multilateration using the FCH algorithm
5. PIV using RND
6. PIV using DIN distances

5.3.3 Localization methods comparison in random uniform nodes deployment

The first step in our analysis is to test the PIV algorithm as an individual positioning method and at the same time compare it with other approaches explained in the last section. Even though the PIV algorithm was initially developed to be applied as a refinement stage, it would be of great interest if this new method could also be used to find the position of the wireless sensor nodes in an iterative manner. The following simulations compare the performance of PIV using RND and DIN distances with the efficiency of the multilateration approaches. All the nodes in the network were deployed in a random manner over the normalized square of $1x1$. The virtual position that every single node had at the very beginning of the localization process was also uniformly distributed.

Figure 5.12 shows the average position errors and their respective standard deviation of different localization approaches obtained after an average of 100 simulation runs using 4 reference nodes. In order to highlight the strong advantage using the PIV algorithm as autonomous localization method we first refer to a drawback identified using multilateration.

Looking at the Figure 5.12 (a) where the average error utilizing the multilateration approach and the RND of the nodes is shown, we can observe that in spite of the fact that we used the real distances between adjacent nodes, the approach does not find the right node position in cases such as the combination of low density sensor networks and low normalized transmission radii. The main reason is the lack of sufficient links between the unknown nodes and the system's beacons. This is the reason that curves using 10 and 20 deployed sensor nodes do not appear after a normalized transmission radio of 0.5 or 0.4 respectively. Then, when the network has node deployment bigger than 55 distributed nodes and values of normalized transmission radius lower than 0.4, the multilateration method does not converge on the real node position. For the rest of the cases, multilateration using RND yields error values lower than 0.033.

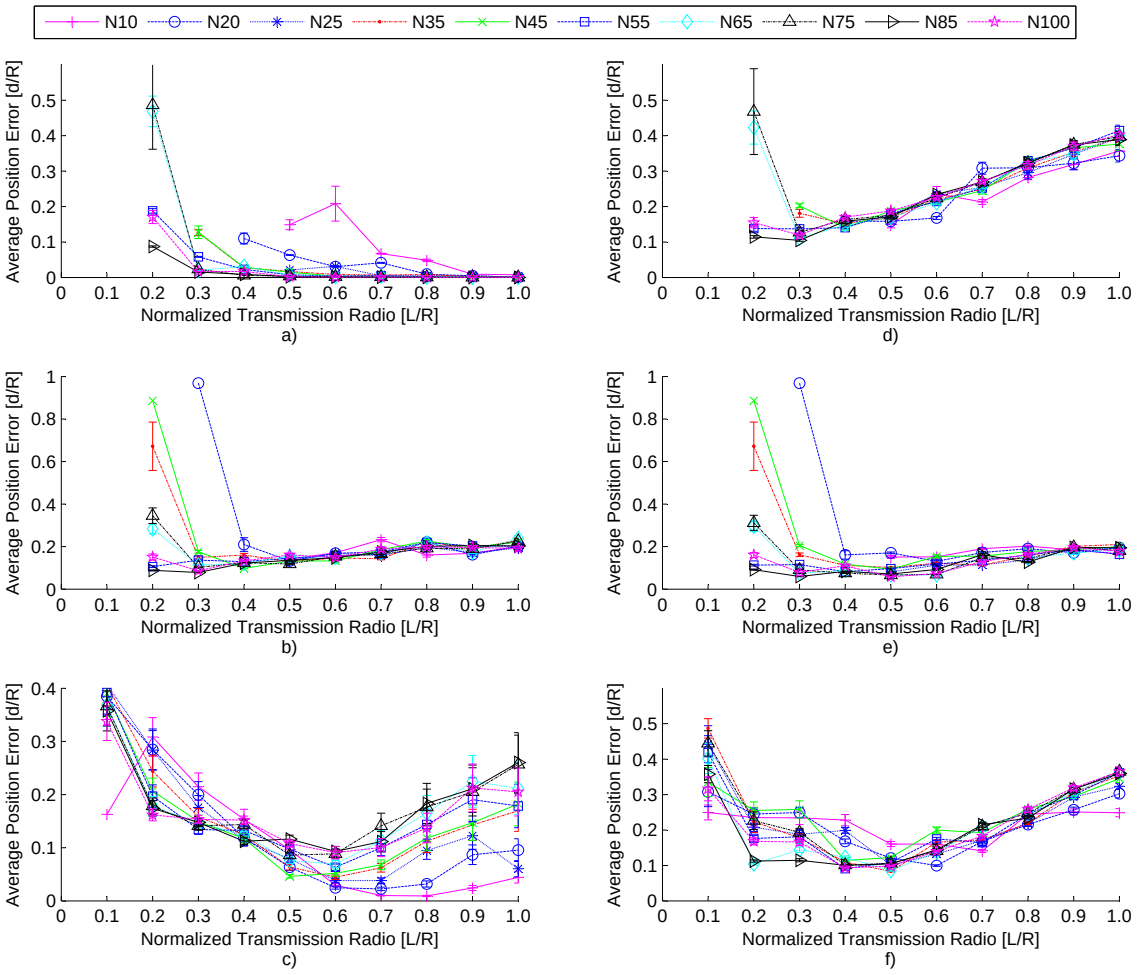


Figure 5.12: Position error comparison of a)Multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN distances for an uniformly distributed network and 4 Anchors

Unlike Multilateration using RND, the PIV algorithm also using RND (see Figure 5.12 c)is

able to locate the sensor nodes for all node densities and all radio communication ranges. For the cases where multilateration fails, PIV can estimate the position of some of the unknown nodes after 500 iterations. The PIV algorithm can overcame these cases because it just needs a single connection with a landmark to estimate the node positions rather than at least 3 beacon links to locate the nodes as in the case of multilateration. Of course, the position accuracy is better when more connections with the reference nodes are available, as can be deciphered in Appendix E Figure E.1 and Figure E.2, where the plots have been produced using 8 and 16 reference nodes.

A weakness of PIV used as localization system is the velocity of convergence. The number of necessary iterations needed to locate a node can be a potential problem. In Figure 5.12 c, we see the effect of slow convergence of the PIV algorithm. This disadvantage is presented for all the transmission radii. Looking closer into the run simulations, we realized that the initial random node positions plays an influential role in making the iterative process longer. After 500 iterations, the algorithm does not reach the real position of the sensor nodes, but at least produces a significant advance positioning nodes for very low node densities and low values of transmission ranges. As we can see in Figure 5.12 f, PIV did not overcome the good results obtained by using Multilateration with FCH with 4 reference nodes.

If we increased the number of landmarks in the system, then the accuracy of PIV became competitive to FCH, as can be seen in Appendix E in Figures E.1 f and E.2 f where the simulations have been accomplished by using 8 and 16 landmarks. For the plots obtained by using 16 beacons, the PIV overcame the error position values of all the approaches producing average position errors lower than 0.04 for network densities bigger than 75 deployed nodes in a range of 0.2–04 tranmission radii. We also observe that PIV with RND begins to reach the right node positions over a transmission range of 0.3 and 0.7 values in all the network densities, something that RND multilaterion fails to do as well (compare Figures E.1 a and c).

PIV can estimate the position of sensor nodes using DIN distances with a good execution for radius values between 0.3 and 0.7 as depicted in Figure E.1 a. Different from the PIV algorithm, the combination of multilateration and the ExDIN distances (Figure 5.12 d) has its best performance for a radius value of 0.3 getting average position error lower than 0.25 for all densities. For bigger radius values, the ExDIN algorithm begins to lose accuracy due to the accumulative landmark to node distances as well as the errors produced by DIN during the neighbor distance estimations. Algorithms like FCH and DV-Hop working with correction factors have the advantage to make up for the accumulative errors over the wide radius range. We can observe in Figure 5.12 b and Figure 5.12 e that the FCH method works slightly better than the DV-Hop approach. FCH effectively compensates the position errors for medium and high network densities yielding values lower than 0.1 in a radius range between 0.2 and 0.6. On the contrary, DV-Hop can produce similar error values for 0.2 and 0.3 radius values over 55, 85 and 100 deployed nodes in the network.

A conclusion of using PIV as localization system is that the average accuracy degrades slightly for networks with higher number of deployed nodes. This tendency in the algorithm behavior is depicted more clearly in Figure 5.13 c where the average position errors of a number of 100 simulations runs using PIV with RND is shown. Here, we can see that for normalized radii bigger than a value of 0.4, the curves tend to slightly misplace the sensor

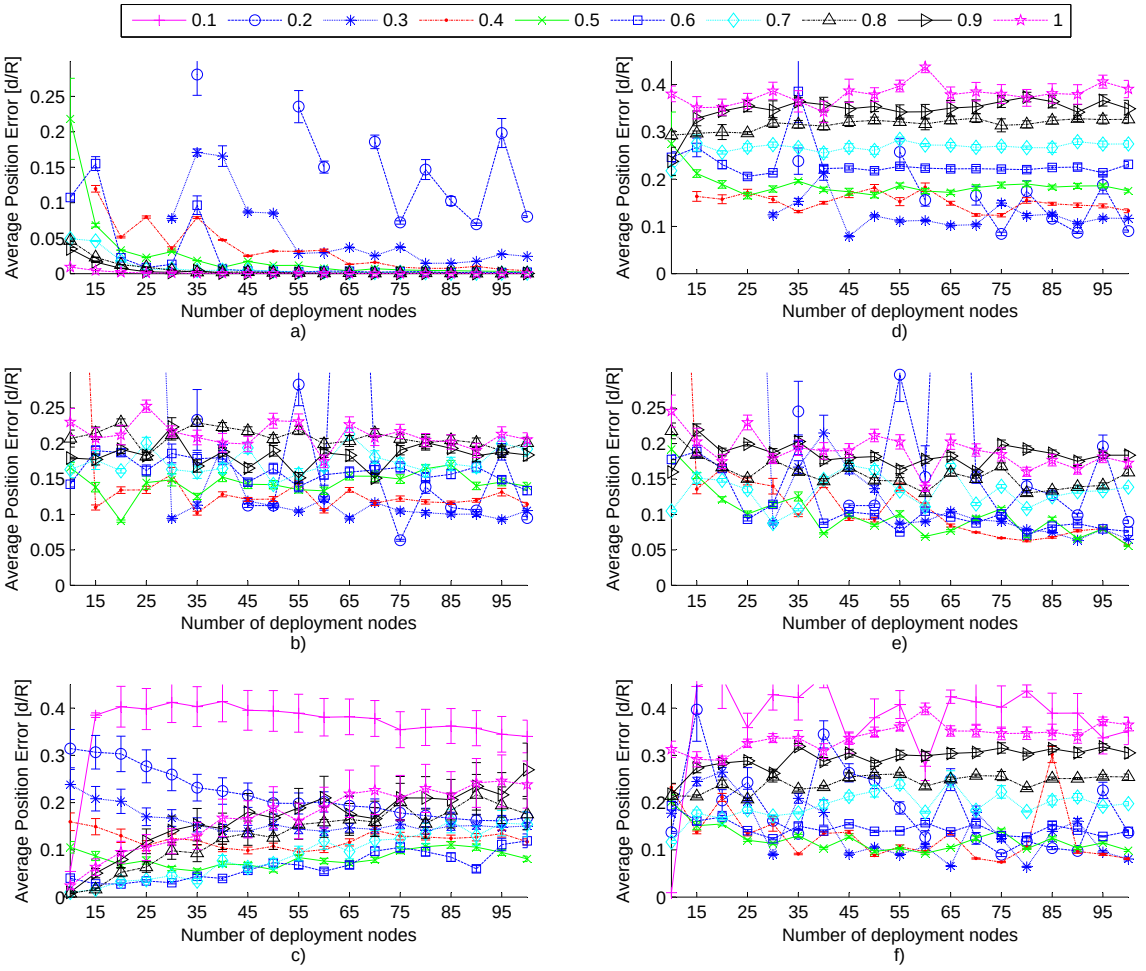


Figure 5.13: Comparison of average position error versus number of deployed nodes with a)multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN Distances for an uniformly distributed network and 4 Anchors

nodes when the number of deployed sensor nodes is increased. Using the DIN Distances, the PIV algorithm presents the same effect as can be seen in Figure 5.13 f. The loss of accuracy of PIV for a large number of sensor nodes is a direct consequence of the velocity of convergence as will be further described in the next section.

An alternative to alleviate this loss of accuracy using PIV for high node densities is incrementing the number of reference nodes in the system. Incrementing the landmarks in the system, PIV produces better position accuracies using either RND or DIN distances, as can be seen in Figure E.3 and Figure E.4 letters c and f. Unlike PIV, the multilateration methods with all the approaches variants maintain relative stable or improve slightly the position errors for all the cases where the algorithm converges. The plots in Figure 5.13 show that multilateration develops the best position estimations using radius values between 0.4 and 0.6, while the PIV algorithm is specially precise for radius values from 0.3 to 0.6 using

DIN neighboring distances. This general trend applies to all the node densities and for all number of reference nodes as can be seen in Appendix E looking at the curves of Figures E.3 and E.4. The behavior of PIV can be attributed to the fact that the best distance accuracies produced by the DIN algorithm also have the best performance for almost the same radius ranges (see chapter 4). Once again, the overall curves of those figures point to the fact that the PIV algorithm yields position estimation using all the radii and over all network densities.

5.3.4 Localization methods comparison in grid nodes deployment

An important point of reference commonly used by the research community is the behavior of algorithms tested in a grid setup. The subsequent simulations resulted by using a grid node distribution over a normalized square deployment area. Similar to the positioning error simulations presented in the section 5.3.3, the number of nodes was varied adding 5 nodes from an initial number of deployment nodes of 10 to 100 nodes. The nodes were placed on 5 columns separated by a fixed distance of 0.16666 units over the x-axis. The number of rows depended on how many nodes the simulation run had. The criterion handled to distribute the rows over the y-axis were determined by the next expresion:

$$Y_{RowsDist} = \frac{1}{\frac{TN}{5} + 1} \tag{5.14}$$

where $Y_{RowsDist}$ is the distance between the node rows (constitued by groups of 5 nodes) and/or the borderline of the normalized square area. TN is the total number of deployed nodes in the system. Once more, the virtual initial position of every single node in the network were placed in a random manner over the square area.

The simulation results using 4 reference nodes in the grid setup can be seen in Figure 5.14. There, it continued to be noted that in all cases using multilateration with DIN distances (letters b, d, and e) the grid configuration had a negative effect on the positioning errors for higher radio transmission compared to the perfomance yield in a uniform node deployment (see section 5.3.3). On the other hand, the PIV algorithm remained in the same range of accuracies as presented in the case of a uniform distribution shown in the last section. However, the node position estimations using PIV continued to be slightly higher as the best results produced by the multilateration and the FCH algorithm.

Comparing the results using the grid and uniform node configurations, we observed that the nodes in the grid setup was better able to locate nodes using the combination of multilateration algorithms in comparison to those cases presented in uniform distribution with low density networks ([10–30] deployed nodes) and low radio transmission ranges ([0.1–0.3] transmission radius). The curves with low node densities and low radio communication links appeared later in the uniform distribution than the grid setup. In other words, the curves with these properties were plotted for higher radio transmission radius than those used with the uniform configurations (see Figure 5.14 and 5.12 letters b, d, and e).

This effect in the uniform distribution is produced due to the requirement to have bigger values of transmission ranges to create direct links with the landmarks or indirect connections through other neighboring nodes, thereby enabling them to find their position in the system.

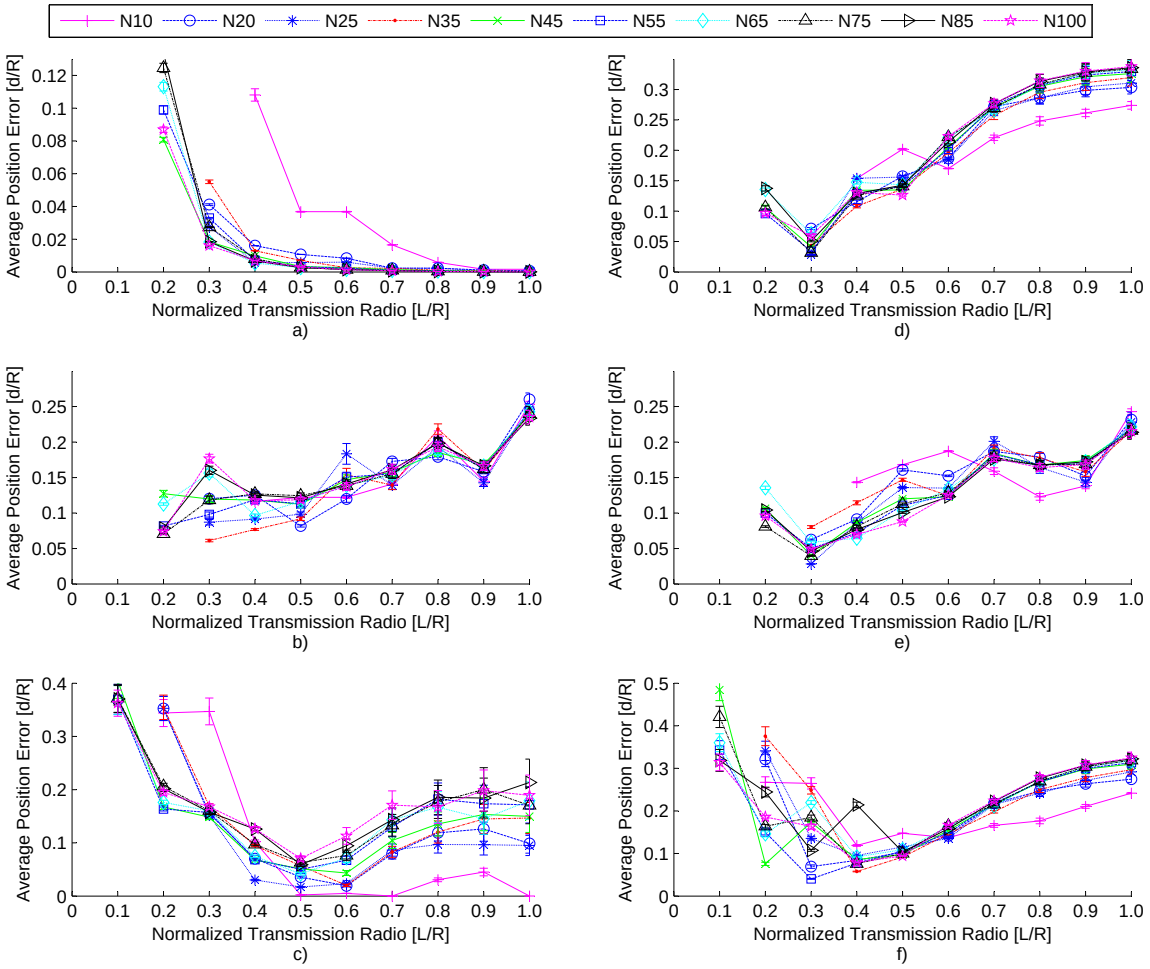


Figure 5.14: Average position errors versus normalized transmission radio using a)multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN Distancces for a grid node distribution and 4 landmarks

On the other hand, the fixed node position in the grid deployment have helped the nodes to connect to each other with low radio ranges and low node densities.

Although the PIV algorithm is capable of estimating node positions where some of them have either direct or indirect landmark contact, the nature of the grid setup negative influences the performance of PIV due to the lack of connections to the landmarks, when using radio transmission ranges with a value of 0.1 and low density networks producing node clusters in best cases (as can be seen in Figure 5.14 a and f). Once again, multilateration is the most affected because of the requirement to obtain at least three node to landmark distance to locate a given node.

The impact of incrementing the number of reference nodes for PIV and the several combinations with the multilateration method can be also seen in the Figures E.5 and E.6 where the plots using 8, and 16 anchors depicted an improvement over all the position errors and

all the localizations algortihms. An inspection of the curves resulting from using the PIV algorithm reveals the usage of different radius values to locate nodes with competitive error values increases when the number of reference nodes are augmented. Such a behavior is also confirmed by the plots of Figures E.7, E.8, and E.9 where the curves show the average position errors using a given radio transmission range versus the number of deployment nodes for 4, 8 and 16 anchors in the system respectively. Looking closer at the curves produced with the PIV algorithm and the DIN neighboring distances, we note that the more number of anchors at the perimeter of the square area, the better accuracies obtained for the different radius curves. The best improved curves correspond to values between 0.2 and 0.7 radius similar to the simulations conducted with uniform distributions.

5.3.5 Velocity of convergence using the PIV algorithm

An interesting question about the performance using the PIV algorithm is how many iterations are necessary to develop an acceptable node position accuracy. To answer this interrogation, we studied once more the results of the average of 100 run simulations in uniform and single simulations with a grid distribution. The same network parameters used in section 5.3.2 were utilized in our search for this section. Certainly the figures belong exclusively to the position errors calculated by using PIV with real neighbor distances (all the curves placed on the left side) and PIV using DIN neighboring distances (curves positioned on the right side of the figures).

PIV in node uniform distribution

First, the plots that show the average position errors versus iteration, with variation in the number of deployment nodes, are depicted in Figure 5.16 and Figure 5.16 using 4 reference nodes. An inspection of the curves of Figure 5.16, where we programmed in low radio transmission ranges, shows that the change of node position estimations for the PIV algorithm using RND is high before reaching an iteration value of 50. In this stage of PIV, the algorithm achieves a noticeable reduction of the average position errors.

After this iteration value, 80% of the average curves are in the area where a low location improvement ratio is reached. On the other hand, the ratio of convergence of the PIV algorithm using DIN neighbor distances is more affected by the size of the radio transmission range used by the sensor nodes, producing a delay affect for the low position improvement stage reached after 70 iterations.

Observing the curves using RND, we can concluded that the lower the node density the network had, the faster it is reached the zone where the average position error is slowly improved. As better displayed in Figure 5.16, the curves with slow densities converge faster than the curves with a high number of deployment nodes for PIV with RND as well as PIV using DIN distances. However, that does not necessary means that the curves have the best accuracies.

It is worth highlighting that PIV produces a satisfactory degree of accuracy for low and medium network densities as can be seen in Figure 5.16 where the curves using 0.5, 0.7, 1 radio transmission values are depicted. Comparing the curves obtained using the RND to

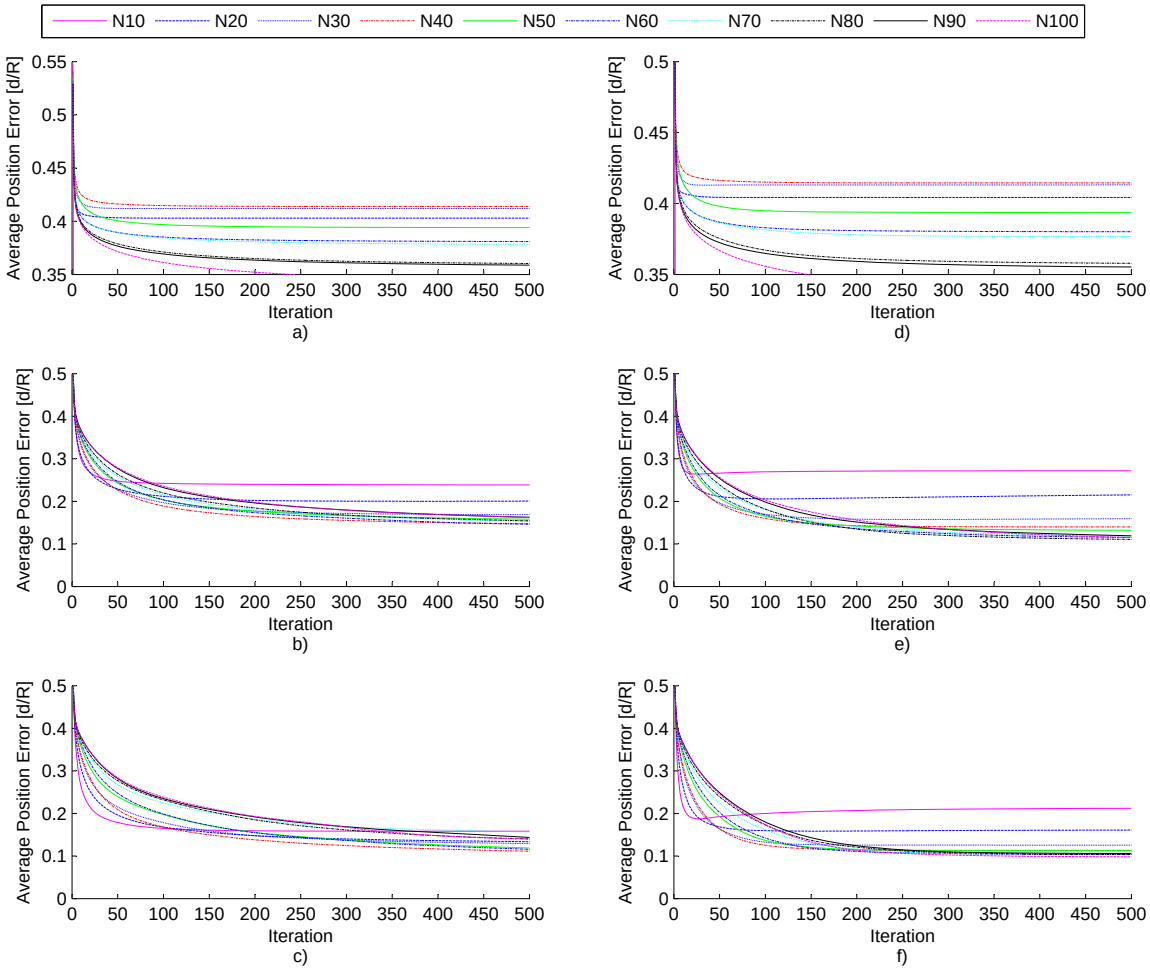


Figure 5.15: Average position errors versus iteration using 4 reference nodes with a)0.1R and RND b)0.3R and RND c)0.4R and RND d)0.1R and DIN distances e)0.3R and DIN distances f)0.4R and DIN distances

those calculated with DIN, the algorithm converges to the values established by the distance model. That is to say, the curves display a reduction in the average position error for all the radius and density values theoretically until they reach the right node position after a large number of iterations. In the systems where the DIN distances are used, the PIV algorithm tried to find the best node location that minimized distance errors in the neighboring node distance model. This behavior is clearly depicted when we compare Figure 5.16 c and f.

The curves of Figure 5.16 tended to improve the position of the nodes slower if the radio transmission was incremented. The slowest values of convergence were for the curves with high node densities and high network densities.

To better understand the impact of changing the node communication radius on the velocity of convergence of the PIV algorithm, the curves of Figure 5.17 and Figure 5.18 were simulated for different network densities using 4 reference nodes in a uniform network distribution.

