Institute of Parallel and Distributed Systems
University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Studienarbeit Nr. 3137

Design and development of a localization system for a sensor network in collective symbiotic organisms

Jeimy Catherine Millán Ochoa

Course of Study:	Information Technology MSc
Examiner:	Prof. Dr. rer. nat. habil. Paul Levi
Supervisor:	Dipl.-Ing. Eugen Meister Sergej Popesku
Commenced:	21.01.2011
Completed:	21.07.2011
CR-Classification:	I.2.9 C.3 C.2.1 C.2.4

Acknowledgement

This master thesis would not have been possible without the guidance, the support and the help of several people who in one way or another contributed and extended their valuable assistance in the preparation and completion of this project.

Thus, I would like to thank my family and friends for their support and understanding through the duration of my studies; most especially to Jonas, Jorge and Matthias for their unconditional help and for encouraging me to pursue this degree.

I am utmost grateful to Prof. Dr. rer. nat. habil. Paul Levi, who gives me the opportunity to work on this very interesting thesis at IPVS institute. And I am heartily thankful to my supervisors, Dipl.-Ing. Eugen Meister and Sergej Popesku, for offering me their invaluable assistance, support and guidance.

Abstract

The REPLICATOR project aims for developing self-adaptive and self-assembling organisms compounded of stand-alone robots. These robots can autonomously and dynamically dock with each other forming symbiotic structures. This system requires that each robot knows the position of the others. Therefore, this master thesis consists of the design, implementation and assessment of a low power and low cost localization system, based on the strength of the signal of the wireless communications protocol ZigBee. Furthermore, a dynamic ZigBee coordinator selection process was developed to separate the localization application and the system's configuration.

The proposed solution is a distributed, anchor-free, self-configuring, cooperative and concurrent algorithm. The localization is iteratively recalculated and updated using a Eucledian method. In order to reduce the errors produced by the fluctuations of the signal, a weighted approach, IIR and FIR filters; and a calibration process were implemented.

It was found that the implemented localization system can achieve localization errors lower than five percent. However, these errors vary greatly depending upon several uncontrolled facts such as the environment in which the system is deployed and the radiation pattern of the antennas; and in addition, the errors increase exponentially with the distance.

Key Words: Localization system, ZigBee, RSSI, iterative,cooperative anchor-free,self-configuring

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

This master thesis belongs to the REPLICATOR project, which aims for developing a collective symbiotic system compounded of stand-alone robots. These robots can autonomously and dynamically dock with each other, forming a new structure of connected robots; share energy, information and resources among the peer in a structure, and interact as a unit with the physical world.

In order to achieve their self-assembling capability, it is required that each robot knows the position of the others. Therefore, the main goal of this thesis is the design and development a localization system constrained by the limitations and requirements of this symbiotic system of robots.

The next section is devoted to a brief description of the REPLICATOR project, which attempts to show a better understanding of the aforementioned requirements and limitations. The following sections introduce the problems faced in the design and implementation of a localization system, and the developed solution.

1.2 REPLICATOR project overview

The REPLICATOR project has as a target the development of evolutionary self-adaptive and self-assembling multi-robot organisms. These robots are autonomous, reliable, highly capable and sensor-rich, and they are able to self-assemble into large artificial organisms on their own initiative.

Self- adaptation in these robots and organisms, is defined as its autonomous assembling without human interaction [rep11]. It pursues the ability to better organize the collective robotic system to allow its performance under hostile or unexpected circumstances, where the number of robots and topologies of the organism cannot be pre-programmed [LK10].

The stand-alone robots, which are the atomic components of the collective symbiotic organisms, should have, in terms of mechanic, electronic and software imple-

mentation, a balanced complexity. A high complexity on their design may result in high power consumptions and expensive manufacturing costs.

On the other hand, the robots should be able to be reconfigurable, adaptable, reliable and self-assembling. They should also exhibit other properties such as locomotion, docking capabilities, communications with other robots or organisms, and interface with several sensors i.e. accelerometers, infrared detectors (for ranging calculation), microphones or cameras (for tasks like orientation). For these reasons, the design of these robots is not simple and it requires that each part of the robots will be as simple as possible.