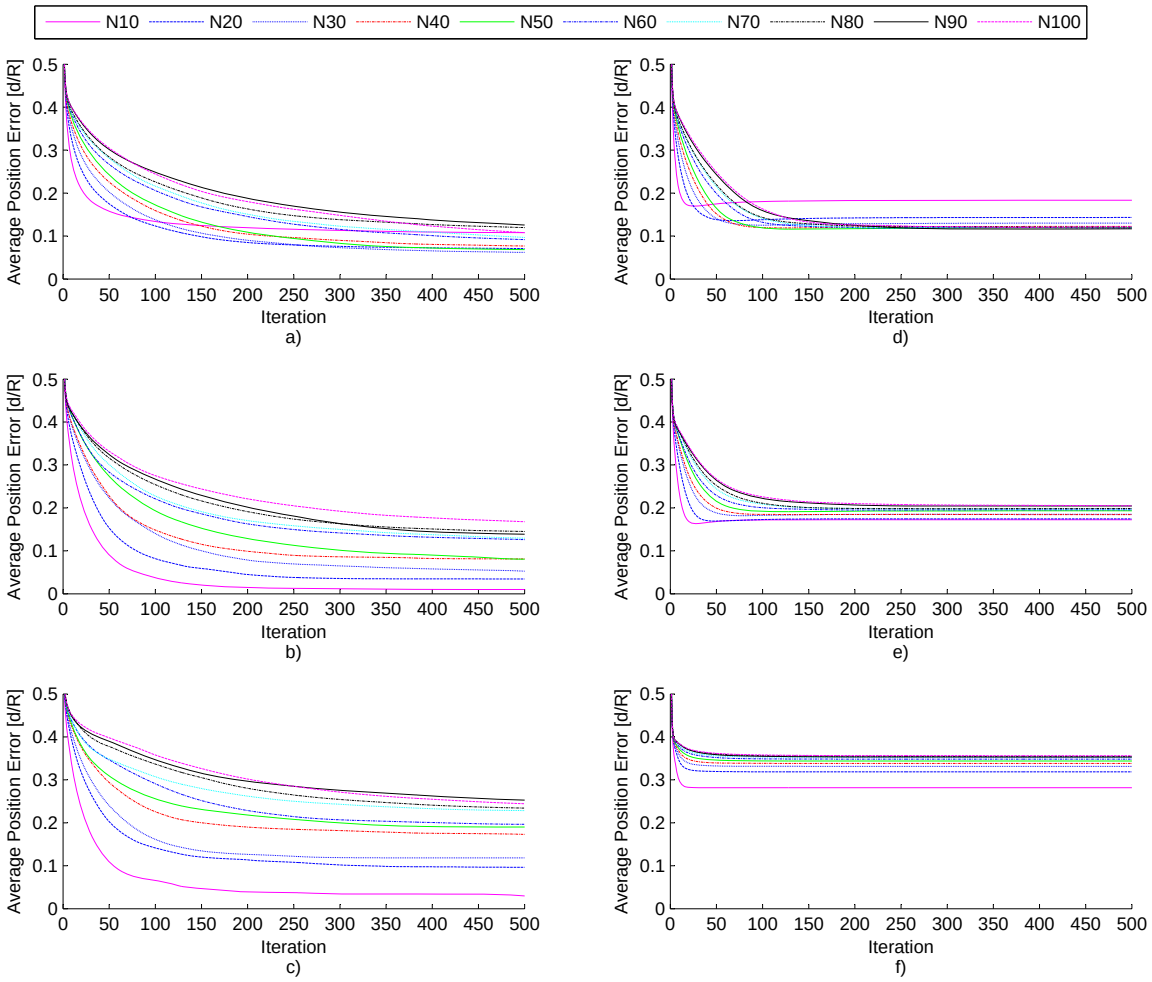


Figure 5.16: Average position errors versus iteration using 4 reference nodes with a)0.5R and RND b)0.7R and RND c)1R and RND d)0.5R and DIN distances e)0.7R and DIN distances f)1R and DIN distances

Once again the average of 100 simulations are presented on the values of the position errors. Both figures show clearly that the most efficient radio transmission values for the PIV algorithm using either RND or DIN distances are in the range of 0.2 to 0.7 (as we also see in section 5.3.3). The curves that present fewer position accuracies are the first to converge on the are of slow error improvement, since most of the cases radio transmission values between 0.4 and 0.7 yield better position estimations but with a slow ratio of convergence.

The convergence of the PIV algorithm is directly related to the number of deployed nodes in the network. Looking closer at the curves produced by PIV using DIN Distances and RND distances from Figures 5.17 and 5.18, it is easier to calculate how the knee of the curves with different node densities is displaced to higher iteration values. This can be explained as follows: the nodes that are closer to landmarks have the advantage of becoming quickly its virtual position to the real one since they used the fixed position of the reference node

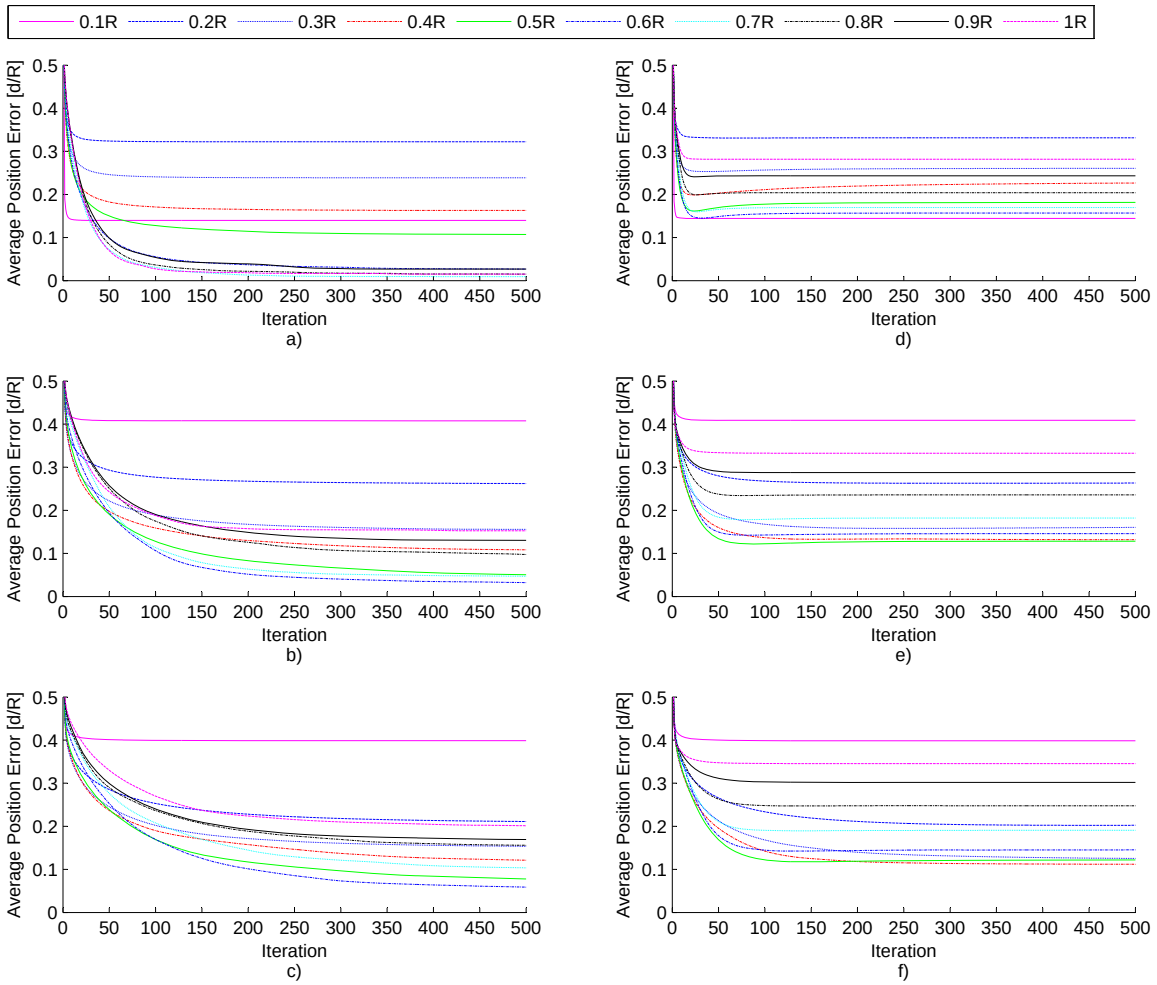


Figure 5.17: Average position errors versus iteration using 4 reference nodes with a)10 deployed nodes and RND b)30 deployed nodes and RND c)50 deployed nodes and RND d)10 deployed nodes and DIN distances e)30 deployed nodes and DIN distances f)50 deployed nodes and DIN distances

as well their comparisons of neighboring distances, as explained in section 5.3.1.

In constrast to those unknown nodes, the nodes that are placed deeper in the network have just as accurate information the comparison to their virtual distances (that are not fixed) and their estimated distances with their adjacent nodes. Thus, these nodes need more iteration to place themself in the right position, since such nodes depend on the iterative position correction of their neighbors, the accurate distances from their neighbor distnace estimations and the initial virtual positions of their neighborhood.

The number of deployed anchors is a critical network parameter for the velocity of convergence of the PIV algorithm, as can be seen in the of appendix F where the position error curves using 16 reference nodes are shown. looking closer at the curves of Figure F.3 and F.4 we can observe that all the convergent knees of the whole plots using PIV with DIN distances are placed before the iteration number of 45. The best value was obtained

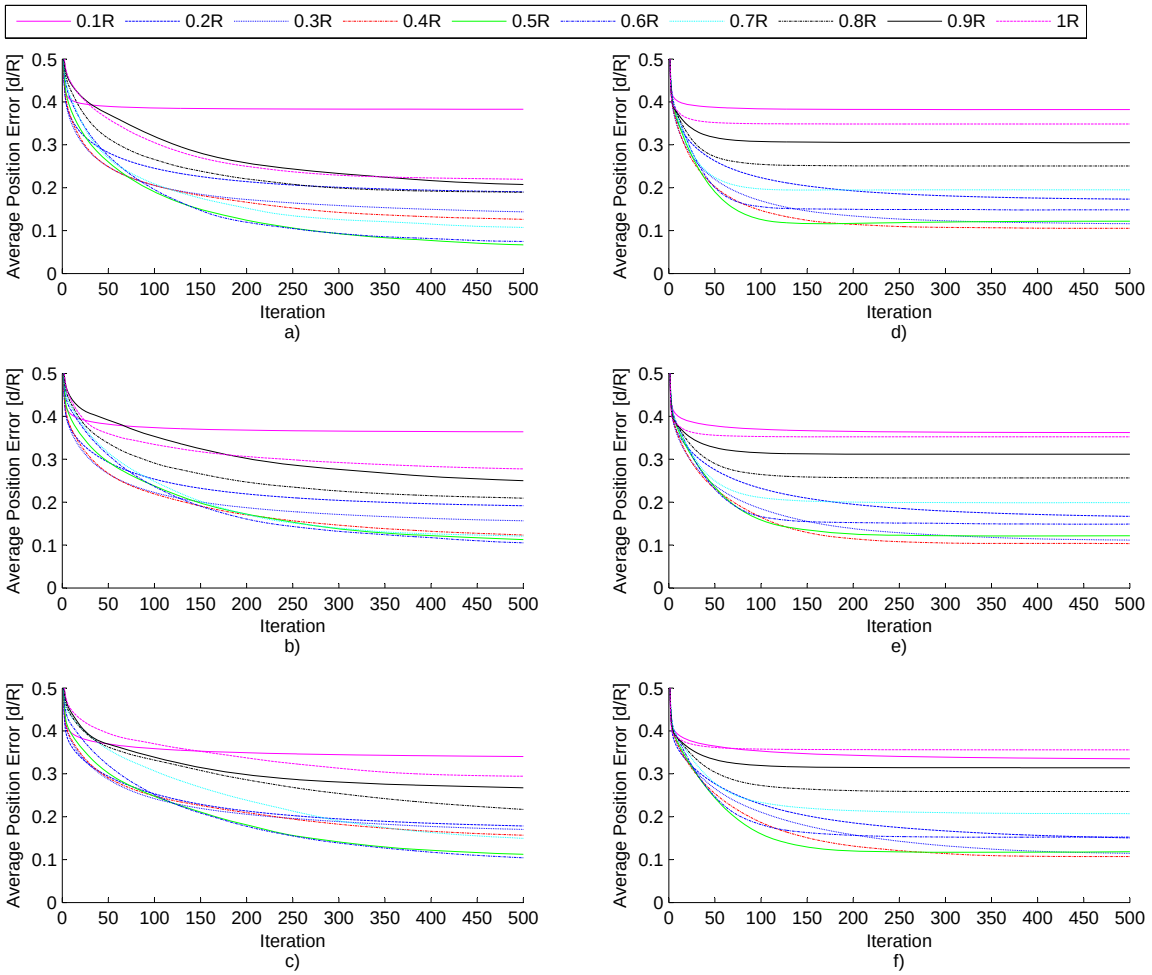


Figure 5.18: Average position errors versus iteration using 4 reference nodes with a)60 deployed nodes and RND b)80 deployed nodes and RND c)100 deployed nodes and RND d)60 deployed nodes and DIN distances e)80 deployed nodes and DIN distances f)100 deployed nodes and DIN distances

for very low node densities [10-20 deployed nodes] where the PIV convergence is lower than 13 iterations (see Figure F.3 d).

5.3.6 Discussion of the PIV algorithm behavior

In the last sections, I analyzed the behavior of the PIV algorithm and a comparison with algorithms based on multilateration. As general conclusions about the perfomance of the PIV algorithm, I must mention the following points:

The PIV algorithm develops similar or slightly better position accuracies than the ones produced by multilateration for low and medium network densities with normalized radio transmission ranges between 0.2 and 0.7 using a low number of reference nodes. If the number of reference nodes are incremented (at least larger by 8 anchors) then the PIV algorithm could be utilized as a refined method after the first phase of a given localization

process (initial position estimations such as multilateration in our case). The PIV algorithm as an autonomous localization method has the advantage of locating sensor nodes with the sole help of a unique reference node. The low converge ratio achieved by PIV for high network densities constitutes a significant drawback for PIV to be a suitable localization system in a real testbed scenario due to the relative high communication cost when the neighboring distance are determined by DIN. An open research topic of the PIV algorithm is to determine the behavior of the algorihm using other signal technologies comparing the results once again with other localization approaches using the same network conditions.

5.4 DIN Based Localization: Experimental Evaluation

The attenuation of radio signal with distance and symmetric radio connectivity are common assumptions in wireless network research. As mentioned in chapter 1, those theoretical radio properties do not fit real scenarios and true hardware platforms. Due to the good results of the FCH algorithm obtained for different network distribution, we decided to put into practice the FCH algorithm on small-wireless sensor devices. At the same time we pretended to compare these results with other approaches such as DV-Hop, two RSSI based location methods and the PIV algorithm. The main goal is to measure the impact of a real environment on the performance of all these approaches. To the extent of our knowledge, the present study constitutes the first performance comparison of several range-based and range-free localization algorithms implemented on real hardware.

5.4.1 Adaptation of DIN Based Localization for experimental Evaluation

We used the ScatterWeb MSB platform to set up our testbed. The same RSSI threshold with a value of 33 (-42.5 dBm) characterized in section 4.3.2 was taken to preserve the artificially transmission range of 3 m.

The general communication protocol of each testbed experiment can be divided into three main stages, which loosely correspond to the constituent execution phases in which the localization algorithms have been structured throughout the present thesis:

1. Internodal ranges estimation, either via the DIN algorithm or by direct application of one of the two considered RSSI-ranging functions.
2. Hop-by-hop node-to-anchor extended ranges estimation (applying DV-Hop, ExDIN, FCH or the RSSI-ranging functions) and initial position determination via multilateration.
3. Refinement of estimated node locations using the PIV algorithm (the location improving is solely applied to the algorithms based on the DIN distance estimations).

In phase one, every node in the network sent a HELLO packet to discover neighboring nodes within its communication range. The transmission range determined by the RSSI threshold valued at -42.5 dBm (number 33 in the CC1020 communication chip) plays an important roll in this phase. The nodes drop out a HELLO packet when they receive the transmission signal with a RSSI value lower than 33, thus only nodes whose transmitted package are received with RSSI values equal or stronger than -42.5 dBm are considered as neighbor

nodes. At the end of this first phase, every node in the network has already constructed its neighbor table.

Unlike in chapter 4, the broadcast was initialized by an external master node which activated the first broadcast of the lowest ID node, which later followed the next node broadcast utilizing a round-robin procedure based on the node IDs. The use of such a succesive communication protocol has the advantage of minimizing the packet collision.

In order to compute the DIN distances between adjacent nodes, the exchange of neighbor tables takes place during the first phase in the localization protocol. To minimize signal collisions in the network, the same strategy was used by acquiring the neighbor tables. That is to say, the neighbor tables interchange with the nodes in a round-robin broadcast depending on the ID number of every node.

When a node acquires all the neighbor tables in its neighborhood, the node can estimate the internodal distances applying the polynomial approximation of Equation 4.10 of chapter 4. Finally, this normalized distance (d_n) is converted into absolute units (meters) multiplying d_n by the known transceiver communication radio of 3m previously determined in section 4.3.2.

Every node then automatically identifies the communication asymmetries the network in the following manner: if a given node in the table interchange process has received a neighbor table from a neighbor node (previously registered as neighbor in the node ID broadcast stage) without its own ID, then both nodes have an assymetrical link. To properly compute the K_i and K_u factors in these cases, the passive node inserts its own ID on the heard neighbor table to later calculated the normalized neighbor distance by the DIN algorithm. A second possible asymmetric case might arise if the node received a neighbor table of another node that was not registered as a neighbor. In these cases, the neighboring distance is established with the maximum value of 3 meters.

In the second phase of the localization protocol, the nodes discovered their distances to the landmarks. The communication protocol implemented for the DV-dist algorithm begins at the reference nodes which sends a distance packet with the anchor's ID together with the hop count and the cumulative distance fields (set to 0 at the very begin). It is important to consider that only neighboring nodes can spread out the landmark-packets. In other words, if a node broadcasts the anchor message to other nodes around it, just the receiver nodes which have the sender node as a neighbor can validate and retransmit this packet through the system. In every retransmission, the unknown nodes add their DIN distances to the anchor message, thus the nodes of the next hops decide depending on those distances which node connection is the shortest path to the reference node.

Once again, the transmission of every anchor packet was implemented similarly to the DIN procedure using the round-robin scheme based on the anchor's ID. The process of the anchor package flooding begins with a unicast of an external node to the lowest ID landmark. The beacon with the lowest ID sends its anchor package into the network after receiving the starting signal from the external node and initializes a delay timer. After the expiration of the timer, the landmark transmits a control packet to the subsequent reference node. Once the anchor with the highest ID value has propagated its own packet throughout the network, then it sends a control packet to the external node, finishing this process.

After the execution of the DV-dist procedure, a set of minimum DIN distances and hops

to the reference nodes is available in every deployed node as well as at each landmark of the system. With this information, every anchor computes the FCH correction factor and sends this data through the network. The FCH corrections packets are transmitted by the reference nodes utilizing the same round-robin process as DV-dist.

Unlike the ExDIN process, the FCH process does not send back the correction packet in a hop bv hop fashion, but through a broadcast signal with the maximum transmit power of the landmark's transceiver. Due to the relatively small dimensions of the deployed area (8x9m) the correction factor is received by all the unknown nodes with a unique broadcast saving energy in the scatterweb nodes.

Whenever a node has the available correction factor of a given reference node, the node can update or correct its corresponding anchor distance estimation using the correction factor together with the number of hops and the ExDIN distances previously obtained in the DV-dist process. When an unknown node has at least three coordinates of different reference nodes and their correction factors respectively, the node computes its own location by multilateration.

The final phase of the localization process is the refinement of the estimated node locations using the PIV algorithm. The execution of every iteration can be divided into two subphases: First, the exchange of the position estimations of the nodes in the local neighborhood, and second, the computation of the estimated node location using the data of the first subphase.

The iterative process begins once more with the transmission of a control packet from the external monitor node to the reference node with the lowest ID number, thereby activating a round-robin communication scheme. After receiving the control packet, every node (including the landmarks) transmits a local broadcast which contains the current position estimation and its own ID. To avoid packet collision, a delay timer is activated on every node after its local broadcast to later transmit a PIV control packet to the next sensor node once the given timer has come to an end.

The iteration of the whole network terminates when the highest ID sensor node sends its own local broadcast and its delay timer expires. Then the last node indicates to the external monitor node the end of the round-robin communication process. The monitor node sends back a packet with the highest power transmission to begin the second subphase of the system. This transmission possible allows every single node to compute its new position estimation using the data collected in the first subphase and preparing the node for the next network iteration.

The unique nodes that do not change their own positions are the reference nodes. It is important to mention that all unknown nodes begin with the position estimation generated by the FCH, DV-Hop or DV-dist algorithm. A number of 30 iterations has been established in our different testbeds.

Furthermore, we considered use of two different RSSI based algorithms in our experimental study to compare the efficiency of the different algorithms implemented in our nodes. The first approximation is the same used in section 4.5.2. The second approximation function to derive distances from the RSSI values is the proposed in [25]. The resulting expression of the different RSSI linear combination and interweaving RSSI values of this work is presented

as a polynomial function:

$$f(x)_{RSSI_2} = f(x)_{indoors\_tx1} = -0.2996x^2 - 1.407x + 33.7234 \tag{5.15}$$

For the distance determination using RSSI values, no RSSI threshold is applied in any of the node configurations. That means, whenever a packet arrives with a signal strength large enough to be processed then, the RSSI value is substituted on the variable x of the two RSSI functions to compute the distances.

The position estimations produced for our algorithm are compared with the DV-Hop range-free technique. Due to the fact that FCH use the hop count between nodes and landmarks, the protocol of DV-Hop is easily implemented with the protocol as the FCH algorithm. Two sole restrictions have been programmed on the sensor nodes to estimate their location through the DV-Hop algorithm. The first requirement is that after the broadcast of the average hop values by every anchor (this broadcast is sent with the same packet of the factor correction for the FCH algorithm), the unknown sensor nodes must have registered its hop-distance to the respective landmark obtained in the phase of ExDIN. Otherwise, the distance to the anchor node is set as unreachable and its respective incoming calibration factor is ignored. The second condition requires that all of the average hop computed by the landmarks and later sent out in the broadcast package into the network has to be a positive value. If the received estimated hop value contains a negative magnitude, the node identifies the packect as corrupted and drop it.

When an unknown node has already at least three average hop values then the node is able to estimate its own position using multilateration. If the node received subsequent packets with new calibrations factored from other beacons, the node will refresh its own position using multilateration, adding the additional information.

5.5 Experimental Results and Testbed Set up

The results of our simulations in sections 5.2 and 5.3.2 were very promising; therefore we decided to implement all the localization algorithms explained in previous sections on the scatterweb sensor nodes. We use nine different testbed layouts in our experiment. The different configurations can be divided into three groups: the grid node deployment named uniform distribution, the configuration that depicts a horseshoe layout, and an assymetric low node density setup. For the uniform and horseshoe layouts, we settled on 50 and 100 nodes for every case. The nodes were deployed in an $8x9$ meter seminar room at the Free University of Berlin. They were placed on desk and the room was cleared over desk height.

The configurations were not just varied in the number of deployed nodes, but also with the number of reference nodes using either 4 or 8 anchors for every single configuration. The last layout named as asymmetric with very low density was populated with 9 unknown nodes and 8 landmarks. The setups using 8 beacons using 50 nodes and 100 unknown nodes for uniform and horseshoe configurations as well as the low density distribution are depicted in Appendix G.

The main results are shown in Table 5.2 and Table 5.3 where the average position error and the standard deviation values associated to the uniform distribution and horseshoe

distribution respectively are summarized. At first inspection of the mean values of both error position tables, we realize that the RSSI-based location approaches yield satisfactory values on position accuracies. The straighforward mapping of the signal strength to node-to-anchor distances and the use of multilateration have been produced a considerable improvement in location estimation compared to the node ranging values of chapter 4.

Another interesting point is the fact that the number of interweaving RSSI values used for the creation of the RSSI polynomial functions do not represent a big impact on the localization accuracies as can be observed by comparing the average values of the RSSI1 and RSSI2 positioning functions in uniform distribution as well as for the horseshoe configuration.

The best position estimation using a uniform distribution has been obtained with the ExDIN/DV-Dist algorithm with 4 reference nodes and 100 deployed nodes resulting in an average position error of 2.02 meters with a deviation standard of ±1.00 meters. Comparing this error value to the best performance obtained using the RSSI-based approach, an increment of 10% has be achieved by the ExDIN algorithm. The efficiency of the FCH algorithm has overcome almost all the averages error values produce by the DV-Hop algorithm. For the testbed where 50 nodes and 4 anchors have been deployed, the DV-Hop has obtained an very slight improvement compared to FCH. A comparison of the FCH average errors to those generated by the RSSI function shows that FCH has reached an accuracy improvement of 12% using 100 unknown nodes and 4 reference nodes. For the remaining node configurations in uniform distribution, the mean position estimations using the RSSI-based algorithms have been better than the ones yielded by the FCH algorithm. Although the best performance for the uniform distribution was provided by the ExDIN/DV-dist algorithm, it was not able to provide better position estimations for all node setups.

Table 5.2: Average position errors and their respective standard deviations in uniform configuration

Testbed Config.		**Absolute Error [m]**	***DV-Hop***	***ExDIN/DV-Dist***	***FCH***	***RSSI1***	***RSSI2***
4A	*50N*	*Mean*	2.6248	2.4281	2.6718	2.4474	2.3161
		Stdv	1.4216	1.1476	1.3093	0.9212	0.9475
	100N	*Mean*	2.4288	2.0269	2.1444	2.5323	2.4487
		Stdv	1.1540	1.0020	1.0171	0.9866	0.9592
8A	*50N*	*Mean*	2.4137	2.5592	2.3353	2.3463	2.2856
		Stdv	1.0455	1.373	1.2232	0.8710	0.8405
	100N	*Mean*	3.1027	2.5686	2.4894	2.3512	2.2388
		Stdv	1.3487	1.1891	1.2833	0.9798	0.9844

The best location estimation using the horseshoe configuration was yielded once again by the ExDIN/DV-Distance algorithm in the setup using 8 landmarks and 100 unknown nodes with a position accuracy of 1.87 meters and a deviation standard of ±0.85 meters. These results are a 15% improvement over the best location estimation produced using the RSSI technique. Analysing the results using 4 anchors with the horseshoe distribution, we observe that in all cases the best mean position has been accomplished by the ExDIN/DV-Distance algorithm. Whereas using 8 landmarks the FCH algorithm produces the best accuracy for the layout with 50 sensor nodes with an average value of 2.16 ± 1.16 meters. For the

configuration with 100 deployed nodes the ExDIN algorithm has the best value over all the cases using the horseshoe configuration.

The accuracy enhancement over all the testbeds developed by the RSSI-based positioning methods shows the benefits of using the least-square method in the presence of high error rates. However, the performance of the different range-free algorithms and the RSSI-based localization methods are surprinsigly different than the expected from the simulations and the previous study seen in section 4.3. The FCH algorithms and DV-hop algorithms, which have been presented as the best position estimations in the simulation environment, do not work properly on our several testbeds due to two main causes:

1. The number of asymmetries in the network
2. The miscalculation of correction factor and average hop values

Table 5.3: Average position errors and their respective standard deviations in horseshoe layout

Testbed Config.		Absolute Error [m]	*DV-Hop*	*DV-Distance*	*FCH*	*RSSI1*	*RSSI2*
4A	50N	*Mean*	2.8469	2.6363	2.7378	2.7092	2.6298
		Stdv	1.5553	1.0241	1.173	0.8001	0.9099
	100N	*Mean*	3.5244	2.1853	3.2709	2.8427	2.7231
		Stdv	1.5693	1.1146	1.2545	0.7657	0.8243
8A	50N	*Mean*	2.5388	2.3732	2.16	2.4373	2.2988
		Stdv	1.2318	1.1351	1.1618	0.7274	0.7636
	100N	*Mean*	3.1032	1.8759	2.1371	2.6317	2.5643
		Stdv	1.2910	0.8558	1.0161	0.7634	0.7930

To understand better the effect of asymmetries produced in the localization process using real hardware, we divide the asymmetries of every node in the groups "Connectivity In Range" and "Connectivity Out of Range" depending on whether the node with the asymmetrical link is inside or outside the 3m artificial communication range.

Figure 5.19 shows the asymmetrical links per node for two of the setups used in our experiments. Here, we can see that a higher proportion of the detected asymmetries corresponds to sensor nodes located outside the ficticious circular communication radio (depicted as red bars). As we see in section 1.2, the distortion of the circular communication radio model are produced by several signal propagation effects such as reflections, refractions, and fading. These radio transmission phenomena are stronger in real indoor scenarios, producing a negative impact on the position estimations achieved by the several localization algorithms.

Looking closer at the data obtained by the nodes, we have realized that this kind of asymmetries have directly affected the hop-by-hop counter and the propagation of information through the whole network. In particular, the "Connectivity Out of Range" is the responsible for the incorrect estimations of the factor corrections calculated in every landmark when the FCH method is used. In a similar manner, the average value of the hop-distance used by the DV-hop approach has been barely influenced by the same factor as well. These miscalculations degrade the performance of both algorithms resulting in the poor position accuracies showed in table 5.2 and 5.3.

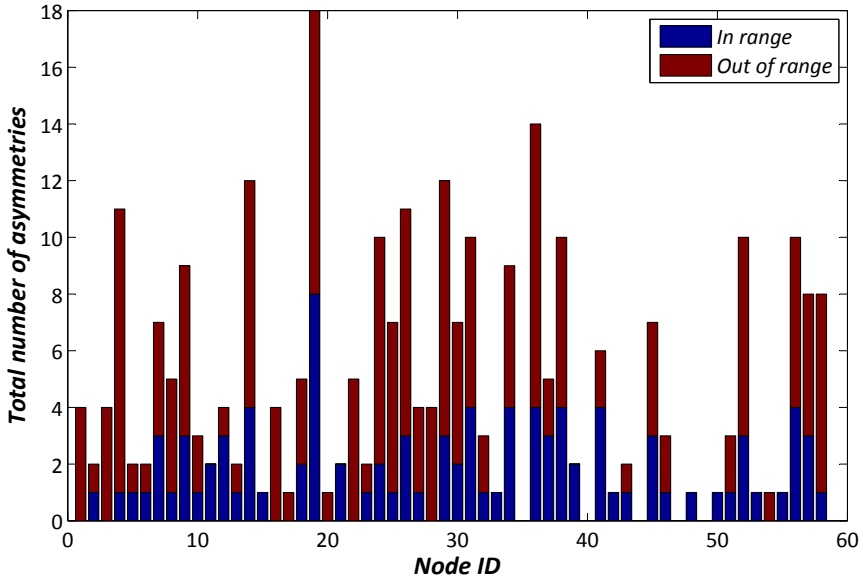
Comparing the accuracy levels shown in the uniform (5.2) and horseshoe (5.3) tables, we find that when using the same anchor fractions and the same number of nodes, the uniformly distributed layouts have achieved slightly more satisfactory location estimations than those obtained in the horseshoe topologies. Such inaccuracies in the horseshoe networks meets our expectation that the distances estimations by DIN, which is the foundation of our positioning algorithms, perform better in uniform distributions than anisotropic networks. Additionally, it can be observed that the results do not produces a final conclusion related to the number of reference nodes as well as the node density in the system to the better position accuracies via multilateration.

An interesting question that we want to examine is the influence of the node placement on the position accuracy achieved by the algorithms implemented on our scatterweb nodes. Figure 5.20 shows one of the tridimensional plots derived from our different layouts. Here, we depict the localization errors of several range-free and range-based algorithms versus node position in a uniform distribution using 4 beacons and 100 unknown nodes. The spatial distribution of the position errors achieved by all the schemes show that the most inaccurate values are located at the edge of the deployment area. This phenomena with the range-free and range-based algorithms is related to the fact that the less information that a node has about the perimeter of the network to locate itself (regions with low local densitiy), the more likely the borders of the network will represent a low accuracy distance estimation area inherited by the theoretical analysis of the DIN algorithm (the ideal radio disc unit is defective on the perimeter). In addition, the multipath, reflections, scattering and other physical phenomena are stronger on the borders due to the proximity of the nodes with metallic doors, furnitures and windows installed in the seminar room where the nodes are deployed.

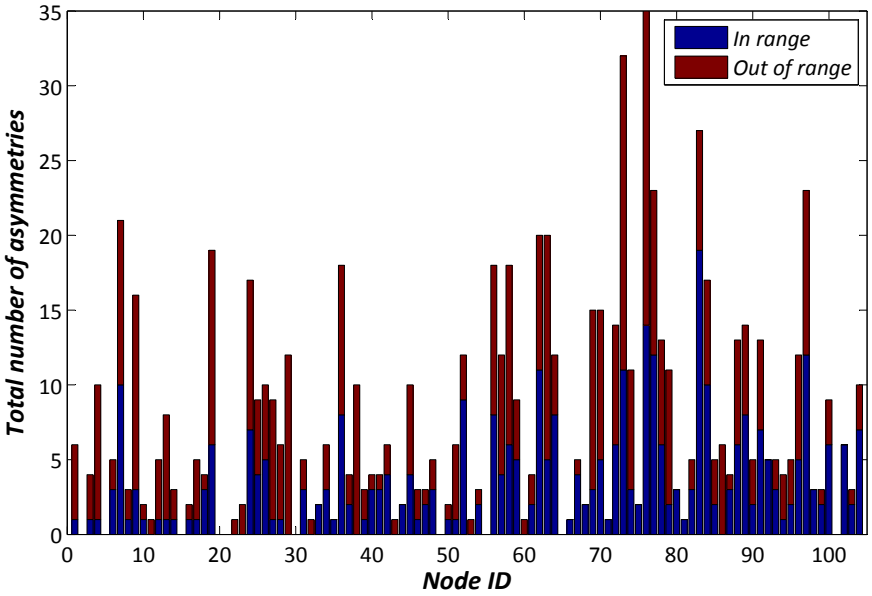
Analyzing the information obtained during run time using the RSSI-based algorithms, we observe both RSSI functions tend to place all the sensor nodes in the middle of the room causing exceptionally good values for the inner nodes. This effect can be seen in Figures 5.20 d and e. Different than the RSSI method, most of the range-free methods try to locate the nodes near its real position with acceptable accuracies not only in the center area of the network, but also for some nodes deployed close to the borders as shown in Figure 5.20 c. This effect, concentrating most of the location estimations on the center of the deployed area, is a common characteristic when the mean least-square method is applied in the presence of distance errors. Due to the result that all our methods seen previously (with exception of PIV) have used multilateration to find the location of the sensor nodes, the center location trend can bee seen in most of the plots in Figure 5.20, which is stronger on the RSSI-based localization algorithms. Overall, we can conclude that node placement does seem to have a strong influence on the position calculation in range-free methods, but more and larger scenarios have to be evaluated to add statistically significant evidence to such a proposition.

5.5.1 The PIV Algorithm as Refinement Phase

Although the FCH and DV-Hop approaches do not present the accuracy expected, the DIN algorithm has still worked producing competitive position estimations through the

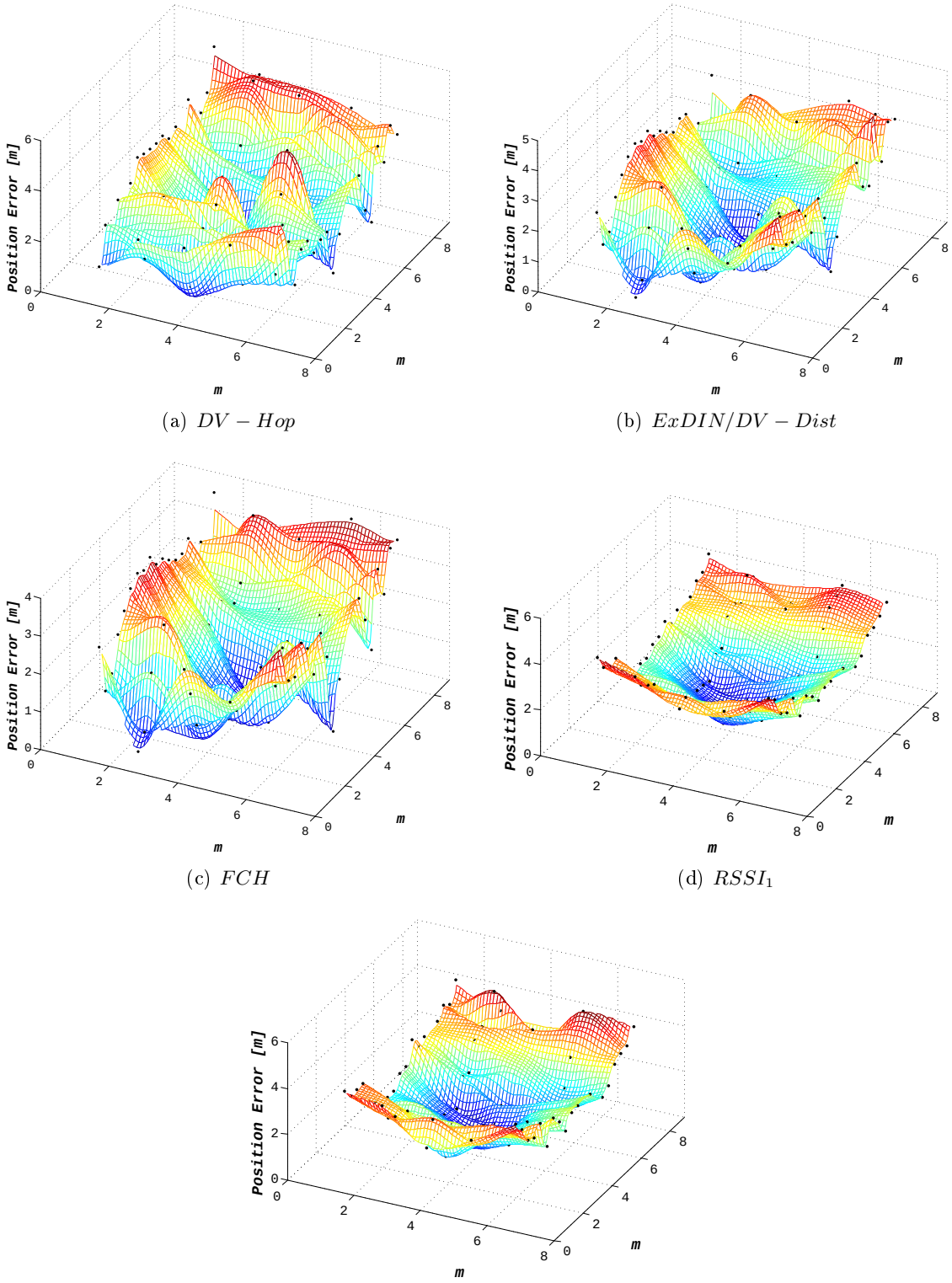


(a) Uniform 8A - 50N



(b) Horseshoe 4A - 100N

Figure 5.19: Link asymmetries per node in uniform and horseshoe configurations



(a) $DV - Hop$ (b) $ExDIN/DV - Dist$

(c) FCH (d) $RSSI_1$

(e) $RSSI_2$

Figure 5.20: Localization error versus position in uniform network configuration using 4 anchors and 100 unknown nodes

ExDIN/DV-dist algorithm. The exceptional location efficiency obtained using DV-Dist in horseshoe with 8 anchors and 100 nodes was produced by unusually direct shortest paths to the landmarks. Since the DIN algorithm presents good position estimations through the ExDIN algorithm, the implementation of the PIV algorithm as a refinement stage in our sensor configurations has been developed as the next step.

The main idea for this approach is to use as initial node position estimations the ones calculated by the ExDIN/DV-Distance, DV-Hop, and FCH approaches. The final node location has been obtained after 30 iterations. The number of iteration was considered sufficient to evaluate the convergence ratio of the PIV algorithm due to the good initial location estimations, and the previous simulation runs which showed that the first iterations contribute to the enhancement of position accuracies. Our findings are summerized in tables 5.4 and 5.5. In order to better describe the behaviour of the PIV algorithm, we introduce a new parameter labeled as Δ_{PIV} in this section. We define Δ_{PIV} as the difference between the initial average position error ($E_{initial}$) and the final average position error (E_{final})after the last iteration using the PIV algorithm. Finally, this difference is divided by the initial average position error and multiplied by 100 to create a porcentage (see Equation 5.16).

$$\Delta_{PIV} = \frac{E_{initial} - E_{final}}{E_{initial}} \cdot 100(\%) \tag{5.16}$$

Equation 5.16 shows the relative degree of the position accuracy variation between the inital and last location phase using the PIV algorithm. A positive magnitude of this metric represents an enhancement of the initial average displacement estimation, whereas a negative value is asociated with a loss of the average node position accuracies.

Table 5.4: Position errors after applying the PIV algorithm in uniform configuration

			DV-Hop		***ExDIN/DV-Distance***		***FCH***	
Testbed Config.		**Absolute Error [m]**	***Initial***	***PIV(30)***	***Initial***	***PIV(30)***	***Initial***	***PIV(30)***
$4A$	$50N$	*Mean*	2.4851	2.1764	2.4637	2.3965	2.4321	2.2346
		Stdv	1.3449	1.3172	1.2096	1.1695	1.3872	1.2981
		$\Delta_{PIV}(\%)$	12.42		2.73		8.12	
	$100N$	*Mean*	3.2712	2.5417	2.2185	1.8087	2.6512	2.1673
		Stdv	1.8767	1.2630	1.0298	0.8408	1.5511	1.0631
		$\Delta_{PIV}(\%)$	22.3		18.47		18.25	
$8A$	$50N$	*Mean*	2.4962	1.6466	2.3529	1.6153	2.2356	1.4968
		Stdv	1.2829	0.9429	1.1002	1.0221	1.1126	0.8858
		$\Delta_{PIV}(\%)$	34.04		31.35		33.05	
	$100N$	*Mean*	2.3737	1.5808	2.0014	1.3724	1.9738	1.4017
		Stdv	1.2417	0.9543	1.2192	0.8704	0.9684	0.8610
		$\Delta_{PIV}(\%)$	33.41		31.43		28.98	

Examining the results from table 5.4, an improvement is noticeable for all the experiments

for all the node configurations for the location precision produced after 30 iterations using the PIV algorithms. Different degrees of position enhancements were successfully achieved by the iterative method using the average positions calculated by the DV-Hop method. This method resulted in the higher location improvement in all the experimental cases. The most successful case of location improvement using PIV was obtained using 50 nodes and 8 reference nodes in the uniform distribution utilizing an initial position attained by the DV-Hop approach. For this case, a Δ_{PIV} value of 34.04% was yielded.

However, the highest enhancement value does not necessarily mean the best accuracy was achieved in all our experiments. The best accuracy calculated by PIV as a refinement algorithm was in the cases where the position estimations where found through the ExDIN/DV-dist algorithm using 100 deployed nodes in a uniform distribution and 8 beacons with an average precision of 1.37 ± 0.87 meters.

We can observe on the initial position reported in the uniform distribution and horseshoe distribution tables that the most precise locations are attained through the ExDIN algorithm confirming the first results presented in this section. Once again, the initial position accuracies are affected by the node configuration as it can be seen in table 5.5. Here, the best average position accuracy has a value of 1.59 ± 0.81 meters using 50 unknown nodes and 8 reference nodes. Although the initial position estimations for the horseshoe layout have higher innacuracies, the PIV algorithm does not present higher correction compared to the Δ_{PIV} values registered in table 5.4. We consider that this behaviour is due to the close relationship between the PIV and the DIN algorithm and the external factors such as fading, interference and the number of link asymmetries. The negative value of Δ_{PIV} in table 5.5 for the testbed where a horseshoe configuration include 4 reference nodes and 100 unknown nodes is a direct consequence of the anomalies from wireless medium resulting in an accuracy degradation.

Table 5.5: Position errors after applying the PIV algorithm in horseshoe layout

			DV-Hop		***ExDIN/DV-Distance***		***FCH***	
Testbed Config.		**Absolute Error [m]**	***Initial***	***PIV(30)***	***Initial***	***PIV(30)***	***Initial***	***PIV(30)***
$4A$	$50N$	*Mean*	4.4617	3.7843	3.1395	2.6018	3.3185	3.1001
		Stdv	1.8726	1.4976	1.5006	1.2438	1.4859	1.2347
		$\Delta_{PIV}(\%)$	15.18		17.13		6.58	
	$100N$	*Mean*	3.7961	3.6606	3.3498	3.5325	3.8940	3.8728
		Stdv	2.3595	1.9991	2.0836	2.0344	2.1884	1.9587
		$\Delta_{PIV}(\%)$	3.57		-5.45		0.55	
$8A$	$50N$	*Mean*	2.2257	1.7417	2.3076	1.5924	2.1862	1.7526
		Stdv	0.9827	0.8817	0.7795	0.8156	1.0816	0.8935
		$\Delta_{PIV}(\%)$	21.74		30.99		19.83	
	$100N$	*Mean*	2.6926	2.0758	2.1115	1.6702	2.1949	1.6479
		Stdv	1.0222	0.9377	0.8078	0.7467	0.7733	0.7551
		$\Delta_{PIV}(\%)$	22.91		20.90		24.92	

To assure the best evaluation of the efficiency with PIV as refinement algorithm, the same local DIN distances of each node in the system used to produce the initial positions by using the range-free methods have been employed to improve the location accuracies through the iterative algorithm. That is to say, the same set of neighborhoods (created in the multilateration phase) is used as an input parameter in the refinement phase. With this constraint on the PIV algorithm, the location estimations must theoretically converge to the same position estimation for all nodes independently of the localization methods utilized before the iterative process.

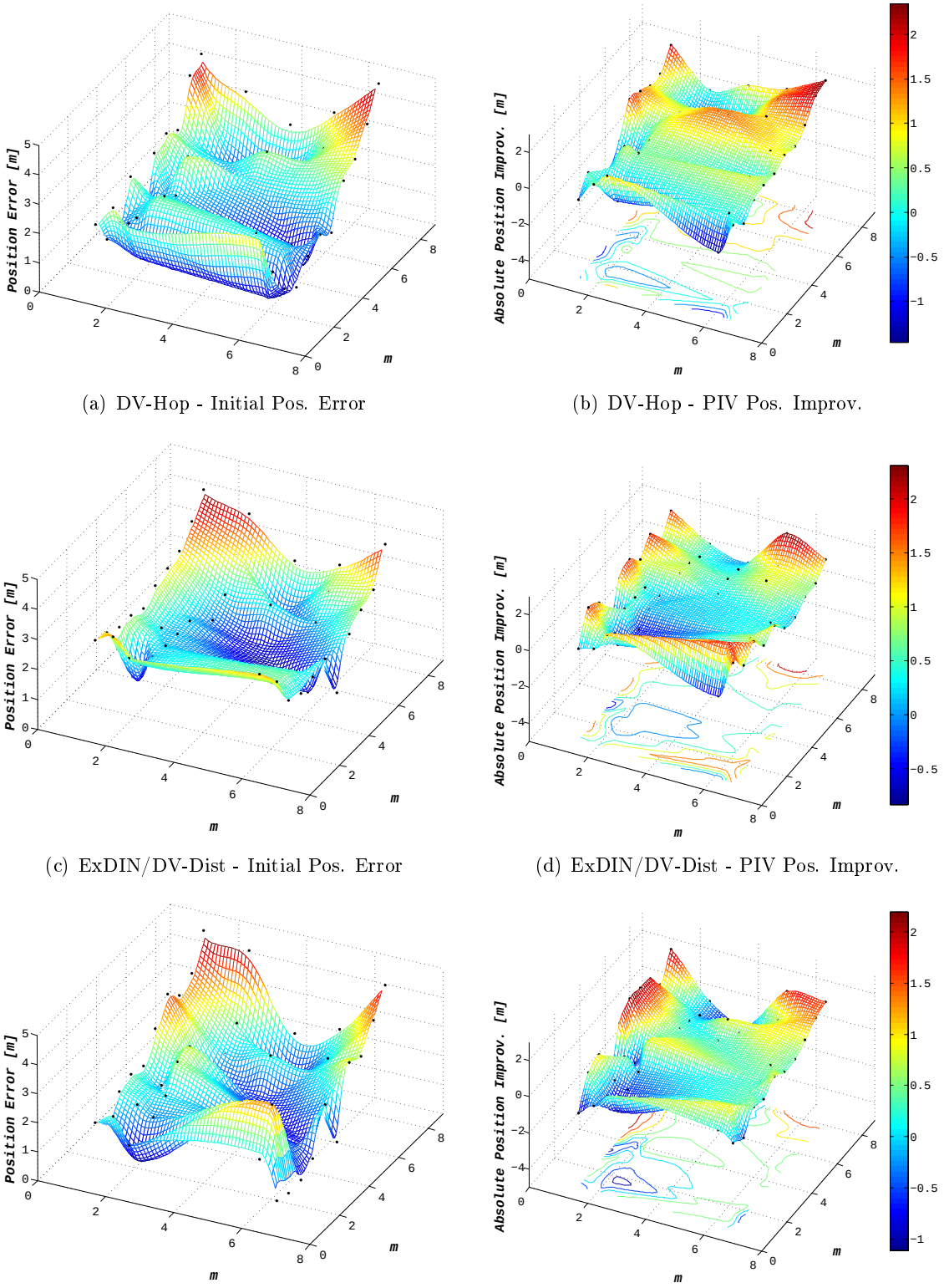
Looking closer on the results presented on tables 5.4 and 5.5, we observe a considerable variation of the final average position estimations after the PIV iterations. This observation constitutes an indicator of the influence of the initial position on the PIV performance. More experimental tests with several network densities must be done to discover the impact of the network topology as well as the node location at the beginning of the iterative localization phase.

The accuracy attained by using the PIV algorithm as an refinement phase has produced a better performance than the results obtained on the multilateration phase. This can be confirmed when the final average positions errors for uniform and horseshoe distribution using the iterative algorithm are compared with the average location estimations of the multilateration phase shown on the tables 5.2 and 5.3. Looking closer at the average values for uniform distributions, the estimations using the PIV algorithm using 8 anchors (see Table 5.4) improve the average location errors with an accuracy enhecement from 31% (DV-Hop algorithm with 8 anchors and 50 nodes of table 5.2)to 49% (DV-Hop algorithm 8 anchors and 100 unknown nodes). The final estimations for all the accuracy improvements by PIV are with 8 anchors are at least 27% more precise than the estimations realized by multilateration using range-based methods.

The accuracy of PIV using 4 anchors in uniform distributions reached the maximal percentage of 17% when the final result of PIV using 50 unknown nodes is compared with the average error obtained using the DV-Hop approach with 4 anchors and 50 nodes (see table 5.3). Comparing the accuracies improved by PIV with 4 anchors (Table 5.4) and the results obtained with the RSSI-based algorithms with 4 anchors shown in table 5.3, we can observe that slight precision enhancements are achieved. Only in the case of the ExDIN with 100 nodes did the results show a significant elevation of accuracy being 21% more accurate than the best value gathered with any of the RSSI-based localization using 4 anchors and 50 nodes.

When the results of PIV on a horseshoe configuration is compared with the average errors gathered through multilateration in the same node distribution, we observe that deploying 8 landamrks, the percentage of accuracy (Table 5.5) over the ones collected in Table 5.3. PIV produced an enhancement of precision up to 36% (PIV-ExDIN 8 anchors 100 nodes versus Multilateration RSS1 8 landmarks 100 nodes) over all the node position errors produced in a horseshoe layout with 8 beacons.

Using 4 landmarks and 50 deployed nodes in horseshoe layout, PIV yields a maximum accuracy improvement of 45% when we use an initial position estimation calculated by the FCH algorithm. However, for the case of horseshoe configurations with 4 reference nodes, the precision has not always improved over those obtained by multilateration with the same



(a) DV-Hop - Initial Pos. Error

(b) DV-Hop - PIV Pos. Improv.

(c) ExDIN/DV-Dist - Initial Pos. Error

(d) ExDIN/DV-Dist - PIV Pos. Improv.

(e) FCH - Initial Pos. Error

(f) FCH - PIV Pos. Improv.

Figure 5.21: Spatial distribution comparison of initial position error vs PIV location improvement using DV-Hop, DV-Dist, and FCH algorithms in a horseshoe configuration with 8 anchors and 50 unknown nodes

network conditions. One must keep in mind that the initial experiment values using PIV with 4 anchors and 50 and 100 deployed nodes (see Table 5.5) have been less accurate than the misplecement errors gathered for multilateration in the same conditions (as can be seen in Table 5.3). Thess bad initial values have restricted the enhancement of accuracy of the PIV algorithm, producing a limited and unsatisfactory improvement over the node position estimations.

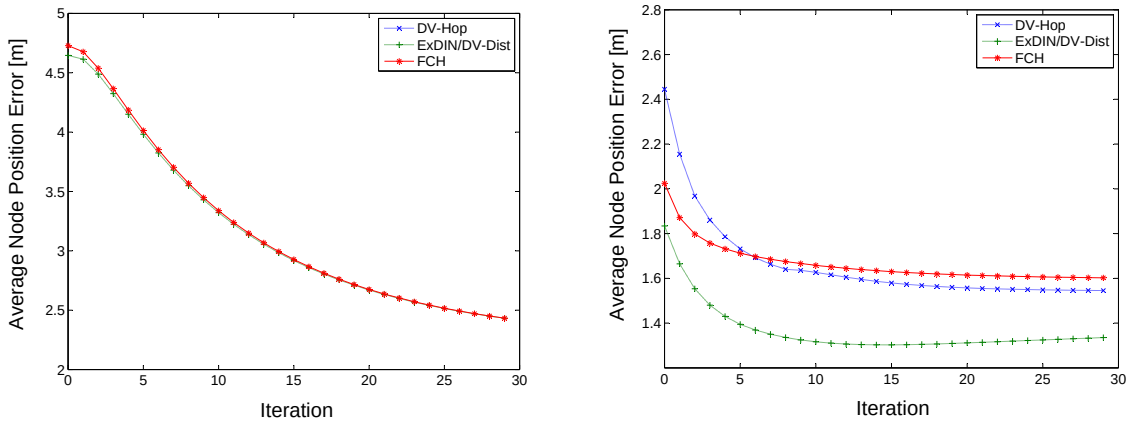
An interesting question we wanted to investigate was the spatial distribution of error correction after applying the PIV algorithm. The tridimensional representation of the position errors for the inital phase of the refinement process (plots on the left columns) and the absolute value of the PIV improvement on its final iteration (plots on the right side) of 50 deployed nodes with 8 landmarks in a horseshoe configuration are depicted in Figure 5.21. The graphs shows that the PIV algorithm improve the node location estimations where the range-free methods lacked precision. The red colored regions of the plot on the right side depict the areas the maximal position corrections achieved by the PIV iterative method. As we can see in Figure 5.21, the zones where the least satisfactory position estimations are obtained after multilateration (see plots of the left column) coincide in most cases with those where the PIV algorithm has produced the most remarkable accuracy enhancements. The areas where the refinement phase produces some accuracy degradations are observed on the blue colored regions which usually take a maximum negative value up to 1 meter. Another example which demonstrates the good results incrementing the location estimation through the PIV algorithm is shown in Figure F.5. With these two examples, we can confirm the ability of PIV to correct the position errors where the inaccuracy levels of the network are higher, as seen in tables 5.4 and 5.5.