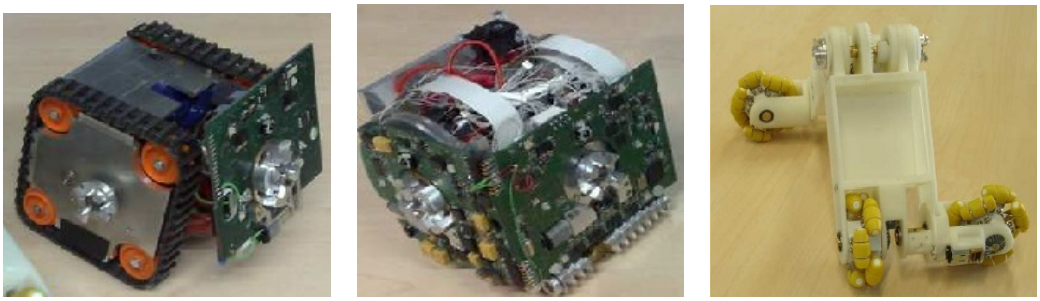

Figure 1.1: Types of robots

Three types of robots, shown in Figure 1.1[1], compose the swarm used with the localization system developed in this thesis. They are:

- *Scouts*: They have long-range sensors and caterpillar tracks, see Figure 1.1 (left). They can move fast and efficiently on rough terrains, characteristic that may be used for environment inspection. They are able to easily swift when they are docking. This type of robots may be useful for locomotion when they are docked in an extreme of the swarm[2] [LK10].

- *Backbone*, Figure 1.1 (center). They are firm and hard robots, which are designed to give stability to the organism. They can lift several docked robots [LK10].

- *Active wheel*, Figure 1.1 (right). They can move in all directions, lift and carry heavy robots, and be energy source [LK10].

[1] Images taken from [LK10]
[2] Extreme point can be considered as the end of a leg or arm in a living organism

Figure 1.2[3] shows an organism formed by a Scout robot, Backbone robot and an Active wheel.

Figure 1.2: Docked organism

1.3 Localization problem

As it was already described, the robots should self-assemble and form collective symbiotic organisms. The first step in this self-assembling process is the localization[4] of all robots that are part of the system. This localization should give the robots the information about the number of robots in the system, their identification, and their position.

Once a robot decides to assemble another robot or organism, it should first identify the best type of robot to join, its internal state and location. Then, it can approach to the selected robot, using its localization system, and dock with it.

The localization problem has the following features and limitations:

- it is limited to 2 dimensions
- no device knows its position, which means there are no anchors[5]
- the number of robots, or organisms, in the system is not fixed, it can be a large network or it can be minimum three[6]

[3] Image taken from [LK10]
[4] i.e. to compute their positions in some fixed coordinate system [Pal10]
[5] from here anchors will be used as device that knows its exact location in a determined area
[6] In this document, robots or organisms are considered as nodes or devices, and these words will be used to refer them

- all nodes in the system should be covered and located
- the devices may be mobile
- the solution should be low cost, low power consumption, no complex, robust and adaptable
- the localization system should work even if a random node of the network is disconnected
- the topology of the system is random, i.e. there are not fixed devices which are always part of the system.

According to these requirements and limitations of the problem, it was designed, implemented and assessed a low power and low cost, localization system. It uses as the main resource for the calculation of the localization the strength of the signal communications system. Specifically, ZigBee was chosen as the communications platform because it already provides a measurement of the strength of the communications signal, and in addition, because it is a wireless communication system that can be used among the devices which compound the collective symbiotic system.

ZigBee features a robust, self-healing, ad hoc[7] and dynamic communications protocol, which allows connecting several nodes with low power consumption and communication cost. In addition, the ZigBee standard offers interoperability among other systems that use this protocol, opening future opportunities to extend the system with minimum changes (see Section 2.2).

The implemented localization system is a merge of different solutions described in this report in Section 2.1. It is implemented as a distributed, anchor-free, self-configuring and concurrent algorithm, whose nodes cooperate with each other to estimate their location based on the Received Signal Strength Indicator (RSSI).

The distance among the nodes is an exponential function of the RSSI. This indicator is extremely dependent on several factors, such as the characteristics of environment in which the system is deployed, the properties of the radio transmitters, losses for connections, and others. As result, the strength of the signal has uncontrolled variations that reduces the precision of the localization system.

[7] In wireless as ad hoc networks it is not possible to place the nodes deterministically, the network does not have a preexisting infrastructure, and each node participates in the routing by dynamically forwarding data [NN01] [YYK02]

Therefore, functions, which aim to reduce the errors produced by the environment and behavior of the signal were implemented. These functions weight and filter the obtained RSSI values, and perform calibration process.

To obtain a function that described the distance between two robots based on the strength of their RRSI, the radios that were used for the system's implementation were characterized, for ranges up to 10m. Since ZigBee messages are continuously being sent between the nodes in the system, and these messages contain the RSSI, the localization of the nodes can be continuously calculated. In the implementation described in this document the localization of the nodes is iteratively recalculated and updated for a minimum of three devices, using an Eucledian method described in Section 2.1.2. In this way, the distances and positions are iteratively recalculated and updated.

In a ZigBee network there are different kinds of devices, a coordinator, routers and end devices. The coordinator establishes the network, thus, normally it is a pre-configured device, always present in the network. However, as one of the characteristics of the Replicator system is that the presence of its nodes is not fixed, it is not feasible to have a pre-configured coordinator. To overcome that problem, a dynamic coordinator selection, always executed at the system's initialization, was developed.

The selection of the coordinator is a process based on the priority assigned to each device, and on their sequence of discovery. Any device can become a coordinator or a router, and as a result, there are no pre-programmed devices, thus the localization application is independent of the system's configuration. The device that wins this process of selection becomes the coordinator and establishes a ZigBee network, while the other devices become routers and join this network.

It was found that using the proposed system, the positioning of the all devices in the network can be determined with an error sometimes lower than five percent, and that this error depends on the device's positions and the cumulative error obtained during the calculation of every device's position. It was also found that the magnitude of this error drastically changes with the antenna's position or due to characteristics of the environment that affect the communications signal. When distances are higher than two meters, the error can be exceed 100%. Therefore the proposed localization system can give a good approximation of the nodes location, but requires the use of other sensors or a fixed configuration in order to improve the accuracy of the localization and to gain independence of these aforementioned fluctuations.

1.4 Structure of this thesis

This document is divided into four main parts:

- Chapter 2 introduces the theoretical foundations on the localization theory, providing a brief description of different localization methods and approaches proposed in this research area. It is followed by a review of the basic theory of ZigBee, which contains the main definitions and procedures used to development of this thesis. Finally, the basic theory of the digital filters which were employed in the implementation of this system is discussed.
- Chapter 3 presents the implementation of the localization system. It includes the description of its hardware, the path loss effects and software platforms, the main phases of the localization application, RSSI data processing, calibration, distance estimation and positioning calculation. Lastly, there is a description of the ZigBee application, which consists of a dynamic coordinator selection, the RSSI propagation algorithm.
- The experimental results are shown in Chapter 4. They are divided into 2 main parts. The first part contains the assessment of the localization application. It consists of the analysis of a set of 6 proposed cases. Finally, the coordinator selection evaluation is expounded. This part covers 3 sets of experiments.
- Chapter 5 contains the summary and analysis of the results, and proposes improvements that future, related works.

2 Theory

This report documents the design and implementation of a localization system used by the REPLICATOR swarm of robots. This system uses ZigBee (2.4GHz) not only to implement its communications system, but also as a resource to obtain signals used in the computation of the distance among the robots that compose it. This chapter consists of two parts; Section 2.1 where the foundations on the localization theory are summarized, and Section 2.2 which contains ZigBee's main definitions and the procedures utilized to development this thesis.

2.1 Localization

Physical location is an important property in cooperative and autonomous organisms. It is related to Wireless Sensors Networks, a research area that has been extensively investigated. When a localization system is to be conceived, it is initially described by the definition shown in Section 2.1.1. Its definition is later constrained by its requirements, the system's hardware limitations or the challenges posed by its implementation. These constrains have led researchers to propose some implementation methods and alternatives described in Section 2.1.2.

The classification of the localization algorithms are presented in Section 2.1.3, and Section 2.1.4 is devoted to the description of four representative algorithms implementations: Assumption Based Coordinates, Hop-TERRAIN, Bounding Box and Error Propagation Aware.