5.5.2 The PIV Algorithm Behavior on Low Node Density Network

The density networks presented in the last section are far away from constituting a typical real network scenario. Taking as reference that most of the real applications work with sparse node networks, we implemented a new testbed using the scatterweb nodes with a low node density distribution, as can be seen in Figure G.5. In order to find the influence of the number of reference nodes in such networks, an extra layout using 4 landmarks placed at every corner of the 8x9 m deployed area was implemented with the same nine unknown nodes.

Figure 5.22 shows the average position error in every iteration of the PIV refinement phase using 4 and 8 landmarks. An increment of accuracy on the average location estimations for both testbeds is achieved by using the PIV algorithm as a refinement phase. Looking closer at the curves using 4 anchors of Figure 5.22 a, we note that the nodes have not estimated their initial positions by the multilateration using the DV-Hop method. This effect was produced because the nodes were not able to forward some of the DV-Hop packages to the different anchors of the system to later compute the average hop-distance. However, the initial positions using the FCH and DV-Distance have been completed for both experiments.

In spite of the low connectivity levels of the layouts, a notable reduction of the position errors has been accomplished by the iterative method. The poor initial average accuracies attained by multilateration with DV-Dist and FCH using 4 anchors have been improved from



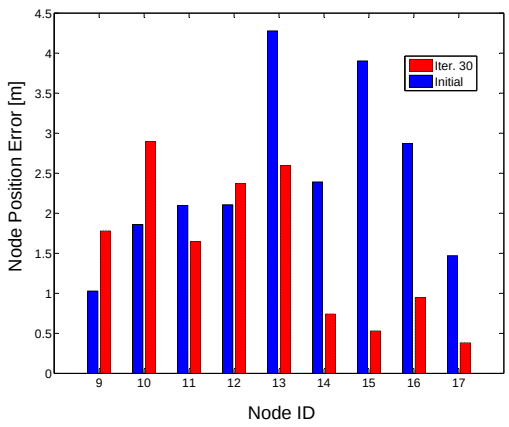
(a) 4 Anchors (b) 8 Anchors

Figure 5.22: Average position error versus iteration in low node density configurations

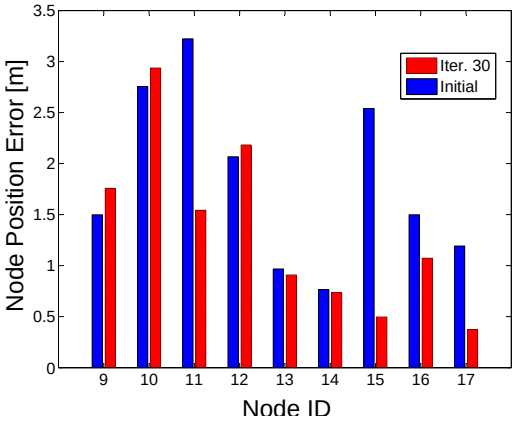
an average misplacement of 4.64 ± 1.52 meters using ExDIN/DV-Dist and 4.72 ± 1.51 using FCH to a final stage after 30 iterations of 2.43 ± 0.42 and 2.43 ± 0.42 meters respectively. These values consitute an accuracy enhancement of more than 50% on the final average error.

In the experiment using 8 anchors, the results collected show good accuracy improvement on the average position errors incrementing the precision from 2.44 ± 1.07 meters using multilateration with DV-Hop, 2.02 ± 0.58 meters achieved by multilateration with FCH and 1.83 ± 0.85 meters attained by multilateration with DV-distance, to a final estimation through PIV with values of 1.54 ± 0.93 meters, 1.6 ± 0.99 meters, and 1.33 ± 0.84 meters respectively. That means, PIV has generated an improvement of a maximum of 36% and a minimum of 27%.

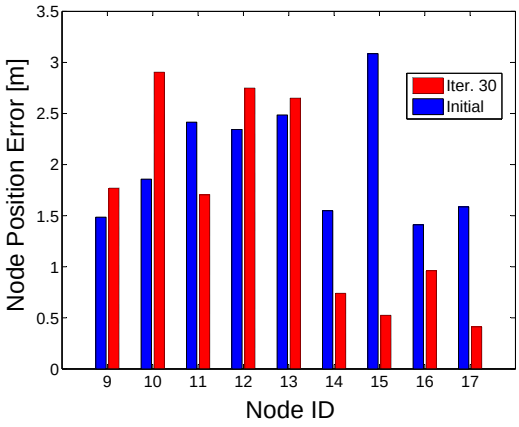
To better understand the enhancement of the node location estimations by using PIV, the position errors for each deployed node and multilateration method in the testbed using 8 anchors are depicted in Figure 5.23. Here, the node error positions after the multilateration phase are depicted by blue bars, while the final location errors after 30 iterations using the PIV algorithm are represented in red. An inspection of the error bars of every node reveals that nodes with higher IDs have placed better after the iterative method. We found that a better neighbor table quality allows a better correction of the misplaced nodes in the network. Nodes, such as the ones with 14, 15, 16 and 17 ID numbers, have precisely calculated their neighbor DIN distances so they locate better using the PIV algorithm. Altough the iterative method is not able to correct the position of all the nodes (such as the nodes with the Id numbers 10 and 12), a vast majority of nodes have presented correction up to more than 2 meters after the refinement process (see error bars of node 15 in Figure 5.23). Thus, the usage of PIV in low density networks in worthwhile from a network-wide perspective to improve the location of the majority of the sensor nodes.



(a) DV-Hop



(b) DV-Dist



(c) FCH

Figure 5.23: Average position error per node after applying the PIV algorithm for several location algorithms in low node density network

5.6 Conclusion

In this chapter, different techniques to locate the nodes of WSN based on the DIN algorithm have been explained. First, different methods to obtain the distances to the reference nodes such as the ExDIN/DV-Distance, FC, and FCH algorithms have been proposed. The efficiency of those node-to-landmark distance methods have been investigated through extensive simulations using the `ns-2` simulator with different network densities. The simulation comparison using multilateration with FCH, ExDIN/DV-Distance and DV-Hop algorithms has shown that our algorithms produce similar or slightly better position estimations in uniform and near-uniform configurations for a number of deployed nodes. The algorithm that produced the best performance for the majority of layouts and local densities was the combination of multilateration with the FCH algorithm.

We suggest a new and innovative algorithm called PIV to find the node locations. This algorithm can be used either as an autonomous localization algorithm or as advanced refinement phase after applying a location sensing approach. The efficiency produced by the PIV algorithm has been compared with different multilateration methods such as the Adhoc Position System, the FCH algorithm and the ExDIN/DV-Distance algorithm using a C++ enviroment with different configurations and network densities. Although the PIV algorithm achieved satisfactory location accuracies in our simulations, in some simulations, the good position estimations using FCH overcame PIV. These results limited the original idea of using the PIV algorithm as a refinement method, but the implementation of different localization methods using real hardware has pointed out the usefulness of PIV in this phase.

A quantitative experimental performance analysis between different distributed positioning systems has been addressed in this chapter. The algorithms implemented on the scatterweb nodes were the DV-hop algorithm, the FCH algorithm, the ExDIN/DV-distance and two rssi based localization approaches. All the algorithms have been tested in multihop wireless networks with different number of deployed nodes and several anchors ratios in the system. To the extent of our knowledge, the practical comparisons between different range-free and range-based algorithms have been accomplished for the first time with a multihop indoor environment using real sensor nodes.

Our findings show that the PIV algorithm produce better position accuracies in practical enviroments than the RSSI-based algorithms and the other distributed algorithms included in the study. Althought our first simulations have been declined the possible use of PIV as refinement phase, the differences between the simulation and practical enviroment have enabled the use of PIV as an iterative method, producing average location errors up to 1.37 ± 0.87 meters for uniform distributed networks and 1.59 ± 0.81 meters for near-uniform distributions. Furthermore, the refinement phase in low network densities have improved the node misplacement up to 53% of the average values achieved by using multilateration. The best average position estimation in low node densities environment has a value of 1.33 ± 0.84 meters using 8 reference nodes on the perimeter of the node deployment area.

The good accuracies of position estimations using the PIV algorithm as a refinement phase with different testbed layouts reflect the need to influence more additional experimental evaluation in order to better understand the network conditions of real-world scenarios and the propagation effects throughout the wireless medium in indoor enviroments.

CHAPTER 6

Conclusions

This work addressed the problem of recovering the topology of an ad-hoc wireless sensor network for indoor environments without dedicated hardware and using distributed system. The foundation of all our location-sensing algorithms was the DIN algorithm which related the area of intersection of two overlapping transmission ranges to the number of local density involved to determine their distances. Using this algorithm as the first step for different localization methods to determine the node locations, we have proposed the FCH and the PIV algorithms. Being the PIV algorithm which achieves the best accuracies in practical experiments.

The first contribution of this thesis is the discovery that the neighboring distances of a given sensor node can be estimated by the analysis of the local node density. The duty zone of the DIN algorithm is between SRR values of 0.0769 and 0.5 for uniformly distribution networks and between SRR values of 0.0625 and 0.5 for near-uniform distribution networks with the majority of normalized error values below 0.39R. Here, the best average, normalized distance error has been 0.127R for a uniform distribution of sensor nodes. The good results obtained by testing the DIN algorithm with different testbed layouts reflect the findings of the simulations, although we were only able to analyze a fraction of the simulation cases.

Figure 4.12 demonstrates the advantage of using DIN to determine neighboring distances versus RSSI ranging methods in uniform and horseshoe configurations. The ranging values yielded by using DIN are more than double the accuracies computed by the RSSI-based approaches.

The advantage of the DIN approach is that it does not need any kind of additional specialized hardware to determine distances, making it suitable for advances ad-hoc localization phases. Furthermore, DIN keeps the usage of memory, the communication overhead, and the computational process load at minimum. We, therefore, have used the DIN algorithm as the ranging method to discover the location of the sensor nodes.

Motivated by the good accuracies accomplished by DIN, four different localization schemes based on the DIN algorithm have been suggested in chapter 5. These algorithms are the ExDIN/DV-Distance algorithm , the FC algorithm, the FCH algorithm, and the PIV iterative method. The first three techniques have been oriented to estimate node to reference node distances, which later helps compute the node position by multilateration. A series

of simulations using the `ns-2` enviroment have shown that the best position accuracies are achieved using the FCH algorithm with multilateration. The FCH algorithm was confronted with the efficiency of the DV-Hop algorithm in uniform and near-uniform networks with several network densities as can be seen in Figures 5.6, 5.9, and 5.10.

We completly simulated the PIV algorithm as an autonomous localization method using DIN and real neighbor distances confronting the multilateration method using real neighbor distances, as well as the DV-Hop, ExDIN/DV-distance, and FCH algorithms in an ideal environment simulator based on the C++ language. This characterization comprises the analysis of the different algorithms, varying the anchor fraction in the system, the number of deployed nodes, and the transceiver communication ranges in uniform and near-uniform distributions.

After the performance analysis of all the algorithms, we realized that in spite of using the real neighbor distances in multilateration for a multihop environment, it does not mean that the method will find the right node positions. This is due to the snake-like twisted paths through other node connectivities in the network, that a given node has to navigate through to reach a landmark. This navigation affects all the algorithms that use the multilateration method. As in our result simulation, the FCH algorithm presented better accuracies, correcting the snake path problem in the majority of node densities and in all the simulations that vary the number of deployed nodes and/or the radio transmission ranges.

Although the PIV algorithm does not share the snake path problem, the simulations of the PIV algorithm are not more accurate than the multilateration approaches for all the number of deployed nodes or all the values of radio communication ranges. The duty zone where PIV a lighter increment of precision than the multilateration methods is confined to normalized radio transmission ranges between 0.2 and 0.4 with medium and high node densities (in our simulations between 45 to 75 deployed nodes) for the most of the network settings with normalized error values less than 0.15.

An interesting point always present in the simulations was that the PIV algorithm could estimate the node positions of very low node density networks. Unlike the multilateration approaches, the PIV algorithm is able to estimate node locations using a single connection with one of the deployed landmarks.

A weakness of PIV used as a localization system is the velocity of convergence. The number of necessary iterations needed to locate a node can be a potential problem for some radio transmission ranges and high density networks as can be seen in Figure 5.12 c. However, if the number of landmarks is increased in the system, then the accuracy of PIV became competitive compared to the FCH algorithm, as can be seen in Appendix E. After different simulations of PIV with uniform and grid distributions varying the number of deployed nodes, radio communication ranges and number of reference nodes, we concluded that the first 50 iterations of the PIV algorithm represent an 80% change ratio of the iterative process. That is to say, the first iterations using PIV have a bigger impact on the increment of the locaction accuracy than the higher iteration values. The number of minimum iterations to reach a good position accuracy is directly proportional to the number of deployed nodes in the network, but inversely proportional to the number of anchors in the system.

The good results observed in the simulations of the several localization methods, it is confirmed by the different practical testbed using the scatterweb sensor nodes. The experi-

Table 6.1: Comparison table for different algorithms and locate-sensing prototypes in WSN

Parameters	Centroid	Bounding Box	Ad-hoc Localization System	Ad-hoc Positioning System	Easy Living System	N-Hope Multilateration	Approximate Point-in-Triangulation	Pushpin System	NIDES	Cricket System	ExDIN	FCH	PIV
Scalability	--	+	+	++	-	+	-	--	++	-	+	++	+
Computational Effort	##	##	#	#	**	O	##	O	#	#	##	##	##
Self-Organizing	O	O	+	++	--	+	O	-	+	+	+	+	+
Distance Estimation	NA	O	+	+	++	+	NA	+	+	+	O	O	+
Anchor fraction	**	*	O	O	NA	O	*	*	NA	O	O	O	#
Energy Efficient	++	+	O	+	--	-	+	O	O	-	+	+	O
Real Hardware Implementation	+	NA	+	NA	++	+	NA	++	NA	++	++	+	++
Cost Implementation	##	NA	O	NA	**	O	NA	*	NA	*	##	##	##
Position Accuracy	-	O	+	+	++	+	+	+	NA	+	O	+	++
Refinement	NA	NA	+	NA	NA	+	NA	NA	NA	+	NA	NA	++

mental analysis presented in this thesis constitutes to the extent of our knowledge the first quantitative comparison which compares several distributed localization algorithms indoors. Nine different network configurations with several anchor fractions and node densties have been employed to evaluate three of our algorithms (the ExDIN/DV-Dist, FCH and PIV algorithm), comparing with the performance of the DV-Hop algorithm and two RSSI-based localization methods.

Examining the results achieved by the algorithms which use the multilateration method, we were aware of the dissimilarities between the ideal simulation enviroment and the real-world scenario. Physical phenomena such as asymmetry communication links between nodes, interference, multipaths, reflections, and strong fluctuetion in the radio signal have been common precision attenuators for all the algorithms tested. In spite of the real wireless characteristics, the best position estimation using a uniform distribution has been obtained with the ExDIN/DV-Dist algorithm resulting in an average position error of 2.02 meters with a deviation standard of ± 1.00 meters (see Table 5.2). An increment of the average position accuracy has been reached for the horseshoe configuration using the ExDIN/DV-Dist algorithm with an average position accuracy of 1.87 meters and a deviation standard of ± 0.85 meters (see Table 5.3).

Although the best average values have been calculated by our algorithms, they represent a maximum enhancement of 11% for uniform distribution and 18% for the horseshoe configuration compared to the best performance values attained by using the RSSI-based localization algorithms of every distribution. Unlike the multilateration algorithms, the PIV algorithm used as a refinement phase has confirmed the better accuracy of position estimations of our approach compared to the multilateration algorithms and the RSSI-based localization systems.

The best accuracy of using the PIV algorithm in uniform distributions was 1.37 ± 0.87 meters (see Table 5.4) which depicts an enhancement of position accuracy of 39% comparing with the best result yielded by using the RSSI technique in this configuration. For the horseshoe layout, the best precision of the PIV algorithm has a value of 1.59 ± 0.81 meters (see Table 5.5) reflecting an increase in the location precision of 30% in comparison to the best

value achieved by the RSSI approaches.

Finally, a test with an asymmetric configuration and low node density has been conducted to examine the behavior of PIV on such network settings. The best accuracy of PIV on this network conditions has a value of 1.33 ± 0.93 meters, improving the initial position of the nodes up to 37% of the node misplacements as can be seen in Figure 5.22. Our findings show that the PIV algorithm produced better node position estimations when the algorithm is put into practice with real sensor nodes rather than other distributed algorithms and the classical RSSI techniques. Nevertheless, we need more experiments to better understand the real-world conditions. In chapter 3, we have compared different range-free algorithms as well as position-sensing prototypes in WSN. Table 6.1 is an extended version of Table 3.3 which includes the three localization algorithms proposed in this thesis. The criteria to evaluate all these algorithms were the same symbols as those used on Table 3.3.

CHAPTER 7

Future Work

The performance of the DIN algorithm was evaluated analytically, by ideal programming environment with help of the `ns-2` simulator and the C++ based simulator, and finally using real hardware in uniform and near-uniform distributions, varying the number of deployed nodes, the anchor ratio, and the radio communication range in chapter 4. However, it is necessary to examine the behaviour of DIN in no-uniform distributed networks. To achieve this goal, simulations with different network densities have to be programmed.

Since the DIN algorithm is the foundation for the PIV algorithm, ExDIN/DV-distance algorithm, and the FCH algorithm proposed in this work, it will be especially interesting to find out whether a lower bound for the number of neighboring nodes and a given accuracy for the three algorithms can be derived for multi-hop, medium, and high density networks.

Another important point to investigate is the impact of changing the ideal transmission range to determine more accurate neighboring distances by using the DIN algorithm and the influence on the performance in our three proposed algorithms.

Taking the practical perspective of the DIN algorithm, it is necessary to minimize the degradation of accuracies due to the hostile features of the wireless medium such as the asymmetry communication links, the fluctuations of the radio signal, reflections, and other attenuators causing external factors.

An investigation into the influence of the fictitious variable transmission range in the DIN algorithm using different transmission power into different real scenarios is a desirable future work.

Another aspect not addressed yet in this thesis is the dropped packages due to external factors on the FCH algorithm and PIV algorithm. The importance of the delivery of information between anchors throughout the network is of decisive importance with respect to the estimation of the factor correction. On the other hand, the neighboring information of a given node is essential to new position computation on the iterative method (see section 5.3.1). The study has to be considered from the network layer perspective to discover the minimum and maximal ratio of the package lost as well as its relationship with the accuracy of the whole system.

To complete this goal a more realistic physical model has to be developed in C++ language

that should take into account phenomena such as reflections, selective fading, spatial and temporal radio signal fluctuations, multipaths, packet collisions, and corruption of transmitted packets in order to provide a more accurate environment simulations.

An open question to be explored is the repercussion of changing the number of reference nodes over the whole system on the accuracy of the FCH and the PIV algorithms. In one hand, the velocity of convergence of the PIV algorithm is notable better when the number of anchors is incremented over the perimeter of the deployed area (see section 5.3.5), but the influence of the landmarks deployed either on previous selected areas or deployed randomly on the position estimations is including as next future work.

A better understanding of the use of a PIV algorithm as a refinement phase has to be conducted by simulations paying special attention to the conditions that achieve the good convergence to the real node position. Factors, such as the initial position estimations, the connectivity of the neighboring nodes, the neighboring distance estimations, number of neighbor nodes, the number of inner unknown nodes or/and reference node on the neighbor tables, and the scope of the communication link directly affect the performance of the PIV algorithm. For this reason, simulation and practical tests using real hardware focused on those parameter is a worthwhile analysis for the future.

A practical usage of PIV as a refinement phase is based on section 5.5.1's very acceptable results. Nevertheless, the analysis of the PIV algortihm as an autonomous localization method is still an open topic in our work. Therefore, we plan to implement the position-aware system with real hardware the help of PIV as a localization approach.

Finally, it is necessary to develop a graphical interface to display on a central computer the location of the sensor nodes. The integration of this interface is especially advisable due to the facility to collect the main parameters of the distributed network. The collection of data on real time under real wireless scenarios will provided an easy and quick analysis of the different algorithms implemented on the sensor nodes.

Appendices

APPENDIX A

We will present the mathematical solution for laterion on two dimensions. The extension for the solution on three dimensions required solely the addition of a fourth distance from the node to the Landmark.

A.1 Solution with three beacons

We consider three reference nodes or beacons with positions determined by the coordinates (x_i, y_i) for $i = 1, 2, 3$, a node with unknown location (x_u, y_u), and its distance values from the reference nodes $r_i, i = 1, 2, 3$

The relationship of these elements can be denoted by a set of equations as follows

$$(x_1 - x_u)^2 + (y_1 - y_u)^2 = r_1^2 \tag{A.1}$$

$$(x_2 - x_u)^2 + (y_2 - y_u)^2 = r_2^2 \tag{A.2}$$

$$(x_3 - x_u)^2 + (y_3 - y_u)^2 = r_3^2 \tag{A.3}$$

Substracting A.3 from A.1 and A.2, we achieve the next two equations rearranging all the terms:

$$(x_1 - x_u)^2 - (x_3 - x_u)^2 + (y_1 - y_u)^2 - (y_3 - y_u)^2 = r_1^2 - r_3^2 \tag{A.4}$$

$$(x_2 - x_u)^2 - (x_3 - x_u)^2 + (y_2 - y_u)^2 - (y_3 - y_u)^2 = r_2^2 - r_3^2 \tag{A.5}$$

Since we are interested in writing the equations as a set of linear equations, we expand all the binomial expressions of the Equations A.4 and A.5 and rearrange terms resulting in

$$2\,(x_3 - x_1)\,x_u + 2\,(y_3 - y_1)\,y_u = \left(r_1^2 - r_3^2\right) - \left(x_1^2 - x_3^2\right) - \left(y_1^2 - y_3^2\right) \tag{A.6}$$

$$2\left(x_3 - x_2\right)x_u + 2\left(y_3 - y_2\right)y_u = \left(r_2^2 - r_3^2\right) - \left(x_2^2 - x_3^2\right) - \left(y_2^2 - y_3^2\right) \tag{A.7}$$

These equations can be rewritten into a linear matrix form as

$$2\begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} \left(r_1^2 - r_3^2\right) - \left(x_1^2 - x_3^2\right) - \left(y_1^2 - y_3^2\right) \\ \left(r_2^2 - r_3^2\right) - \left(x_2^2 - r_3^2\right) - \left(y_2^2 - y_3^2\right) \end{bmatrix} \tag{A.8}$$

For a system of linear equations in the form $Ax = b$, we get the solution of x when the mean square error $||Ax - b||^2$ is at a minimum. Applying the property of 2-norm of a vector (the square root of the sum squares of the vector elements), we obtain:

$$||Ax - b||_2^2 = (Ax - b)^T(Ax - b) = x^T A^T Ax - 2x^T A^T b + b^T b \tag{A.9}$$

Minimizing this expression is equivalent to minimizing the mean square error. Regarding this as a function in x, its gradient has to be set equal to zero:

$$2A^T Ax - 2A^T b = 0 \leftrightarrow A^T Ax = A^T b \tag{A.10}$$

APPENDIX B

We will present the mathematical basis for the algorithm called Weighted Density of Node Intersection (WDNI). The basic idea of WDNI is to approximate distances between two nodes using only the knowledge of local node densities, which is the number of nodes within the communication range of another. To derive a distance from this knowledge and implement an algorithm accordingly, we proceed in the following three steps:
First, we find a mathematical expression for the distance between two nodes in terms of the intersection area of their communication ranges. This distance we relate to the number of the neighboring nodes in the intersection area, constructing an approximation by means of evaluating different sizes of uniformly distributed networks. Finally, we weigh this distance with the number of nodes in the union of their communication ranges.

B.1 Relating distance to radio intersection area

We based WDNI on an idealized radio model in order to define an approximation model. Although we are aware of this assumption being false in reality as discussed in section 1.2, we use it in order to simplify the mathematical foundation.
Three main assumptions were taken into consideration:

1. Unit disc graph radio transmission range.
2. Identical transmission ranges for all the nodes in the network.
3. Uniform distribution of nodes in the network.

Considering the two nodes as shown in Figure B.1, we can obtain the intersection area A_i of the overlapping transmission ranges depicted as circles by geometrical analysis. The equations of the two circles with r, R being the radii describing the range of the nodes, are:

$$X^2 + Y^2 = R^2 \tag{B.1}$$

$$(X - d)^2 + Y^2 = r^2 \tag{B.2}$$

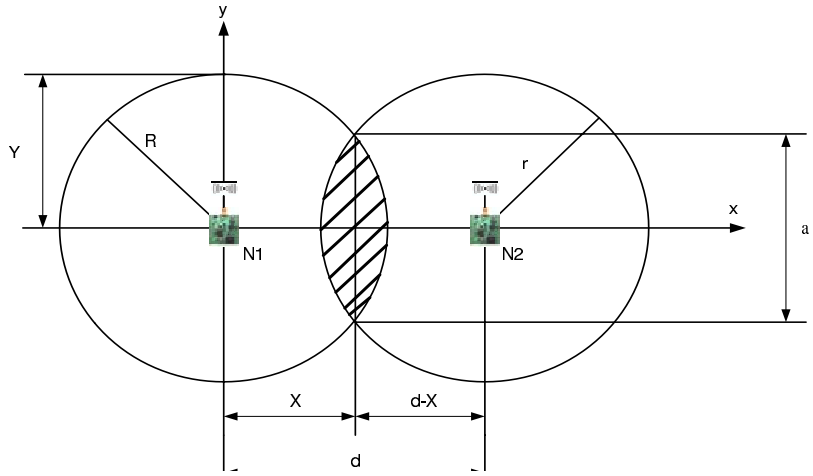


Figure B.1: Geometric analysis of the intersection area of overlapping transmission ranges

Combining B.1 and B.2 and solving for X results in

$$X = \frac{d^2 - r^2 + R^2}{2d} \tag{B.3}$$

Substituting B.3 in B.1 and solving for Y, half of the length of chord α is obtained

$$Y = \frac{\alpha}{2} = \sqrt{\frac{4d^2R^2 - (d^2 - r^2 + R^2)^2}{4d^2}} \tag{B.4}$$

Assuming similar transmission ranges R=r, the entire chord length is α=$2Y$, thus

$$\alpha = \sqrt{4R^2 - d^2} \tag{B.5}$$

To find the intersection area between the two circles, we take two times the formula of a circular section of radius R

$$A_i = 2R^2 \cos^{-1}(\frac{d}{2R}) - \frac{d}{2}\sqrt{4R^2 - d^2} \tag{B.6}$$

Finally, the normalized intersection area A_n can be described as a function of the transmission range and the normalized distance between the nodes.

$$A_n = 2R^2 \cos^{-1}(\frac{d_n}{2R}) - \frac{d_n}{2}\sqrt{4R^2 - d_n^2} \tag{B.7}$$

Since we are interested in finding an expression for the distance, we have to solve equation B.7 for d_n. The Newton-Rapson Method has been used to find the roots of the function with MatLab

$$f(d_n) - f^{-1}(A_n, d_n) = 0 \tag{B.8}$$

resulting in a family of polynomial expressions of different degrees. We verified that the a polynomial of degree 3 is sufficient for our algorithm.

$$d_n = 0.0239A_n^3 + 0.245A_n^2 - 1.006A_n + 1.9254 \tag{B.9}$$

Equation B.9 now gives us the normalized distance between nodes that share intersecting transmission ranges dependent on the normalized area of this intersection.