2.1.1 Problem definition

Lets consider a network of N nodes:

$$L = [l_1, l_2, l_3...., l_N] \text{ [Pal10]}$$

Where every l_i consists of l_{xi} for the node's position $X-$coordinate and l_{yi} for the node's $Y-$coordinate.

Determining these locations constitutes a localization problem, and in this study, focused to the context of a network where its nodes do not have anchors, and each

one finds the position of other nodes using their ZigBee's RSSI (Receiver's Signal Strength Indicator)(RSSI)[1] of each one.

2.1.2 Foundations

Localization in wireless networks is an extensive area of research which can be divided in 3 main portions:

1. Determining distance or angle estimation [Pal10]
2. Localization
3. Refinement

Determining distance or angle estimation

For the distance estimation, the most used methods are:

- RSSI: With the use of theoretical or empirical models, the strength of a communications signal can be translated into the distance between the transmitting node and the receiver. Although this is a low cost and low power consumption solution, it has a low accuracy due to the deleterious of the signal caused for fading, reflections, multi-path and other effects [Sug06] [Pal10]. These effects are described in the *Path loss theory* Section of this chapter 2.2.5. This method is used in [Sug06] [AV06]
- Angle of arrival (AOA).By using the received signal, this method estimates the angle of transmitter that generated it, and using simple geometric calculations the position of this transmitter is obtained. AOA is more accurate than RSSI but it requires expensive antenna arrays on each node [Pal10] [SRB01]. This approach is implemented in [NN03]
- Time based methods. Time of arrival (TOA) or time-distance of arrival (TDOA) of a signal is a method where the distance between a receiver and a transmitter is calculated based on the propagation time of the signal and the signal's propagation speed. The signals used in this method can be RF, acoustic, infrared or ultrasound. This method is highly accurate when the nodes are in the line of sight; otherwise the estimation is deteriorated due to multi-path effects [Pal10]. This method requires a system capable of high speed calculations and thus, expensive hardware. TOA is implemented in the global positioning system (GPS) [Log92] or in less cost applications as is showed in [JZS05]

[1] RSSI, is an indication of the power level being received by the antenna. The higher the RSSI level (less negative number in some devices), the stronger the signal. This numeric value is an integer with an allowable range of $0-255$[80297]

- Interferometric. In this approach is required that two transmitters located at two different positions, emit simultaneously signals with slightly different frequencies. When combined, they result in a signal with interference patterns and a phase that is a function of the receiver's location. Two different nodes can receive this signal, and calculate their difference of phase. From this difference, the spatial coordinates of the two transmitters and two receivers can be computed [PH06]

Localization methods

Some of the most popular methods used for the localization of a node in a collective are:

- Triangulation. This method uses the direction of the node with respect to an absolute point or to another node, instead of the distance. The localization is estimated by using the trigonometry laws of sines and cosines [Pal10]. This method is used in [NN01] and [CSL02]

- Eucledian - tri or multi-lateration. The localization is calculated using the intersection of at least 3 circles 2.1. The advantage of this approach is that no global resources or communications are needed. However, when using this method, the convergence of a computed solution may require longer time than using other methods, it is not well suited for systems with collinear nodes, and its complexity increases when a system has nodes with high mobility. This method is used in [PBDT05] [SRB01]

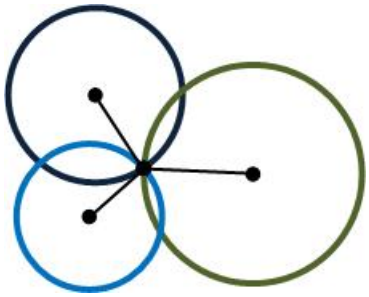

Figure 2.1: Eucledian−trilateration

- Maximum Likelihood estimation. This method calculates the position of a node by minimizing the differences between the measured and estimated distances [Pal10] [JBS03]. This approach is used in [JBS03]

- Hybrid methods combine 2 or more localization methods like ***DV−distance***, ***DV-hop*** [NN01]. This approach is anchor-based and its implementation consists of the calculation of the distances' sum between nearest nodes (DV-distance) or sum of hops (DV-hop), for the nodes localization. In the case of ***DV-hop***, each anchor obtains the distances to other anchors and calculates the *length's average* for one hop. That is, anchors calculate the distance, using time based methods as an example, to other anchors and then divide the sum of resulting distances by the sum of minimum hops among them. For instance, in Figure 2.2 the length's average for: A_1 is $\frac{10+20}{2+3} = 6$; $A_2 = \frac{10+30}{2+5} = 5.7$ and $A_3 = \frac{30+20}{5+3} = 6.25$.

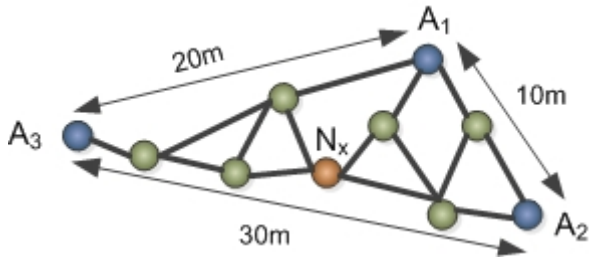

Figure 2.2: DV-Hop example

These *length's averages*, named also *corrections*, are transmitted to the nodes. Then each node can estimate its distance to the anchors, based on the correction of its nearest anchor and on the minimum number of hops between the node and anchors. For example node N_x has as nearest hop A_1 with a correction of 6; The estimated distances are this correction multiplied by the minimum number of hops: that is between N_x and A_1 $6x2 = 12$; N_x and A_2 $6x2 = 12$ and N_x - A_3 $6x3 = 18$. Finally using triangulation N_x calculates its position.

In ***DV−distance*** case, it is used the strength of the signal between hops, instead of the number of hops, for the calculations. The advantages of this approach are, its simplicity and the fact that it does not depend on error measurement propagation.

Refinement

There are four main refinements schemes to consider:

1. Iterative multi-lateration. Nodes iteratively recompute their positions using the most recently computed coordinates and range measurements of their neighbors. This method is used in [SRB01]

2. Heuristic. Proposed in [CSL02], a vector of weights, representing the confidence on the obtained solutions, is iteratively used to reduce the error produced for unreliable information.

3. The Weighted Least Square (WLS) based methods. WLS, used in [APA06], is designed to minimize the mean square error in the calculated positions of the nodes. One approach, implemented in [CDGS04], attempts to minimize the relative position error between nodes, instead of reduce the error in the node's absolute positions. In their approach, called *Iterative Least Mean Squared Refinement* (LMSR), it is assumed that ranging errors between node i and j, are Gaussian with known standard deviations σ_{ij} equal to the range measured between the nodes r_{ij}. The measurements are weighted with their observed standard deviation σ_{ij}, obtained by a previous calibration phase. Then the position is calculated and continuously exchanged and updated among the current estimated location of other nodes until they converge.

4. Kalman filter based approach. The Kalman filter is a recursive estimator. A two phases approximation of this filter is implemented in [ASS02]: The first phase predicts the position in a point of time, based on a model of the system's behavior; The second phase consists of the estimation of the current position (based on measurements) and the prediction's refinement. The final position is a combination between the results of these two phases.

2.1.3 Classification of the Localization algorithms

The algorithms used in localization can be classified by the topologies used by the nodes responsible for performing the data computation:

- Centralized. The data received from every node is processed in a central base station and estimates the localization of each node. This information is later transmitted back to the nodes. The advantage of this topology is that the computation is performed by the base station, reducing the amount of communication traffic and of computation time [AV06]

- Distributed. The computation of the localization is performed in every node and transmitted to the others. These distributed algorithms can be classified according to the use of anchors:
  - ◇ Anchor based. If the algorithm relies on nodes with prior knowledge of their locations [NN01] [ASS01] [SRB01] [SS02]

◇ Anchor free. It is implemented for instance in [NBPT03] and in ABC algorithm [SRB01]. The later is described in Section 2.1.4.