B.2 Relating distance to the probable number of nodes in the intersection area

Due to the restricted resources available on real sensor nodes, it is not suitable to design an algorithm that requires in-situ complex mathematical computations. The idea is to map this problem in such a way to create a more light-weight one by demanding the sensor nodes to conclude their distances from the number of their neighboring nodes. Under the assumption of uniformly distributed networks, the number of nodes in the intersection area A_i is proportional to this area. We refer to this quantity of nodes as K_i.

$$A_n \sim p(d, K_i) * K_i \tag{B.10}$$

where $p(d, K_i)$ describes the relationship between distances between nodes and the number of nodes in the intersection area. We obtained this approximation by running a multitude of simulations with a variety of node densities.
The graphs plotted in Figure B.2 have been derived using the `ns-2` network simulator. Into a rectangular area of 400 x 400 meter, a number of nodes ranging from 5 to 100 nodes has been deployed in a uniform fashion. For our approximation, the radio communication range was set to 250 meters. This way, we vary K_i, getting an idea about its behaviour in networks with low, medium and high density. To plot each graph, the average distance between two nodes for a given K_i using 20 simulations has been taken into account. Note that for better comparability both axis of the graph are normalized. As can be seen clearly, the different graphs of Figure B.2 converge in the function $p(d, K_i)$. Therefore, we can use the knowledge about K_i to determine the distance between nodes, remaining aware of the transmission range and the network density.

B.3 Weighting the approximation with local node densities

Although using the distance approximation $p(d, K_i)$ in equation 4.10 is valid, the error of the estimation can be further minimized by taking into account the local node densities of participating nodes. The approximation can be made more accurate by weighting it with the number of nodes that are in the union of their transmission ranges, which we denote as K_u.

$$A_n \sim \frac{p(d, K_i) * K_i}{K_u} * \pi \tag{B.11}$$

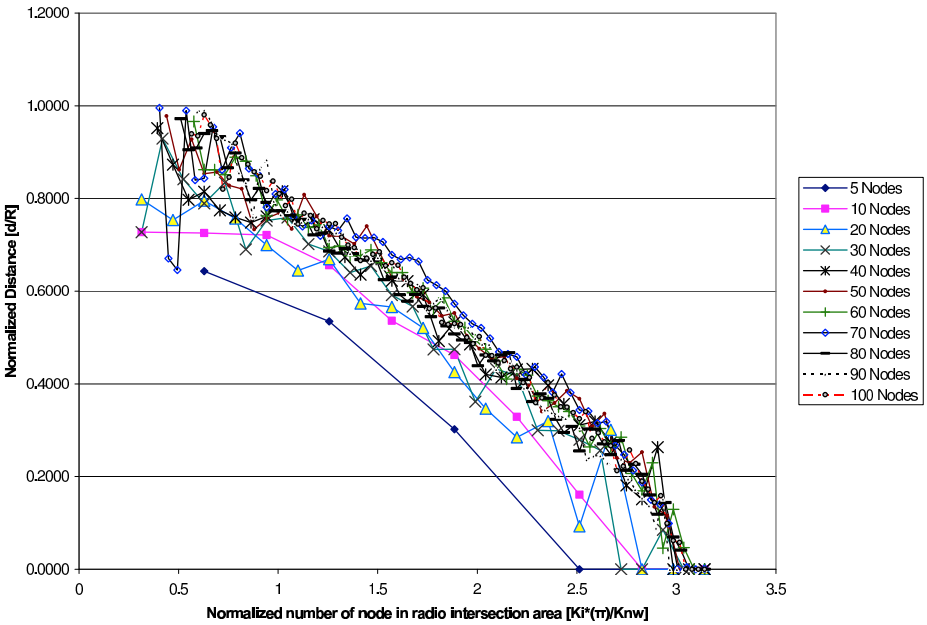


Figure B.2: Approximation graphs to estimate distance depending on (K_i) for different node densities

The equation is multiplied by π due to circular, normalized transmission ranges, thus producing a unit disk graph model. This completes the mathematical steps of WDNI since every node in the network can estimate its distance to adjacent nodes by using equation B.11 in combination with equation 4.10.

APPENDIX C

Here we present the plots obtained by the simulation of the ExDIN and DV-Hop algorithms using a fixed number of 100 nodes. Four reference nodes have been deployed on the corners of a square area which have been varied from 250 meters to 2500 meters. Every graph displays the result of 30 simulation with uniform deployment. The length of network is specified in every figure caption of this section.

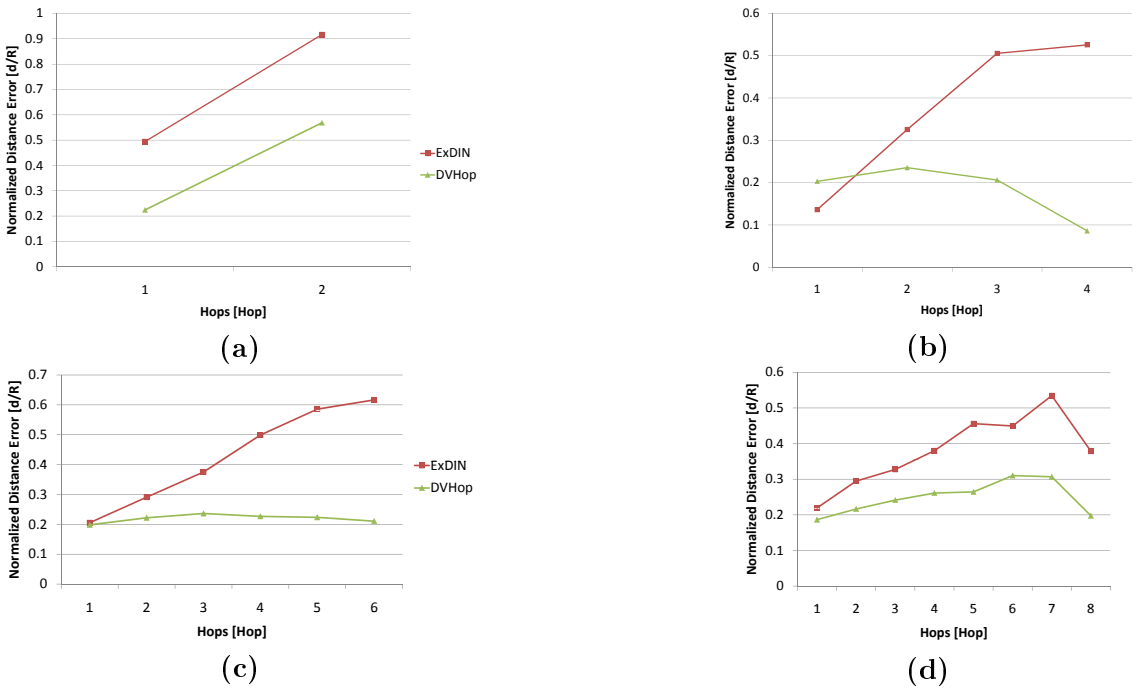


Figure C.1: Comparison of the error resulting from simulations of the ExDIN and DV-Hop techniques for an uniformly distributed network with (a) $L = 250$, (b) $L = 500$, (c) $L = 750$, and (d) $L = 1000$ meters

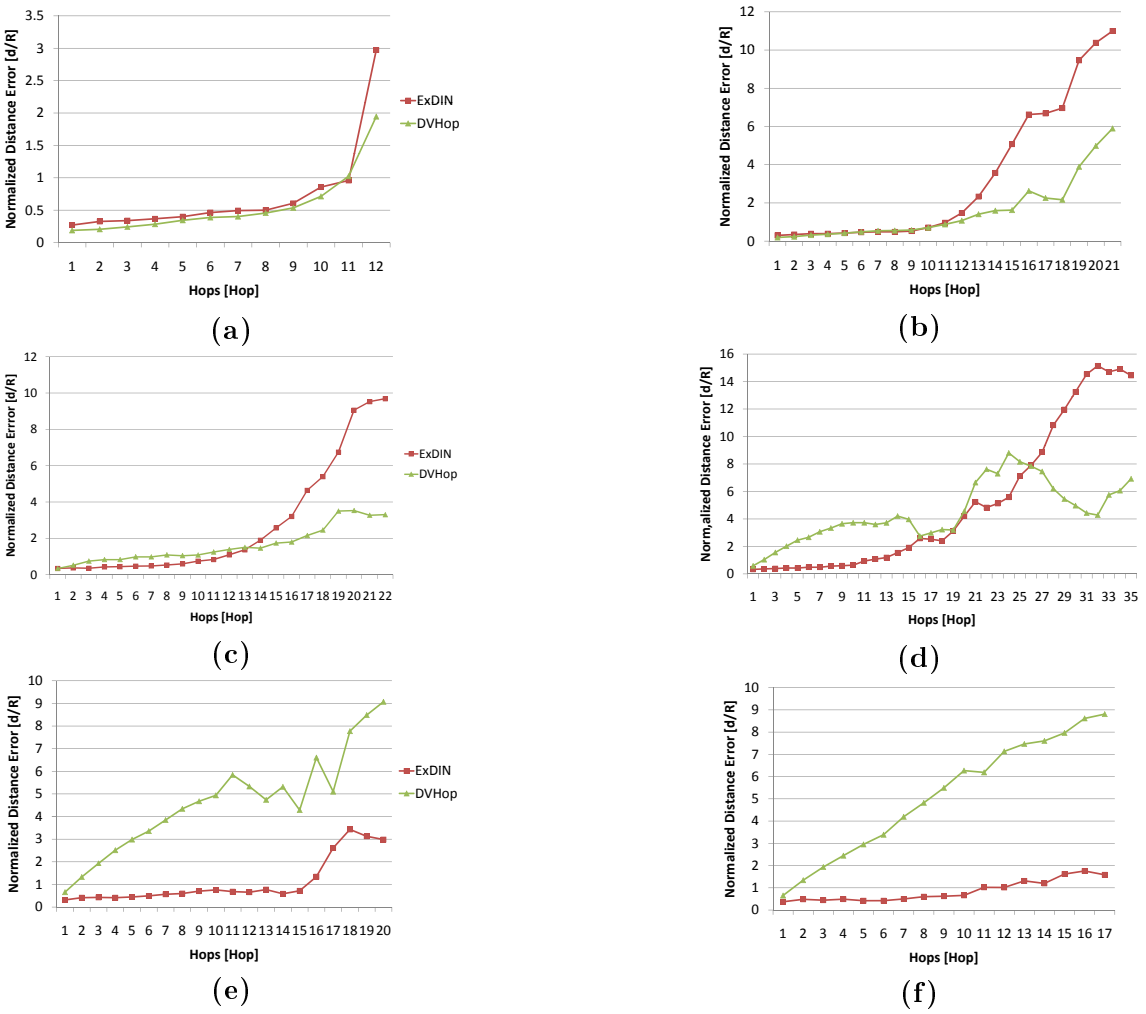


Figure C.2: Comparison of the error resulting from the simulations of the ExDIN and DV-Hop algorithms for a uniformly distributed network with (a) $L = 1250$, (b) $L = 1500$, (c)$L = 1750$, (d) $L = 2000$, (e) $L = 2250$, and (f) $L = 2500$ meters

APPENDIX D

In this section, the curves of normalized distance errors of the FC, FCH and DV-Hop methods using the `ns-2` simulator are presented. Four Landmarks are placed in every corner of the square area. The Length (L) of the square side where the nodes are deployed is varied from 250 to 2500 meters. The average of 10 simulation runs using a fixed number of 100 nodes with different network densities are displayed in each graph.

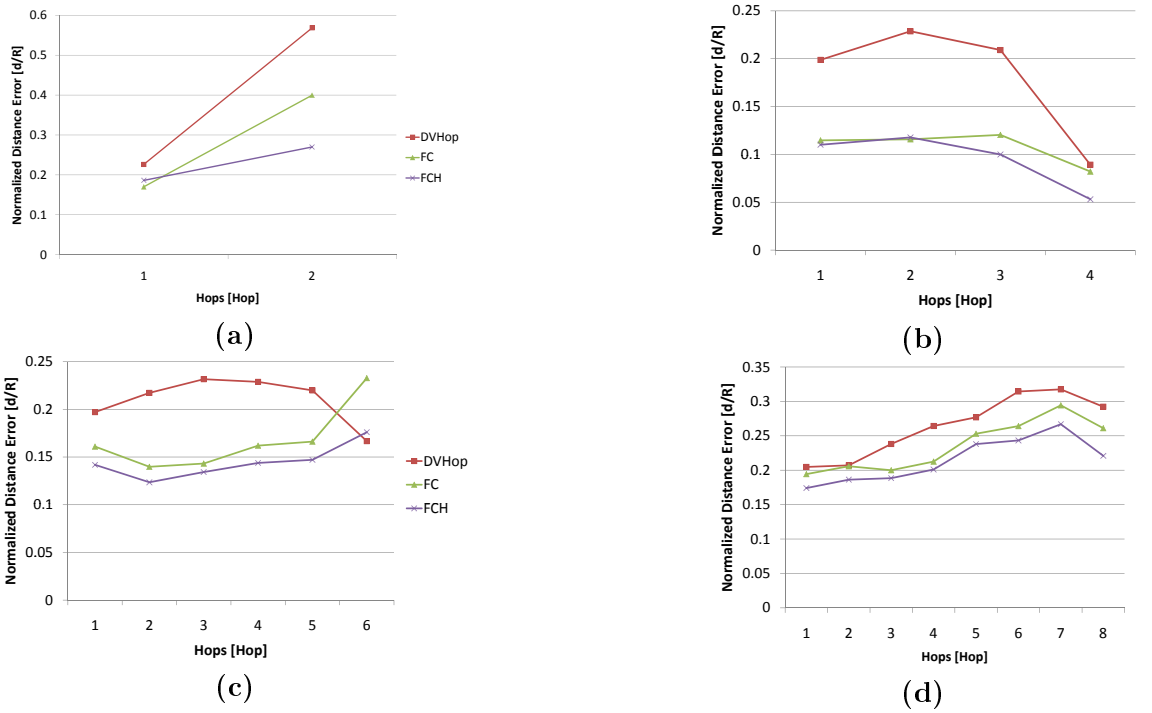


Figure D.1: Comparison of the normalized errors resulting from simulations of the FC, FCH and DV-Hop techniques for an uniformly distributed network with (a) $L = 250$, (b) $L = 500$, (c) $L = 750$, and (d) $L = 1000$ meters

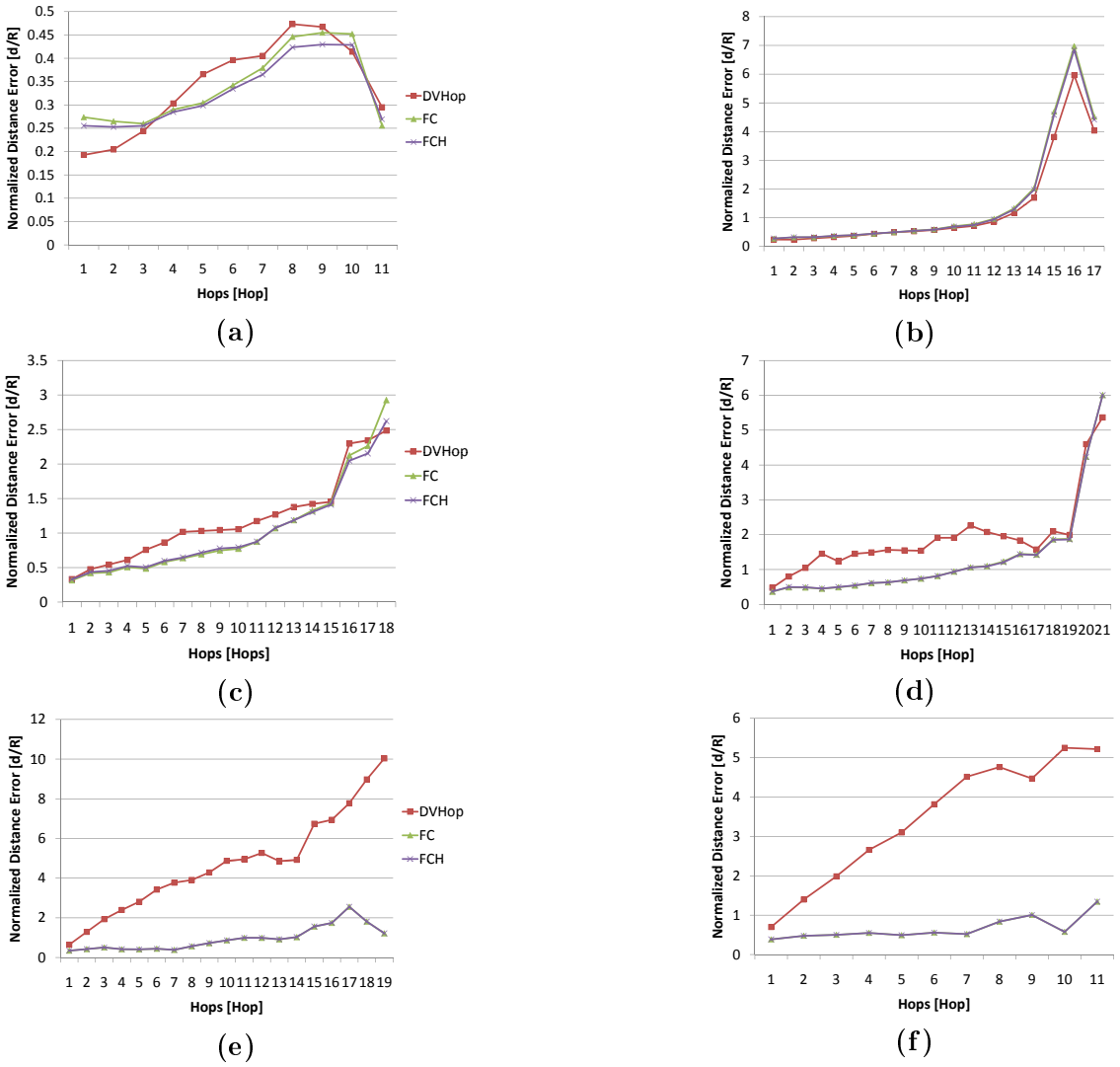


Figure D.2: Comparison of the normalized errors resulting from the simulations of the FC, FCH and DV-Hop algorithms for an uniformly distributed network with (a) $L = 1250$, (b) $L = 1500$, (c) $L = 1750$, (d) $L = 2000$, (e) $L = 2250$, and (f) $L = 2500$ meters

APPENDIX E

In this section, the curves of average position error obtained by using PIV with RND and DIN neighbor distance as well as the multilateration method with RND in combination with ExDIN, DVHop and FCH lagorithms are presented. The simulation were programmed with C++ programming language.

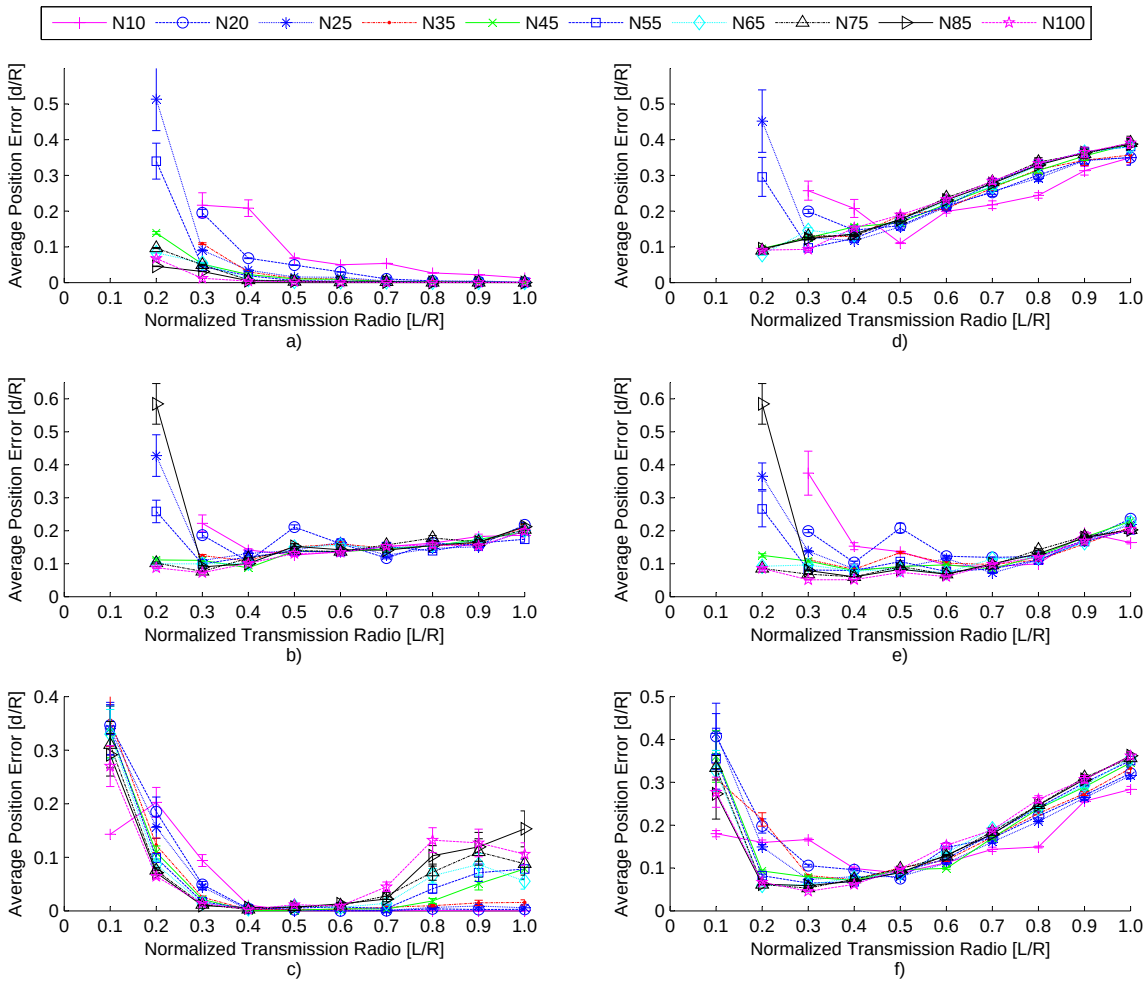


Figure E.1: Position error comparison of a)Multilateration using RND b)Multilateration using DV-Hop c)PIV using RND d)Multilateration using ExDIN e)Multilateration using FCH and f)PIV using DIN Distancces for an uniformly distributed network and 8 Anchors

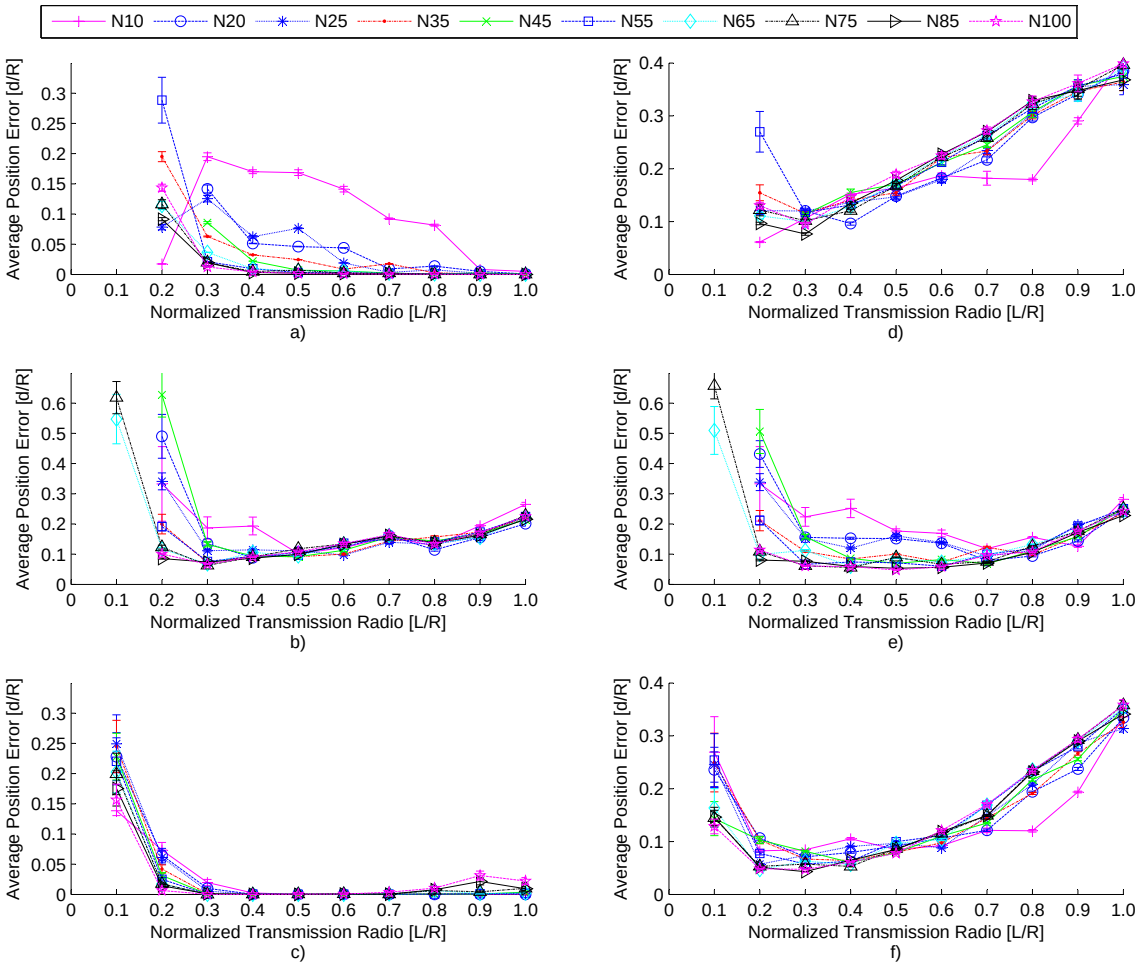


Figure E.2: Position error comparison of a)Multilateration using RND b)Multilateration using DV-Hop c)PIV using RND d)Multilateration using ExDIN e)Multilateration using FCH and f)PIV using DIN Distancces for an uniformly distributed network and 16 Anchors