If the localization is performed in few nodes and then propagated to the others, or if every node estimates its localization:

◇ Incremental. In this approach, the algorithms usually start the calculation of the position between 3 or 4 anchors, then later other nodes are added and their position is estimated based on trigonometric formulations, using the positions already estimated and distances or angles between all the devices. The disadvantage of these algorithms is that the errors obtained in the measurement of the distances or the angles between the nodes are propagated. A phase of refinement can be implemented to reduce the error. [SRB01][NN03]

◇ Concurrent. In this approach, all nodes simultaneously estimate and refine their coordinates. In some cases to attempt to avoid the propagation of distance or angle measurement errors, an iterative optimization is used to reduce the error between the measured and calculated values. [NBPT03] [SRB01] [CSL02]

Other classifications are:

- Relaxation-based [Pal10], in this case, a roughly localization followed by a refinement phase are performed [NBPT03][SRB01]
- Overlapping and merging. In this method, the network is divided into small overlapping sub regions, which creates a local map; then the local maps are combined into a global map [DMT04] [MF04]
- Hybrid, where the algorithm uses two or more different localization techniques, i.e. [ASS05].

2.1.4 Related work

Assumption Based Coordinates (ABC) algorithm

This algorithm, shown by [SRB01], sequentially estimates the location of each node. The nodes start the localization, making assumptions about the other node's position when necessary, and through correction and redundant calculations using new available information to compensate the errors.

Initially, the algorithm assumes that a node n_0 is in the position $(0, 0, 0)$, as shown in Figure 2.3. The first node n_1 establishes communication with n_0 and its position is assumed at $(r_{01}, 0, 0)$, where r_{01} is the distance between n_0 and n_1 extracted from the RSSI. The next node n_2 is located at $(x_2, y_2, 0)$, see equations (2.1); where y_2 is positive.

$$\begin{aligned} x_2 &= \frac{r_{01}^2 + r_{02}^2 - r_{12}^2}{2r_{01}} \\ y_2 &= \sqrt{r_{02}^2 - x_2^2} \end{aligned} \tag{2.1}$$

The next node n_3 is located at the coordinates (x_3, y_3, z_3), where z_3 is positive:

$$\begin{aligned} x_3 &= \frac{r_{01}^2 + r_{03}^2 - r_{13}^2}{2r_{01}} \\ y_3 &= \frac{r_{03}^2 - r_{23}^2 + x_2^2 + y_2^2 - 2x_2y_3^2}{2y_2} \\ z_3 &= \sqrt{r_{03}^2 - x_3^2 - y_3^2} \end{aligned} \tag{2.2}$$

For a new node m, the algorithm can be applied again with the equations 2.2:

$$\begin{aligned} x_m &= \frac{r_{01}^2 + r_{0m}^2 - r_{1m}^2}{2r_{01}} \\ y_m &= \frac{r_{0m}^2 - r_{2m}^2 + x_2^2 + y_2^2 - 2x_2y_m^2}{2y_2} \\ z_m &= \sqrt{r_{0m}^2 - x_m^2 - y_m^2} \end{aligned} \tag{2.3}$$

The next phase is to perform a cooperative ranging approach used to iteratively determine the global position using the interaction of every node with each other. All nodes repetitively and concurrently perform the following functions:

- Receive ranging and location information from neighboring nodes
- Execute ABC algorithm for the localization
- Transmit the obtained results to the neighboring nodes.

After the algorithm finds a converged solution, new calculations are rarely needed.

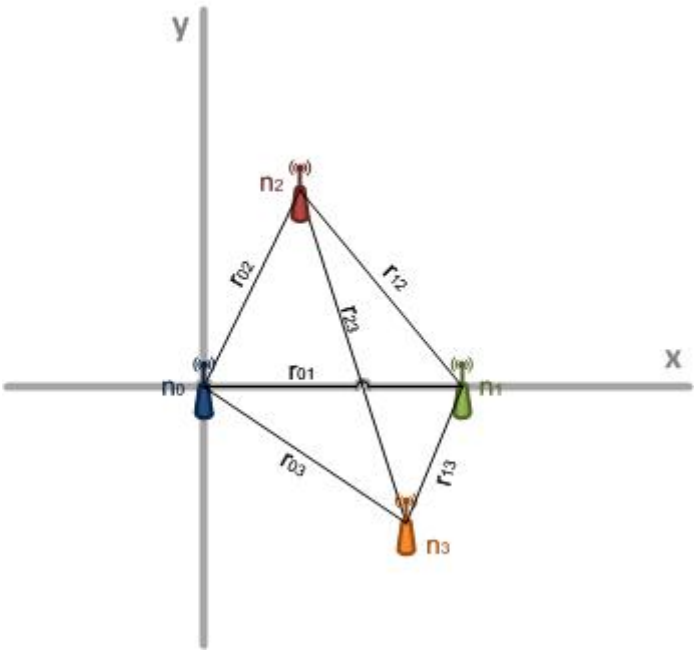

Figure 2.3: Assumption Based Coordinates

The advantage of this approach is that no global communications or resources are needed, although the time for convergence may be long. Also, one of its disadvantages is that nodes with a high mobility may be hard to cover.