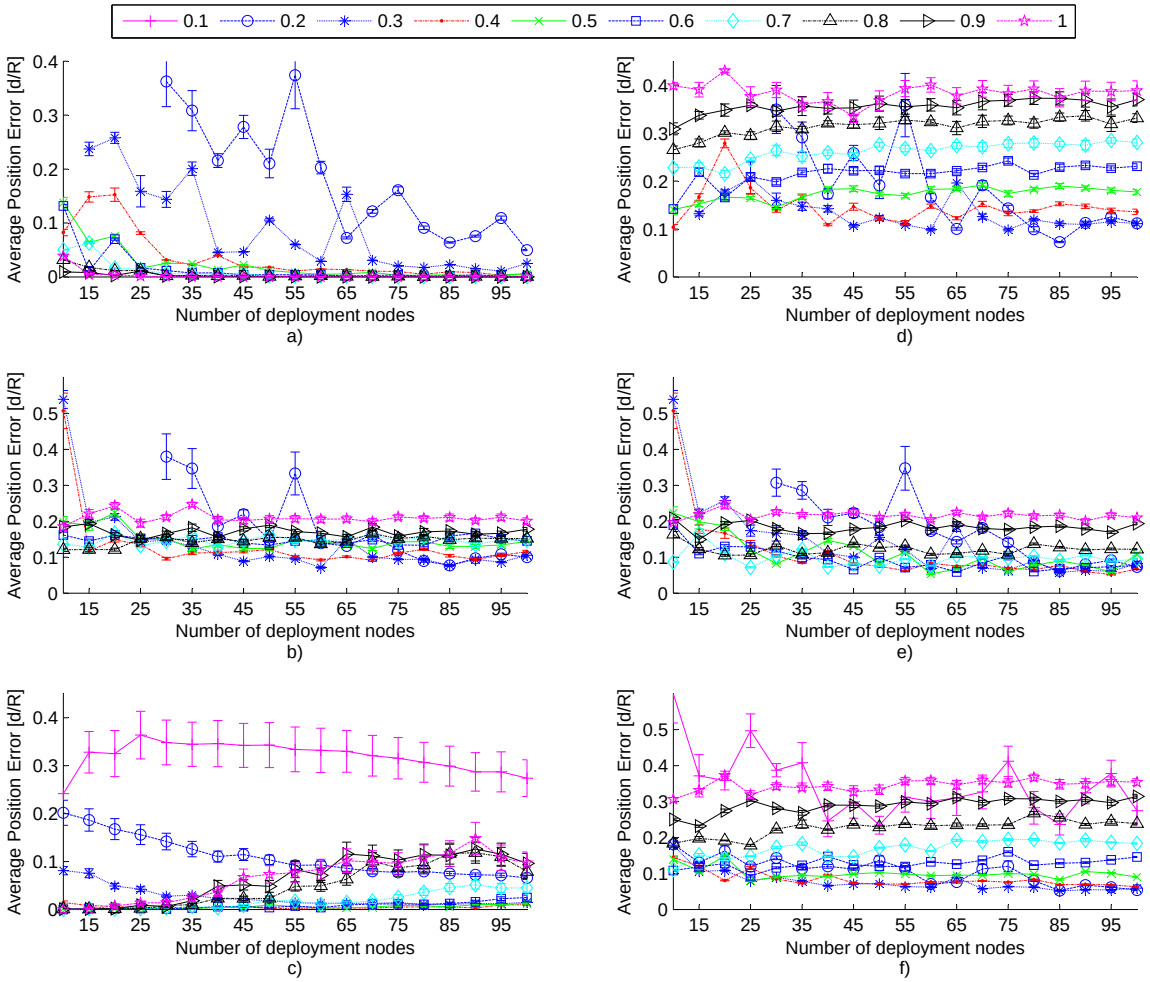


Figure E.3: Comparison of average position error versus number of deployed nodes with a)multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN Distancces for an uniformly distributed network and 8 Anchors

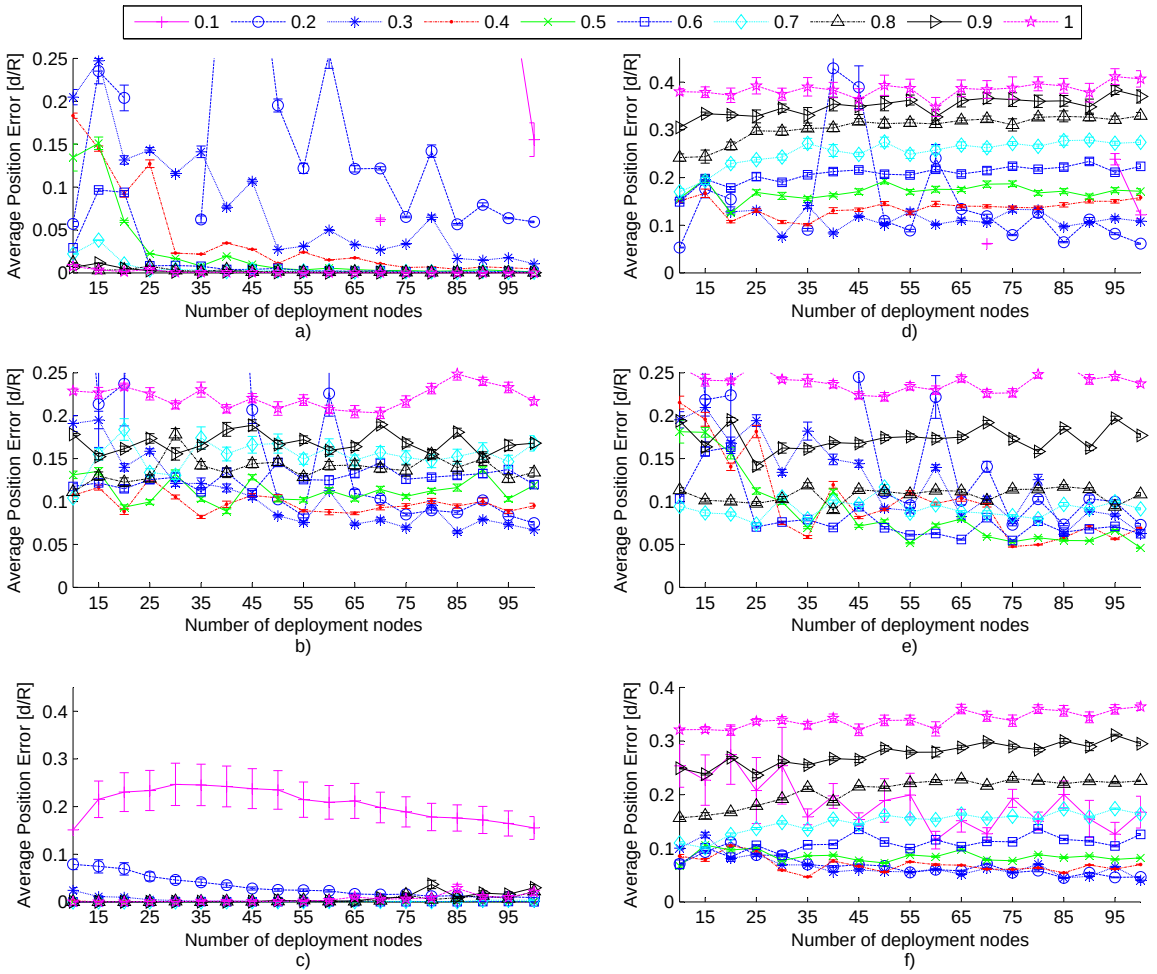


Figure E.4: Comparison of average position error versus number of deployed nodes with a)multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN Distancces for an uniformly distributed network and 16 Anchors

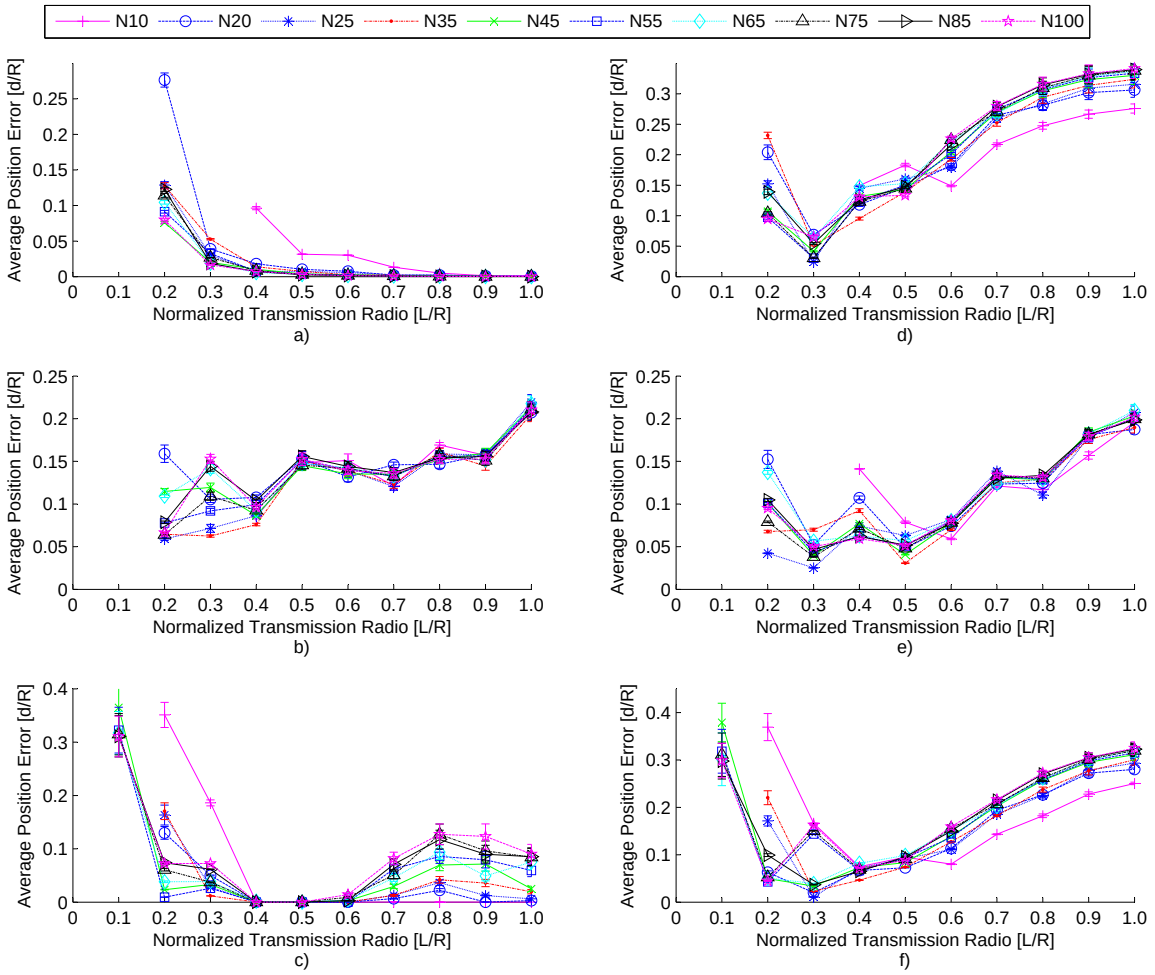


Figure E.5: Average position errors versus normalized transmission radio using a)multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN Distancces for a grid node distribution and 8 landmarks

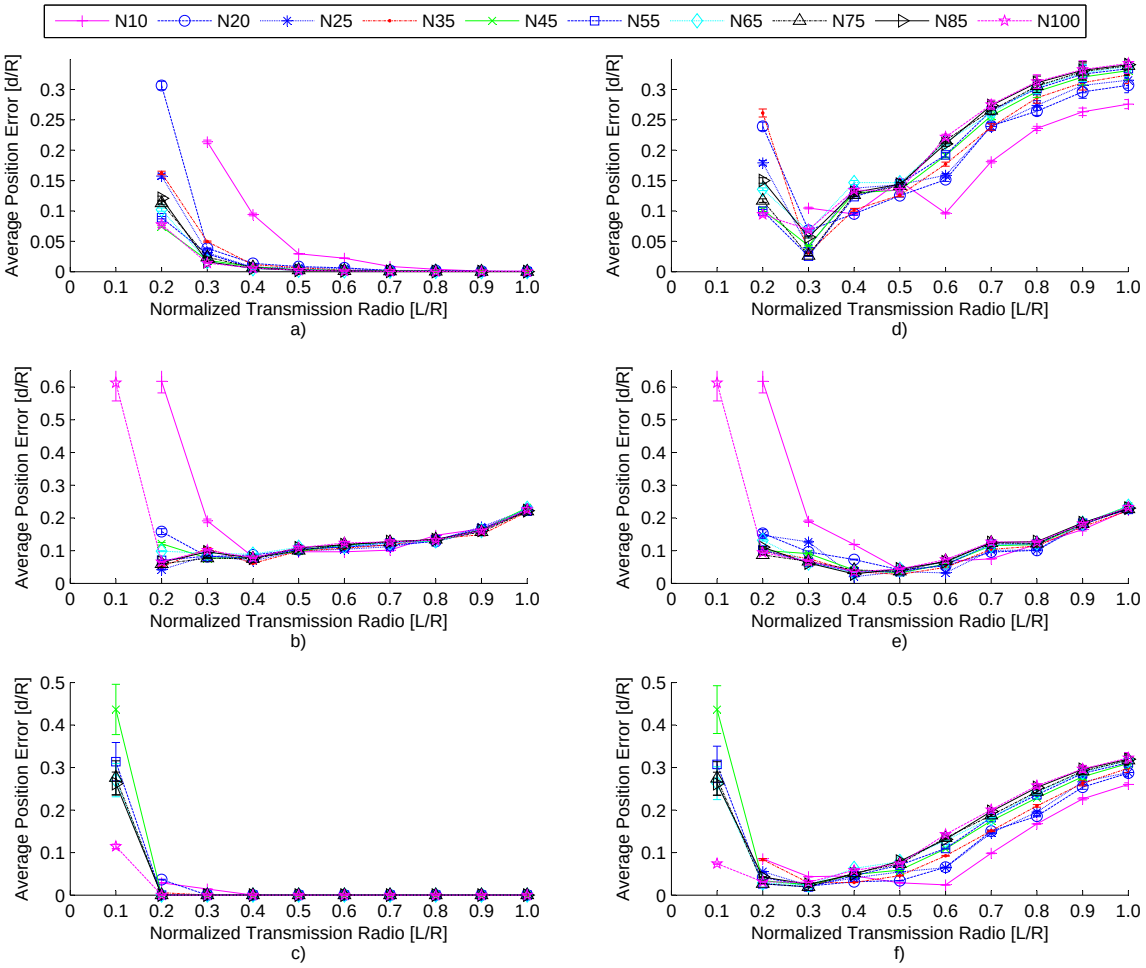


Figure E.6: Average position errors versus normalized transmission radio using a)multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN Distancces for a grid node distribution and 16 landmarks

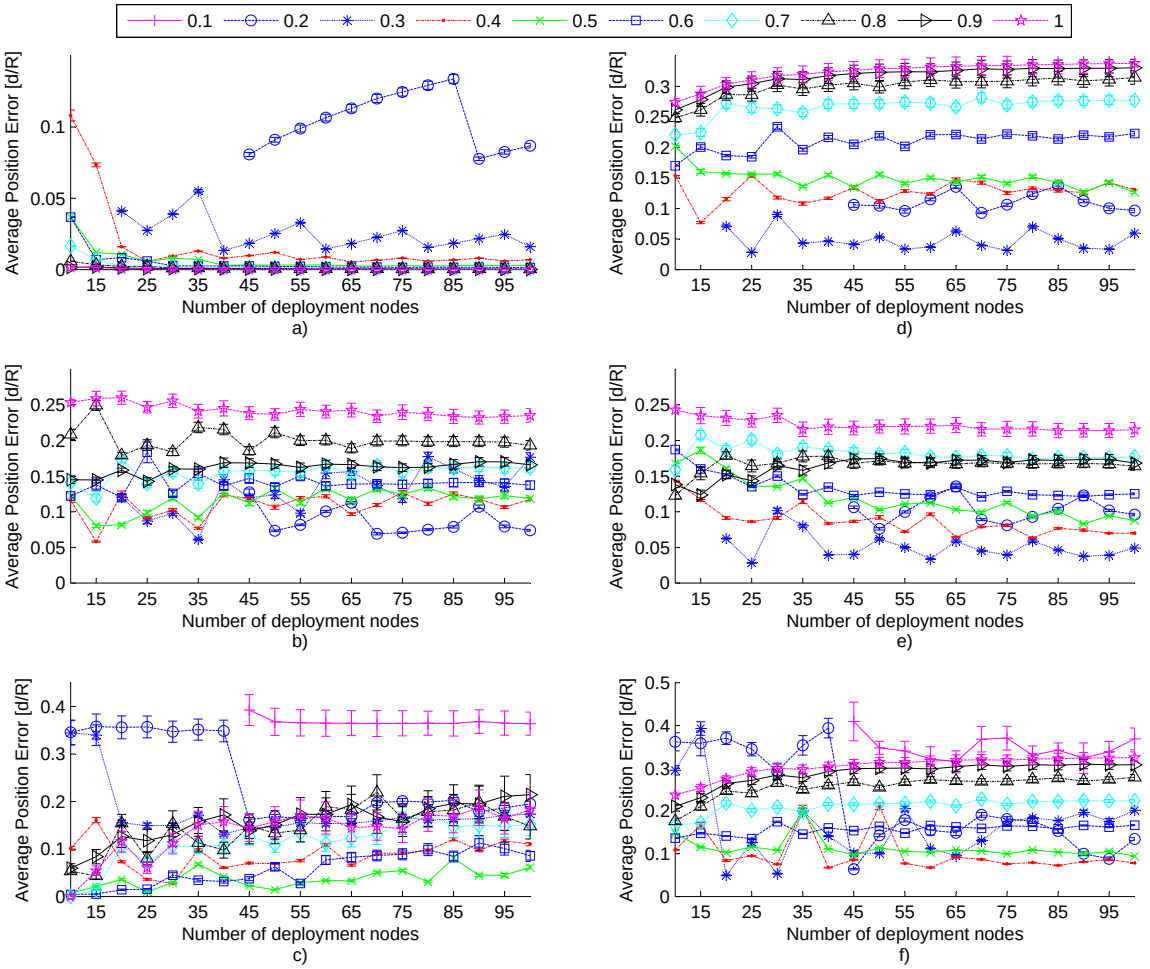


Figure E.7: Average position errors versus number of deployment nodes using a)multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN Distancces for a grid node distribution and 4 landmarks

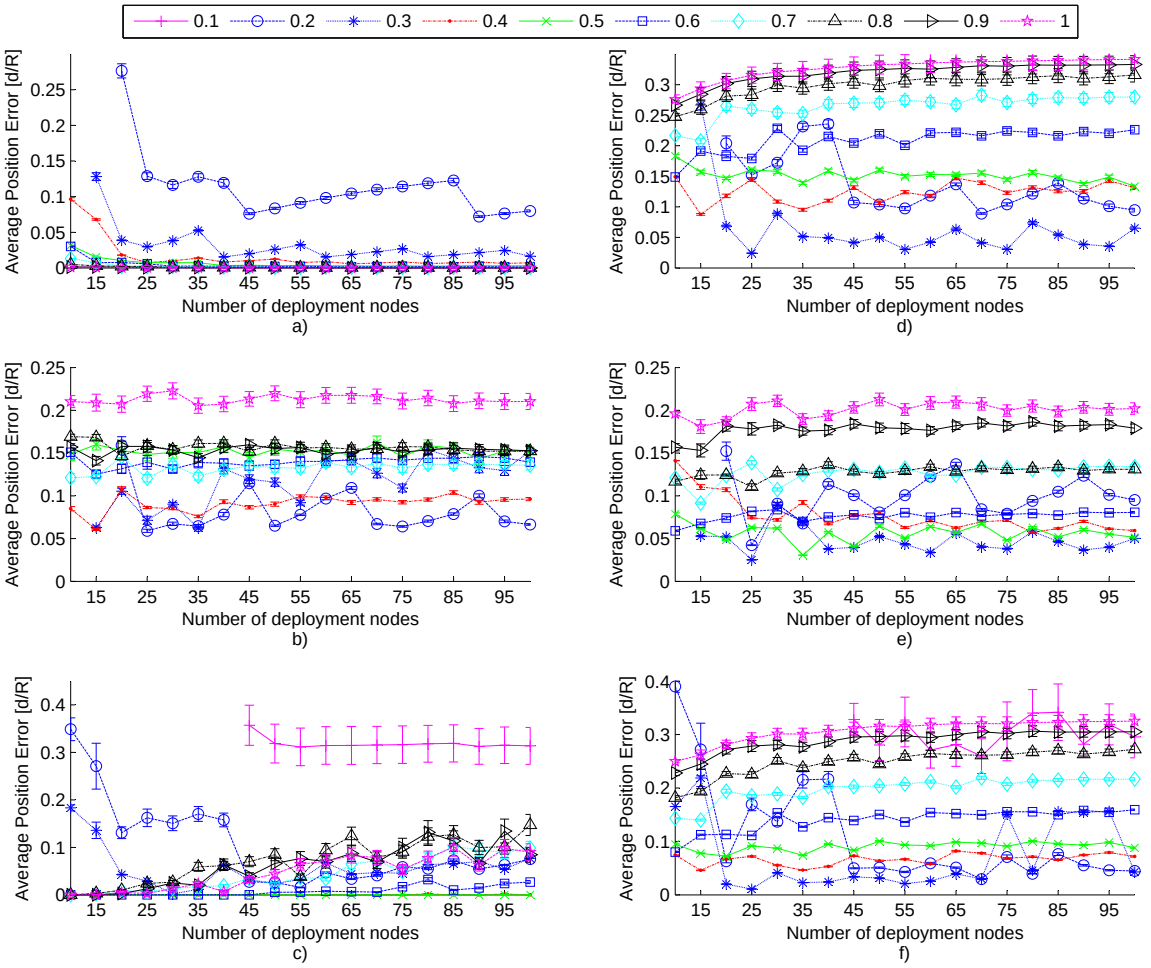


Figure E.8: Average position errors versus number of deployment nodes using a)multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN Distancces for a grid node distribution and 8 landmarks

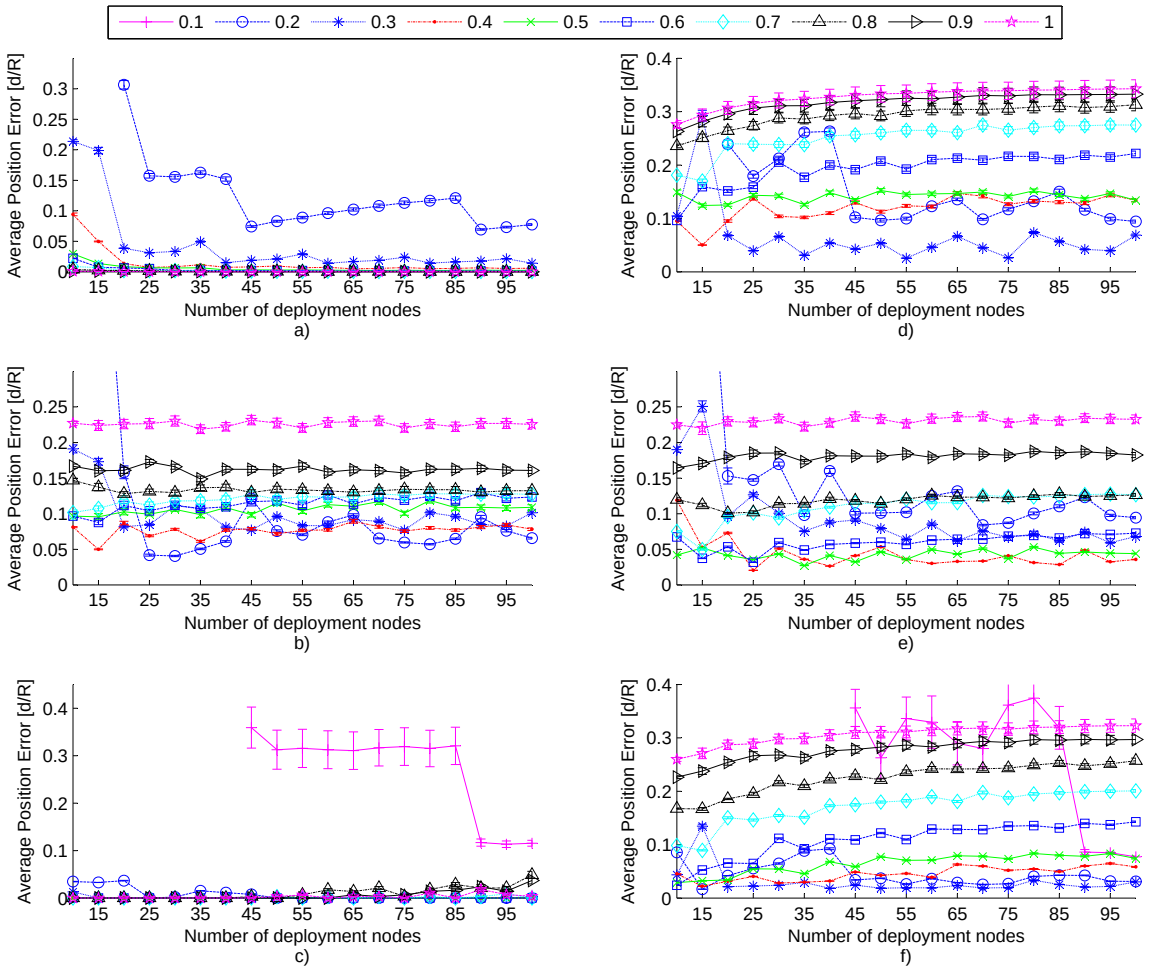


Figure E.9: Average position errors versus number of deployment nodes using a)multilateration using RND b)multilateration using DV-Hop c)PIV using RND d)multilateration using ExDIN e)multilateration using FCH and f)PIV using DIN Distancces for a grid node distribution and 16 landmarks

APPENDIX F

In this section, the curves of average position error versus number of iterations obtained by using PIV with RND and DIN neighbor distance with different number of reference nodes are presented and plots related with the behaviour of the PIV algorithm using real sensor nodes.

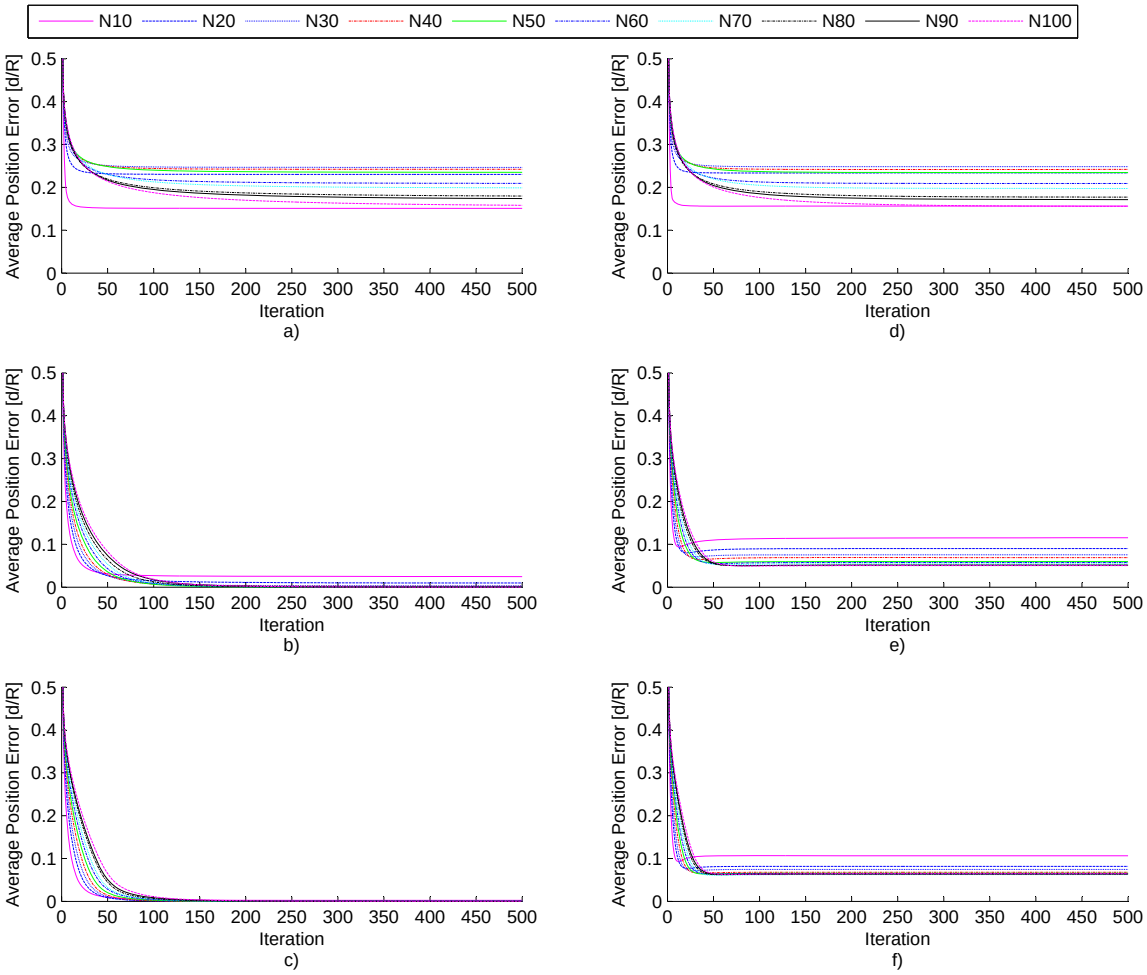


Figure F.1: Average position errors versus iteration using 16 reference nodes with a)0.1R and RND b)0.3R and RND c)0.4R and RND d)0.1R and DIN distances e)0.3R and DIN distances f)0.4R and DIN distances for a uniform node distribution

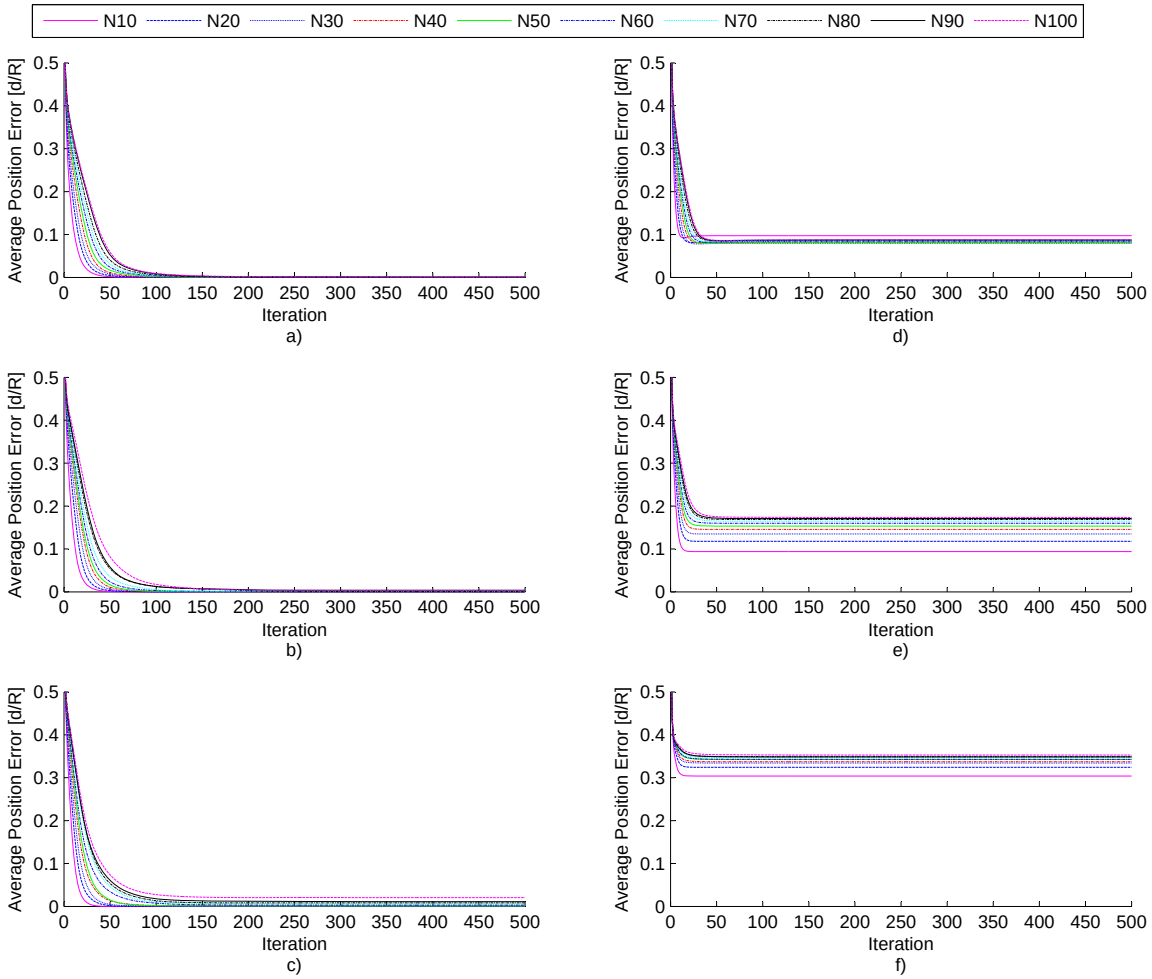


Figure F.2: Average position errors versus iteration using 16 reference nodes with a)0.5R and RND b)0.7R and RND c)1R and RND d)0.5R and DIN distances e)0.7R and DIN distances f)1R and DIN distances for a uniform node distribution

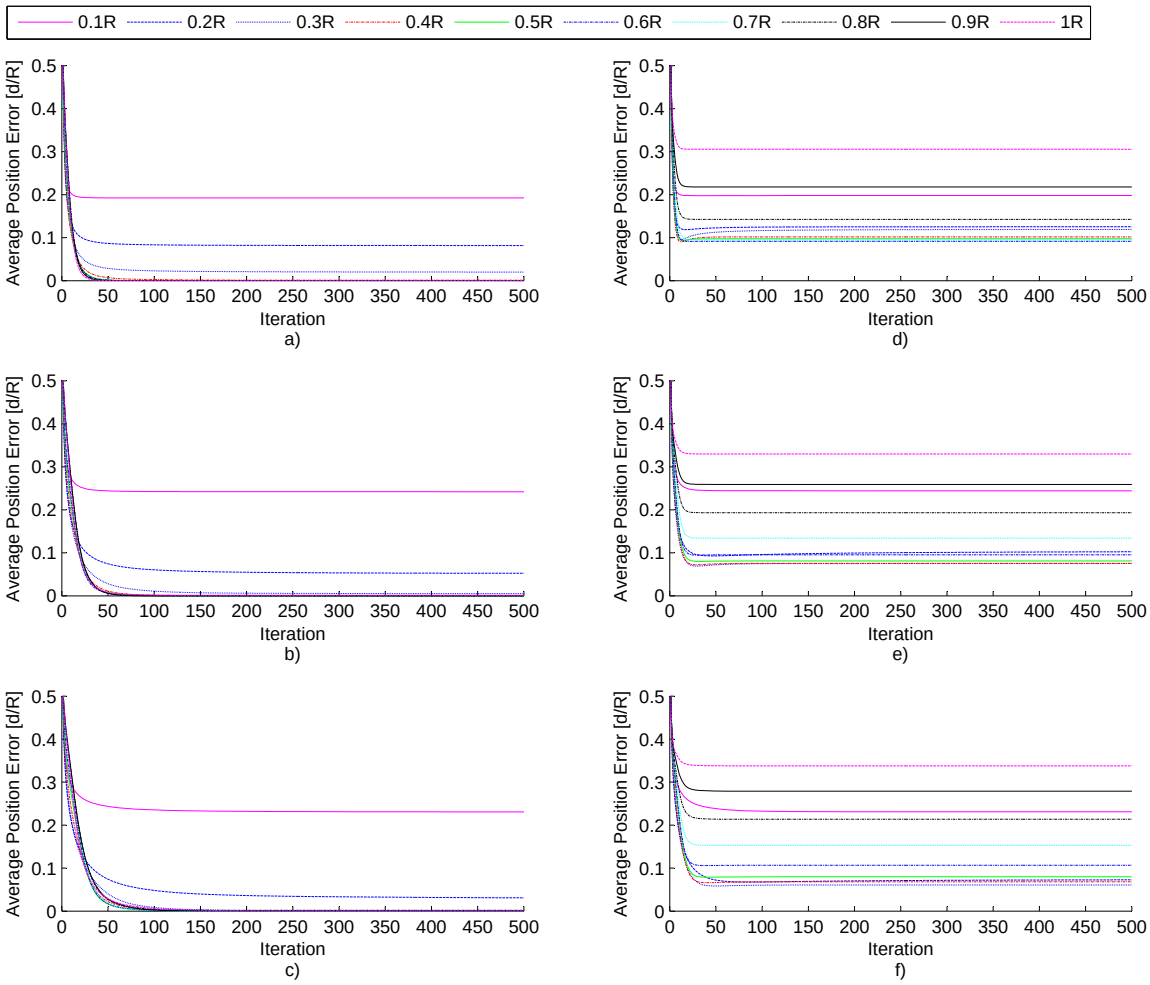


Figure F.3: Average position errors versus iteration using 16 reference nodes with a)10 deployed nodes and RND b)30 deployed nodes and RND c)50 deployed nodes and RND d)10 deployed nodes and DIN distances e)30 deployed nodes and DIN distances f)50 deployed nodes and DIN distances for a uniform node distribution

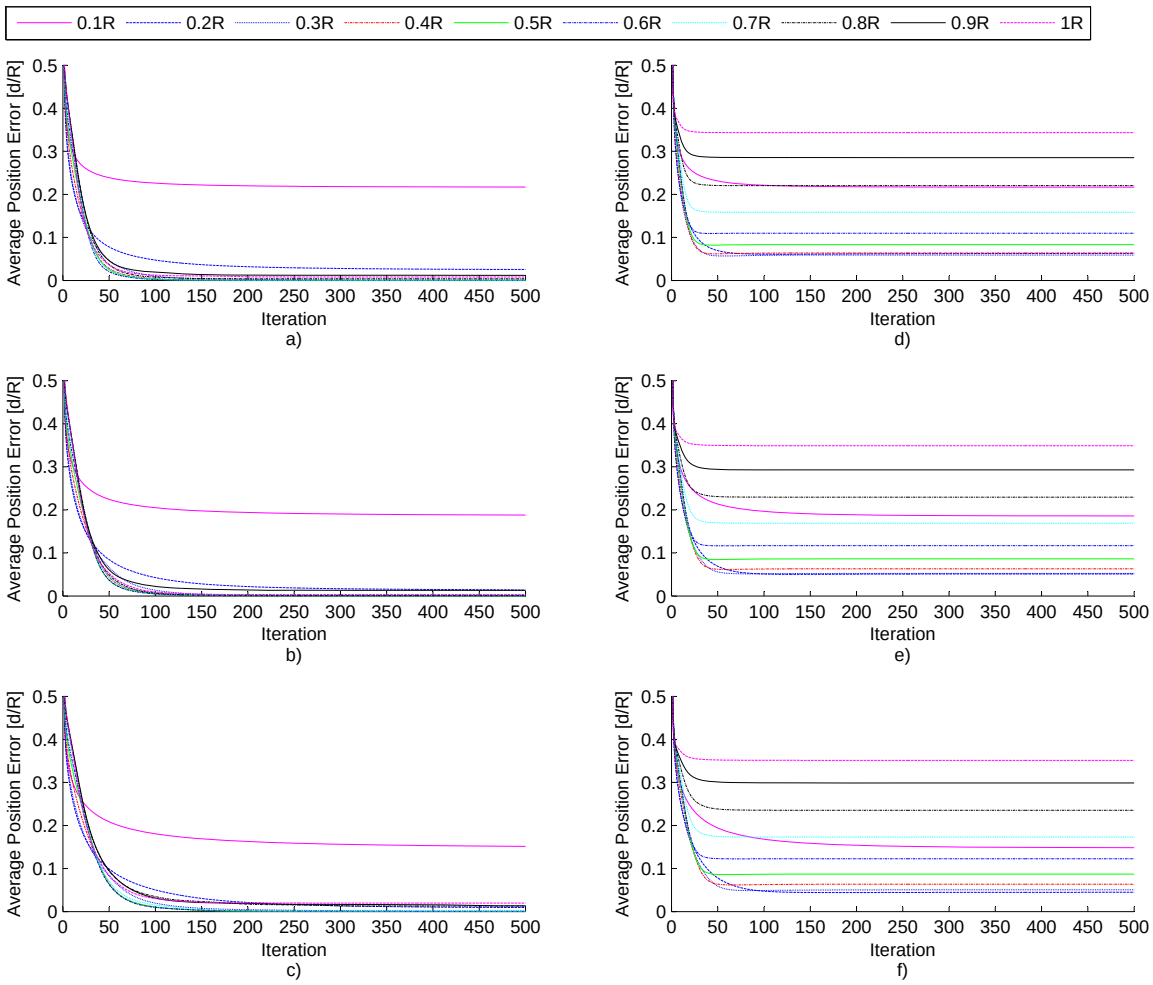
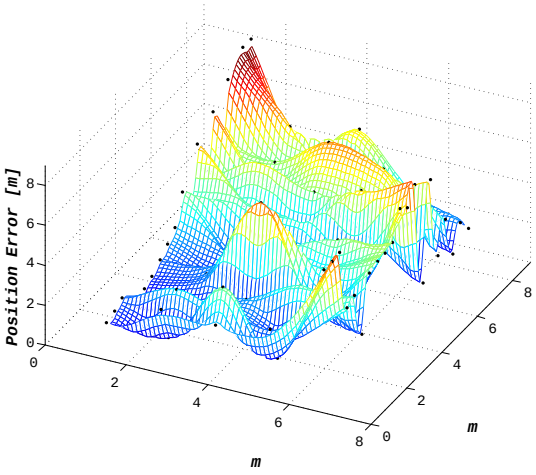
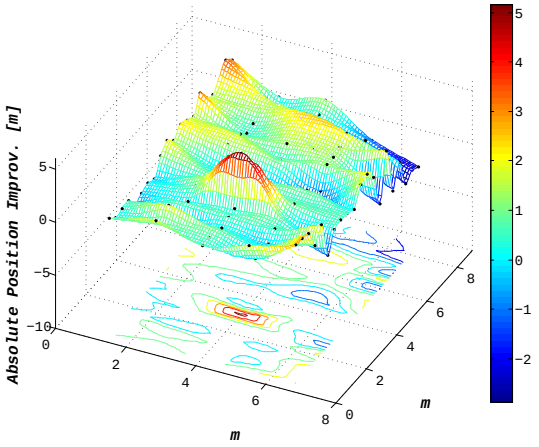


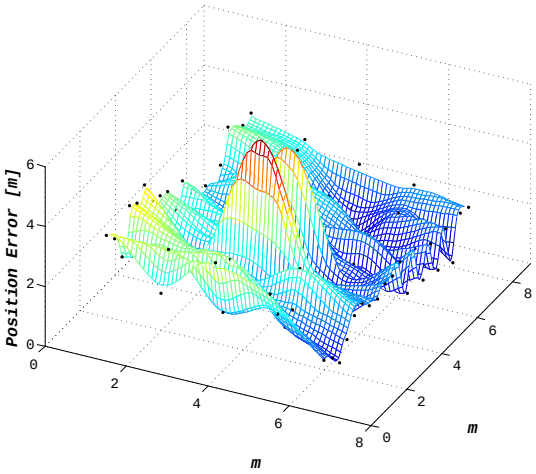
Figure F.4: Average position errors versus iteration using 16 reference nodes with a)10 deployed nodes and RND b)30 deployed nodes and RND c)50 deployed nodes and RND d)10 deployed nodes and DIN distances e)30 deployed nodes and DIN distances f)50 deployed nodes and DIN distances for a uniform node distribution



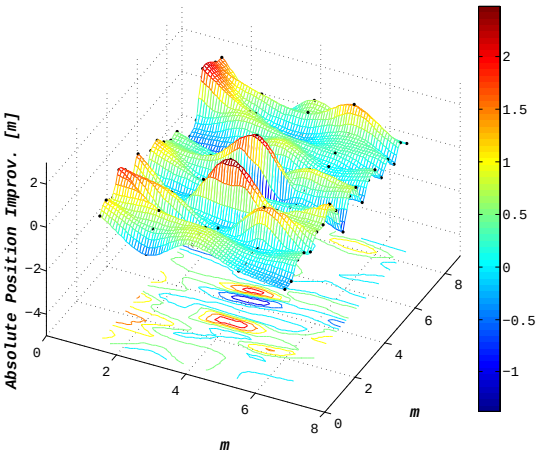
(a) DV-Hop - Initial Pos. Error



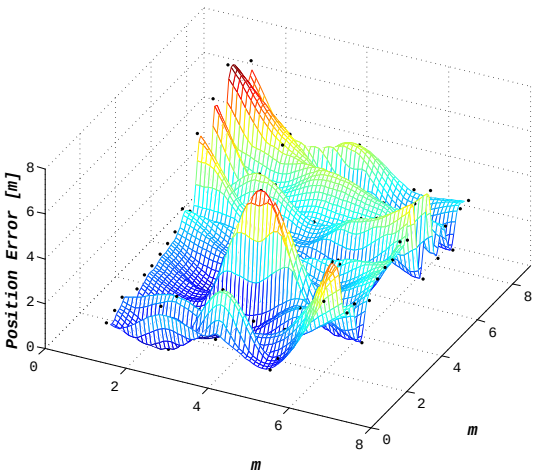
(b) DV-Hop - PIV Pos. Improv.



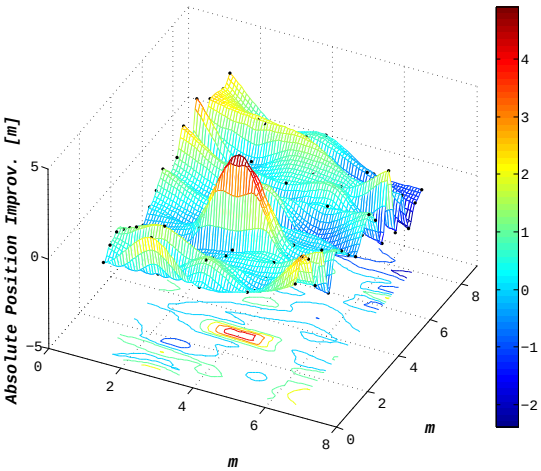
(c) ExDIN/DV-Dist - Initial Pos. Error



(d) ExDIN/DV-Dist - PIV Pos. Improv.



(e) FCH - Initial Pos. Error



(f) FCH - PIV Pos. Improv.

Figure F.5: Spatial distribution comparison of initial position error vs PIV location improvement using DV-Hop, DV-Dist, and FCH algorithms in a horseshoe configuration with 4 anchors and 100 unknown nodes

APPENDIX G

In this section, different node configuration used in the real experimental testbed are shown. The layouts that used four reference nodes can be obtained removing the landmarks from number 5 to 8 and sustracting four for every ID of the unknown deployed nodes.