Hop-TERRAIN and Refinement with confidence weights

The authors describe in [CSL02] an algorithm with two phases, the first phase *Hop-Terrain*, is a variant of Terrain, an anchor-based algorithm with foundations in ABC [SRB01] and similar to DV-hop [NN01]. This phase runs once at the beginning of the localization. Next the refinement (second phase) is performed iteratively.

Hop-TERRAIN algorithm finds the number of hops from a node to each of the anchors and then multiplies this hop count by a shared metric (average hop distance) to estimate the distance between the node and the anchors. Each anchor performs a triangulation, which consists of solving a system of linear equations $Ax = b$, gets the node's estimated position and broadcast this information. Once a node has received data from at least 3 (4) anchors[2], it estimates its location with

[2] 3 for 2-Dimensions and 4 for 3-D

triangulation. Next, each node executes a concurrent and iterative optimization (refinement), using the distances to the anchors and their coordinates.

The second phase, refinement, is an algorithm in which the nodes update their positions, first broadcasting its calculated position, receiving the positions and corresponding estimated distance from its neighbors, and then computing its new position through a least squares triangulation. When the update position has small changes, the refinement phase stops and reports the final position.

The Refinement algorithm includes a *confidence* associated with each node's position. These confidences are used to weigh the system of linear equations. Instead of solving $Ax = b$, it is solved $wAx = bw$, where w is the vector of confidence weights. Nodes with reliable position, such as anchors, will have a high confidence value I.e. close to 1. And node with poor conditions will have low confidence, close to 0. With this confidence the impact of poor connections or fallible information is lower when triangulation is performed by the nodes.

Bounding Box [ASS02]

This algorithm is based on collaborative multi-lateration, where nodes estimate accurately their locations by using the locations of the anchors, which can be several hops away, and by measuring the distance to direct neighboring nodes. Then the nodes, through ranging sensors, measure distances to their neighbors, share their measurement and location information with their neighbors and collectively estimate their locations.

One method for establishing a coordinate system is to deploy some anchors that are capable of accurate long distance ranging (i.e. long range ultrasound or laser range finders). These anchors establish a local coordinate system. The anchors can communicate with each other and decide on a local coordinate system. As an example, if the decide to use triangulation, they create a local coordinate systems at each anchor and then merge them together into a global coordinate system.

Next the nodes can estimate their location. To do that that,it is required that there are at least 3 neighbors which location has unique solutions, i.e is 3 anchors (*One-Hop multi-lateration*), or 2 neighboring nodes, each one surrounding of at least by 2 non-collinear anchors (*Two-Hop multi-lateration*). To determine if nodes located within n hops from the anchors have unique solutions, the node with unknown position tests if it has at least three neighbors with tentatively unique locations. If the three neighbors have unknown positions as well, a recursive call is executed

at each neighbor to determine if its location is unique. Each node used as an independent reference is marked in order to prevent other nodes to re-use it as an independent reference. This iterative method is called *N-Hop multi-lateration*.

After using N-Hop multi-lateration, to obtain the localization of every node, a bounding box for each node is constructed using their reference anchors' position and estimated distance; then the intersection of these boxes or intersection of the circles is determined, see Figure 2.4. The position of the node is set to the center of the intersection box.