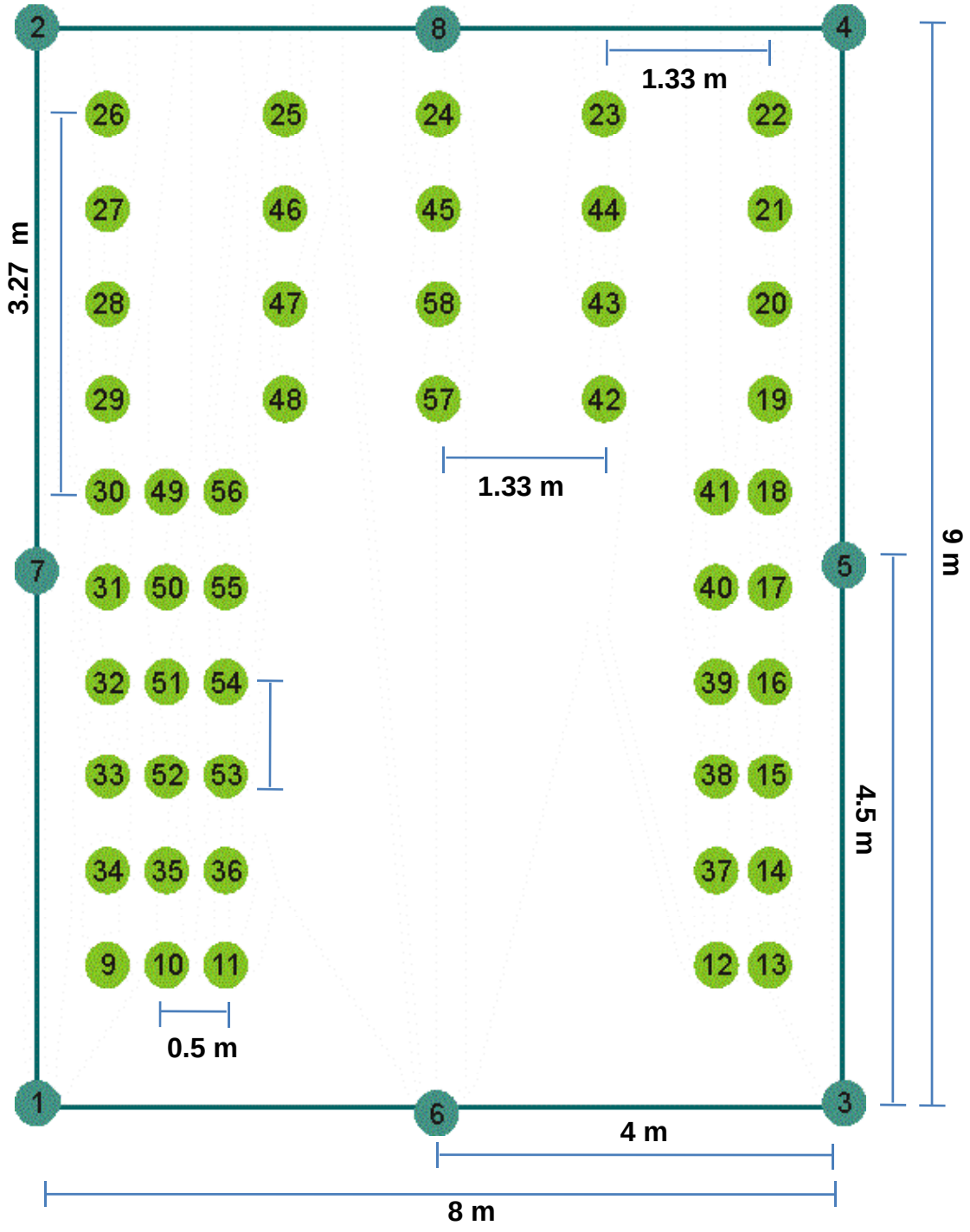


Figure G.1: Horseshoe layout using 50 deployed nodes and 8 landmarks

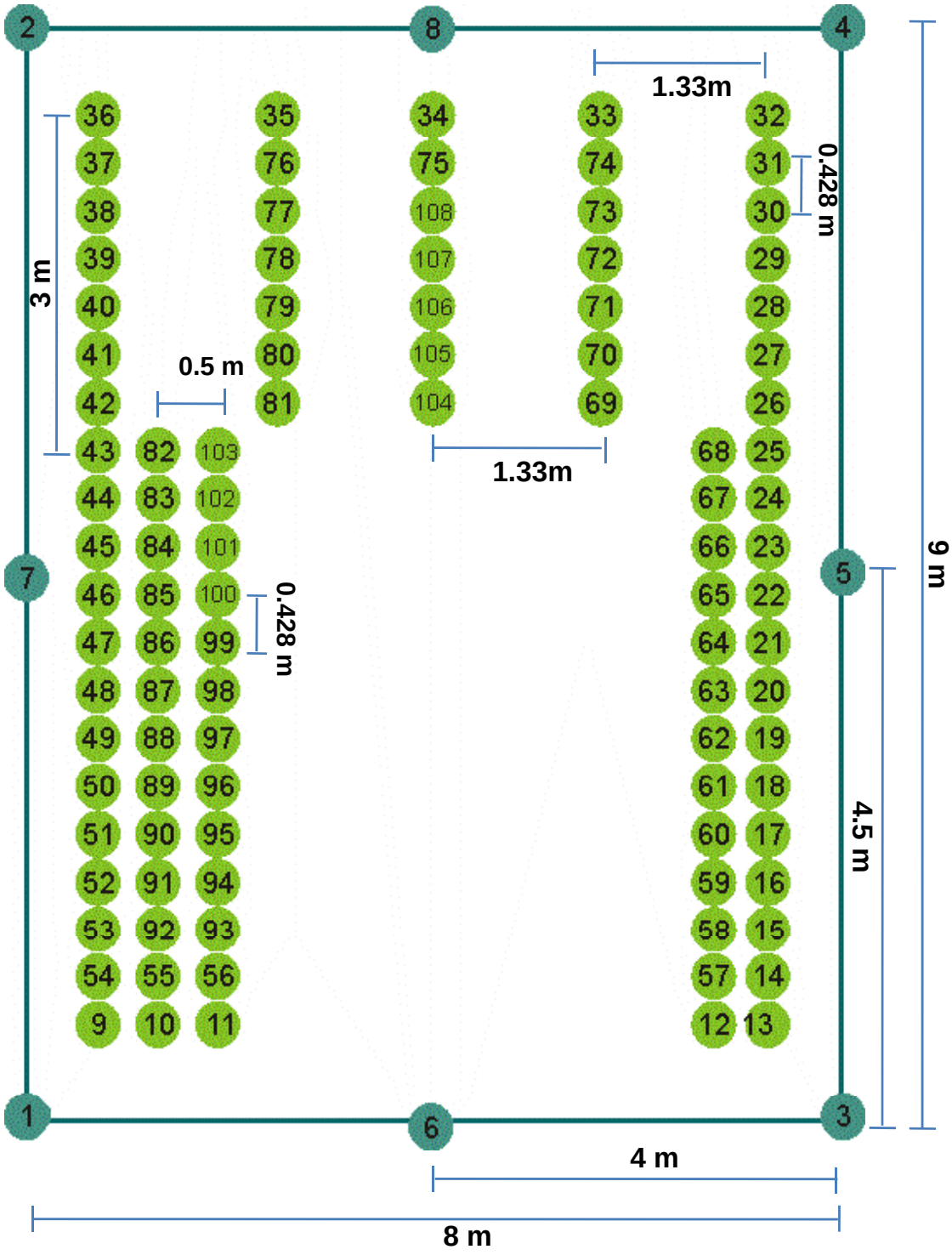


Figure G.2: Horseshoe layout using 100 deployed nodes and 8 landmarks

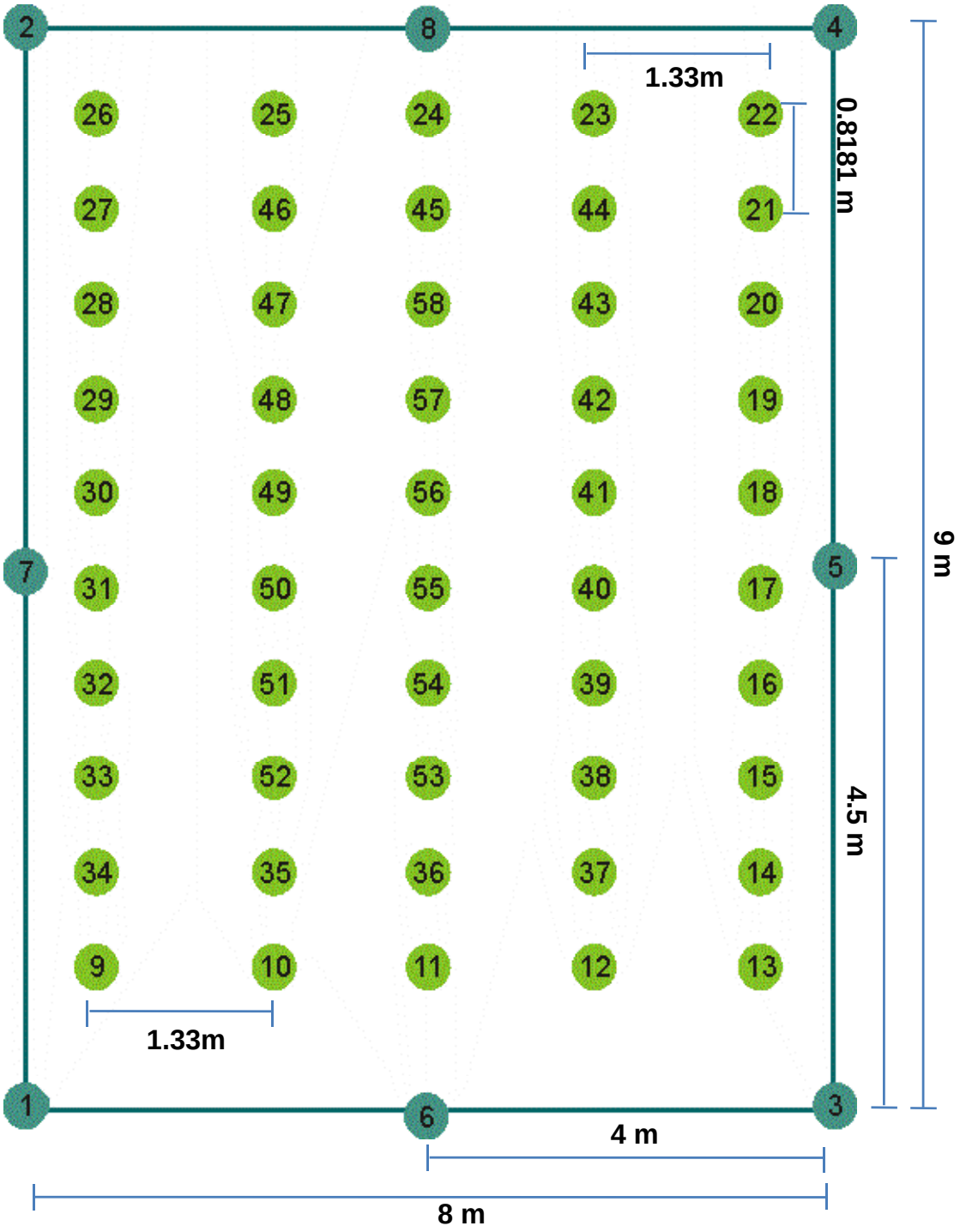


Figure G.3: Uniform node distribution using 50 unknown nodes and 8 reference nodes

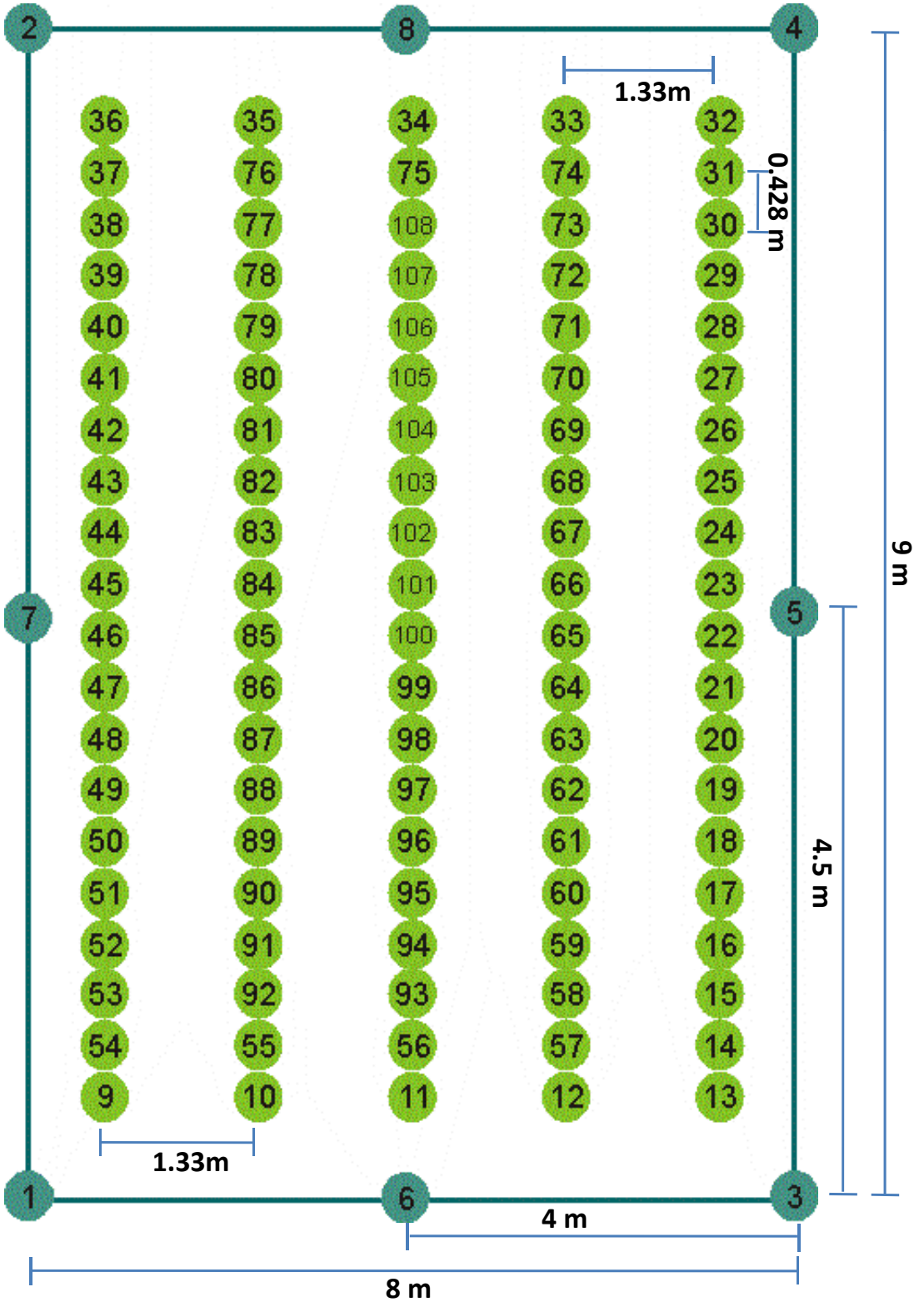


Figure G.4: Uniform node distribution using 100 unknown nodes and 8 reference nodes

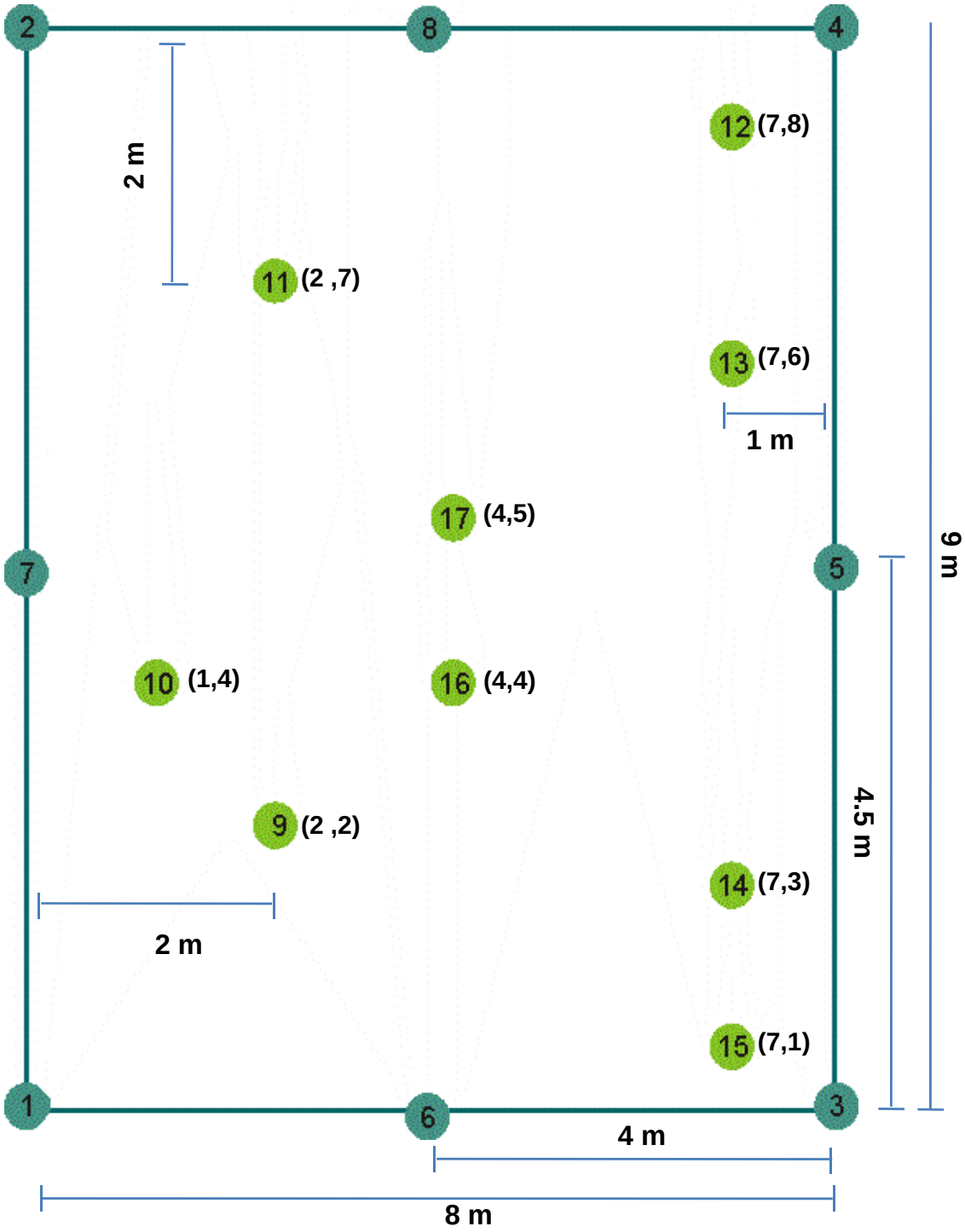


Figure G.5: Low node density distribution using 9 unknown nodes and 8 reference nodes

Bibliography

[1] Aeroscout-Company. Homepage of the aeroscout project. `http://www.aeroscout.com/`, August 2008.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. In *Journal Communications Magazine*, pages 102–114. IEEE, 2002.

[3] R. Amsalem and D. Aljadeff. Radio-frequency identification (RFID) tag employing unique reception window and method therefor, November 13 2007. US Patent 7,295,115.

[4] B. D. O. Anderson, P. N. Belhumeur, T. Eren, D.K. Goldenberg, A.S. Morse, W. Whiteley, and Y.R. Yang. Graphical properties of easily localizable sensor networks. *submitted for publication in Wireless Networks, Journal of Mobile Communication, Computation and Information*, 2005.

[5] J. Aspnes, W. Whiteley, and Y. Yang. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, 2006.

[6] P. Bahl and V. N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies.*, pages 775–784, 2000.

[7] A. Barabell. Improving the resolution performance of eigenstructure-based direction-finding algorithms. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 8:336–339, 1983.

[8] S.R. Bible, M. Zyda, and D. Brutzman. Using Spread Spectrum Ranging Techniques for Position Tracking in a Virtual Enviroment. *Second IEEE Workshop Networked*, 2000.

[9] M. Broxton, J. Lifton, and J. Paradiso. Localizing a sensor network via collaborative processing of global stimuli. In *Proceedings of the Second European Workshop on Wireless Sensor Networks*, pages 321–332. IEEE, January 2005.

[10] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 7(5):28–34, 2000.

[11] N. Bulusu, J. Heidemann, and D. Estrin. Density Adaptive for Beacon placement in Wireless Sensor Networks. *Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS-21)*, pages 489–498, 2001.

[12] C.Buschmann, H.Hellbrück, S.Fischer, A.Kröller, and S.P. Fekete. Radio propagation-aware distance estimation based on neighborhood comparison. In *European Workshop*

on Sensor Networks, volume 4373 of *Springer Lecture Notes in Computer Science*, pages 325–340, 2007.

[13] C. Chien. Location position system for relay assisted tracking, January 28 2003. US Patent 6,512,478.

[14] Fedaral Communication Commission. Homepage of the e911 project. `http://www.fcc.gov/pshs/services/911-services/`, August 2008.

[15] Ubisense Company. Homepage of the ubisense systems. `http://www.ubisense.net/`, August 2008.

[16] Rosum Corporation. Homepage of the rosum project. `http://www.rosum.com/`, Agust 2008.

[17] D.Koks. Numerical calculations for passive geolocation scenarios. *Research Report, Australia*, Book on line, 2007.

[18] L. Doherty, K. S. J. Pister, and L. El Ghaoui. Convex Position Estimation in Wireless Sensor Networks. *IEEE INFOCOM, Anchorage, AK, April*, 2001.

[19] T. Eren, D.A. Goldenberg, W. Whiteley, R. Yang, A. Morse, B. Anderson, and P. Belhumeur. Rigidity and randomness in network localization. *IEEE INFOCOM*, 4:2673–2684, 2005.

[20] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. *Pervasive Computing, IEEE*, 1(1):59–69, 2002.

[21] K. Terfloth F. López-Villafuerte and J. Schiller. Using network density as a new parameter to estimate distance. In *The Seventh International Conference on Networking, ICN08*, page 6, Cancun, Mexico, 2007. IEEE Press.

[22] FU-Berlin. Homepage of the scatterweb project. `http://www.scatterweb.mi.fu-berlin.de`, July 2007.

[23] B. Fidan G. Mao and B. D. O. Anderson. Wireless sensor network localization techniques. *Comput. Netw.*, 51(10), 2007.

[24] United States Coast Guard. Homepage of the loran system. `http://www.navcen.uscg.gov/loran/default.htm`, August 2008.

[25] J. González Guerrero. Radio Signal Strength Multilateration Approach in Wireless Sensor Networks. *Pre-degree work*, 2007.

[26] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 59–68, New York, NY, USA, 1999. ACM.

[27] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95, New York, NY, USA, 2003. ACM Press.

[28] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher. Range-free Localization Schemes in Large Scale Sensor Networks. *proceedings of the 9th annual international conference on Mobile computing and networking*, 2003.

[29] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[30] J. Hightower, R. Want, and G. Borriello. SpotON: An indoor 3d location sensing technology based on RF signal strength. UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA, February 2000.

[31] S. S. Intille, K. Larson, J.S. Beaudin, J. Nawyn, E. Munguia Tapia, and P. Kaushik. A living laboratory for the design and evaluation of ubiquitous computing technologies. *Conference on Human Factors in Computing Systems*, pages 1941–1944, 2005.

[32] E. D. Kaplan and C. Hegarty. Understanding GPS: Principles and Applications Second Edition. *Artech House*, 680, 2005.

[33] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005.

[34] M. Kaveh and A. Bassias. Threshold extension based on a new paradigm for music-type estimation. *International Conference on Acoustics, Speech and Signal Processing*, 5:2535–2538, 1990.

[35] J. Krumm, G. Cermak, and E. Horvitz. RightSPOT: A Novel Sense of Location for a Smart Personal Object. *Ubiquitous Computing: 5th International Conference, Seattle, Wa, Usa, October 12-15*, 212, 2003.

[36] K. Laasonen, M. Raento, and H. Toivonen. Adaptive On-Device Location Recognition. *Pervasive Computing: Second International Conference, Pervasive 2004, Linz/Vienna, Austria, April 18-23*, 2004.

[37] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place lab: Device positioning using radio beacons in the wild. In *Proceedings of PERVASIVE 2005, Third International Conference on Pervasive Computing*, Munich, Germany, 2005.

[38] K. Langendoen and N. Reijers. Distributed Localization in Wireless Sensor Netwroks: a Quantitative Comparison. *Computer Networks: The International Journal of Computer and Telecommunications Networking, 2003, New York, USA, pages 499-518*, 2003.

[39] J. Y. Lee and R. A. Scholtz. Ranging in a dense multipath environment using an UWB radio link. *Selected Areas in Communications, IEEE Journal on*, 20(9):1677–1683, 2002.

[40] D. Lenz. Homepage of the ekahau project. `http://www.ekahau.com/`, August 2008.

[41] F. López-Villafuerte and J. Schiller. Din: An ad-hoc algorithm to estimate distances in wireless sensor networks. In *7th International Conference on AD-HOC Networks and Wireless (AD-HOC NoW 2008)*, page 14, Sophia Antipolis, France, September 2008. IEEE Press.

[42] D. Lymberopoulos, Q. Lindsey, and A. Savvides. An empirical characterization of radio signal strength variability in 3-d ieee 802.15.4 networks using monopole antennas. In *EWSN*, pages 326–341, 2006.

[43] M. Maróti, P. Völgyesi, S. Dóra, B. Kusỳ, A. Nádas, Á. Lédeczi, G. Balogh, and K. Molnár. Radio interferometric geolocation. *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 1–12, 2005.

[44] M. Limon Mendoza, V. H. Zarate Silva, and A. Perez Diaz. An efficient algorithm for localization in wireless sensor networks based on internal array of nodes within cells. In *ICPPW '05: Proceedings of the 2005 International Conference on Parallel Processing Workshops*, pages 405–412, Washington, DC, USA, 2005. IEEE Computer Society.

[45] M. Limon Mendoza, V. H. Zarate Silva, and Arturo Perez Diaz. An efficient algorithm for localization in wireless sensor networks based on internal array of nodes within cells. *Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on*, pages 405–412, 2005.

[46] Microsoft-Research. Homepage of the easyliving project. http://research.microsoft.com/easyliving, July 2007.

[47] motionstarwireless. Homepage of the motionstarwireless project. `http://www.ascension-tech.com/products/motionstarwireless.php`, September 2007.

[48] R. Nagpal. Organizing a Global Coordinate System from Local Information on an Amorphous Computer. Technical Report AIM-1666, MIT A.I. Laboratory, 1999.

[49] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. *Information Processing in Sensor Networks: Second International Workshop, Ipsn 2003, Palo Alto, Ca, Usa, April 22-23, 2003: Proceedings*, 2003.

[50] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. pages 105–111, New York, NY, USA, 2002. ACM.

[51] D. Niculescu and B. Nath. Ad-Hoc Positioning Systems (APS). *Proceedings of IEEE GLOBECOM*, 1:25–29, 2001.

[52] D. Niculescu and B. Nath. Dv based positioning in ad hoc networks, 2003.

[53] M. O'Dell, R. O'Dell, M. Wattenhofer, and R. Wattenhofer. Lost in Space Or Positioning in Sensor Networks. *REALWSN, June*, 2005.

[54] Y. Ohta, M. Sugano, and M. Murata. Autonomous localization method in wireless sensor networks. In *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 379–384, Washington, DC, USA, 2005. IEEE Computer Socety.

[55] PlaceLab Organization. Homepage of the placelab project. `http://www.placelab.org/`, August 2008.

[56] D. Papadogkonas, G. Roussos, and M. Levene. A Navigation Engine for Ubicomp Environments. *The 4th UK-Ubinet Workshop*, pages 10–12, 2006.

[57] C. Peng, G. Shen, Y.Zhang, Y. Li, and K. Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 1–14. ACM, 2007.

[58] J. W. Pierre and M. Kaveh. Experimental evaluation of higher-resolution direction-finding algorithms using a calibrated sensor array testbed. *Digital Signal Processing*, 5:243–254, 1995.

[59] N. B. Priyantha. *The Cricket Indoor Location System.* PhD thesis, Massachusetts Institute of Technology, 2005.

[60] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, 2000.

[61] M. Rabinowitz and J.J. Spilker Jr. A New Positioning System Using Television Synchronization Signals. *Broadcasting, IEEE Transactions on*, 51(1):51–61, 2005.

[62] V. Ramadurai and M. L. Sichitiu. Localization in wireless sensor networks: A probabilistic approach. In *Proceedings of 2003 International Conference on Wireless Networks*, pages 300–305, Jun 2003.

[63] T. Rappaport, J. Reed, and B. Woerner. Position location using wireless communications on highways of the future. *IEEE Communications Magazine*, 34(10):33–41, 1996.

[64] K. Rehrl, N. Göll, S. Leitinger, and S. Bruntsch. Combined indoor/outdoor Smartphone navigation for public transport travellers. *Location Based Services & Telecartography-Proceedings of the Symposium*, 2005.

[65] K. Römer. The Lighthouse Location System for Smart Dust. *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 15–30, 2003.

[66] R. Roy and T. Kailath. Esprit-estimation of signal parameters via rotational invariance techniques. *Signal processing part II: control theory and applications*, pages 369–411, 1990.

[67] R. Roy, A. Paulraj, and T. Kailath. A subspace rotation approach to signal parameter estimation. In *Proceedings of the IEEE*, volume 74, pages 1044–1046, 1986.

[68] C. Savarese, J. M. Rabaey, and K. Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *ATEC '02: Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*, pages 317–327, Berkeley, CA, USA, 2002. USENIX Association.

[69] A. Savvides, C. C. Han, and M. B. Strivastava. Dynamic fine-grained localization in Ad-Hoc networks of sensors. *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, 2001.

[70] A. Savvides, H. Park, and M.B. Srivastava. The Bits and Flops of the N-Hop Multilateration Primitive for Node Localization Problems. *Proceedings of the First International Workshop on Wireless Networks and Applications held in conjunction with Mobicom 2002*, 2002.

[71] A. Savvides, M. Srivastava, L. Girod, and D. Estrin. Localization in Sensor Networks. *Journal of Wireless Sensor Networks*, pages 327–349, 2004.

[72] S. V. Schell and W. A. Gardner. High resolution direction finding. *Handbook of statistics*, 10:755–817, 1993.

[73] R. Schmidt. Multiple emitter location and signal parameter estimation. *Antennas and Propagation IEEE Transactions*, 34(3):276–280, 1986.

[74] S.A.N. Shafer. Ubiquitous Computing and the EasyLiving Project. *40th Anniversary Symposium, Osaka Electro-Communications University. ht tp://www. research. microsoft. com/easyliving*, 2001.

[75] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 201–212, 2003.

[76] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 1–12, New York, NY, USA, 2004. ACM Press.

[77] P. Steggles and S. Gschwind. The Ubisense Smart Space Platform. *Adjunct Proceedings of the Third International Conference on Pervasive Computing*, 2005.

[78] Blip Systems. Homepage of the blip systems. `http://www.blipsystems.com/`, Agust 2008.

[79] R. Want, A. Hopper, V. Falc ao, and J. Gibbons. The active badge location system. In *ACM Transactions on Information Systems*, pages 91–102. ACM Press New York, NY, USA, 1992.

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

[81] K. Whitehouse. Homepage of the calamary project. `http://www.cs.virginia.edu/~whitehouse/research/localization/`, August 2008.

[82] S. Wireless. Home page of the skyhookwireless project. `http://www.skyhookwireless.com/index.html`, September 2007.

[83] S. Wireless. Home page of the wherenet project. http://http://www.wherenet.com/, January 2008.

[84] S. Y. Wong, J. G. Lim, S.V. Rao, and W.K.G. Seah. Density-aware hop-count localization in wireless sensor networks with variable density. In *IEEE Wireless Communications and Networking Conference (WCNC '05), USA*, 2005.

[85] S. Y. Wong, J. G. Lim, S.V. Rao, and W.K.G. Seah. Multihop localization with density and path length awareness in non-uniform wireless sensor networks. In *IEEE Vehicular Technology Conference, Sweden*, 2005.

[86] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan. Ecolocation: a sequence based technique for rf localization in wireless sensor networks. In *IPSN '05: Proceedings of the 4th international symposium on Information processing in sensor networks*, page 38, Piscataway, NJ, USA, 2005. IEEE Press.

[87] A. Youssef, J. Krumm, E. Miller, G. Cermak, and E. Horvitz. Computing location from ambient FM radio signals [commercial radio station signals]. *Wireless Communications and Networking Conference, 2005 IEEE*, 2, 2005.

[88] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 1–13, New York, NY, USA, 2003. ACM.