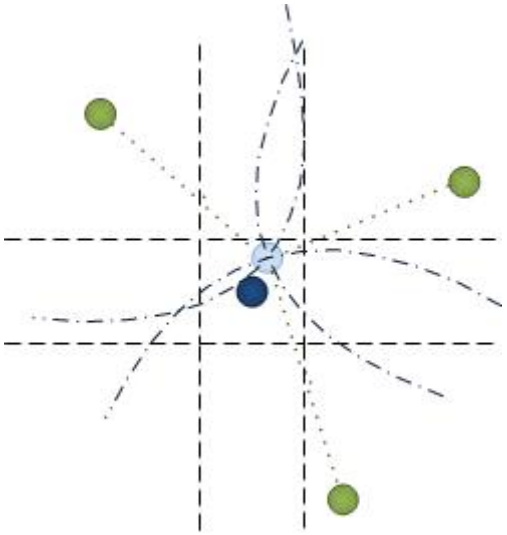

Figure 2.4: Bounding box

The bounding box of a node n_1 is created by adding and subtracting the estimated distance d to the anchor position:

$$\begin{aligned} &(x_1 + d) \;\; (x_1 - d) \\ &(y_1 + d) \;\; (y_1 - d) \end{aligned} \tag{2.4}$$

The intersection of the bounding boxes is computed with the maximum of all minimum coordinates and the minimum of all maximums:

$$\begin{aligned} &min(x_1 + d), \;\; min(y_1 + d) \\ &max(x_1 - d), \;\; max(y_1 - d) \end{aligned} \tag{2.5}$$

The final position is the average of both corner coordinates. Finally this algorithm has an iterative refinement phase using a Kalman Filter.

Error Propagation Aware (EPA) algorithm

An algorithm which integrates the path loss and distance measurement error model is proposed in [APA06]. When a transmitting node is physically close to the receiving node, the power of the received signal is very strong. When one node moves away from the other, the strength of the signal decays gradually and then it reaches a region where the signal is weak; any calculation of the distance can be acceptable with occasionally, large errors. Finally, when a node can no longer be detected by the other because the signal falls below a certain threshold, neither positioning nor communication can take place.

This means that the positioning of the nodes experience different kinds of errors according to the detected strength of the received signal, and since these errors are associated to the characteristics of the communications channel, its behavior cannot be ignored. The authors base their algorithm on path loss and distance measurement errors models, developed before the algorithm application.

At the beginning of the algorithm, the anchors broadcast their IDs, global coordinates, and the calculated error variance of its position. Each node records the TOA[3] information of each anchor and their transmitted data. Then the nodes estimate the variance of the strength of the received signals based on their error models. These estimated variances and the received from the anchors are employed to build a weighting matrix.

Next, the node computes its position using multi-lateration techniques, which attempt, in general, to minimize the objective function given by:

$$f(x) = \sum_{i}^{N} \left(\sqrt{(x - x_i)^2 + (y - y_i)^2} - \widehat{d}_i \right)^2 \tag{2.6}$$

where given n anchors, (x_i, y_i) are the coordinates of the i^{th} anchor, $i \in [1, n]$ and (x, y) is the coordinate of the node. $\widehat{d}_i = d_i + \epsilon_i$ is the measured range between the anchors and the node, d_i is the physical distance between the node and the anchor i and ϵ_i is a zero-mean Gaussian random variable with a variance, resulting from the error models. The minimization of (2.6) is performed through a Weighted Least Square algorithm [Kay93] with the weighting matrix.

After getting its own position the node becomes an anchor and starts broadcasting its ID, global coordinate and variance. This process is repeated until all the nodes obtain their position and are transformed into anchors.

[3]Time Of Arrival

2.2 ZigBee

ZigBee is a low-cost, low-power consumption, wireless protocol based on IEEE 802.15.4 2003 standard [XP09]. The main features of ZigBee include self-healing system of redundant, self-configuring, energy-efficient mesh networking and support of networks with hundreds of devices [all11]. The principal concepts involved in the understanding of this project described in this document are summarized in this section. These concepts are the device types, Section 2.2.1 and PAN ID 2.2.2.

The main features of ZigBee Stack are expounded in 2.2.3 and the procedure of establishing a network in 2.2.4. Lastly, Section 2.2.5 presents the basis of a path loss model and the effect of the losses in radio frequency transmissions.

2.2.1 ZigBee device types

A ZigBee network consists of a number of ZigBee devices or Nodes. A node is a piece of hardware that shares a single radio. Nodes can have several subunits, which are physical devices such as sensors. In the case used in this project, every robot is a node.

ZigBee defines two kinds of functional devices in the network [zig08a] [XP09]:

- Full Function Device (FFD). It offers most or all the services, it is always powered and is always on, even when it is in idle state.
- Reduced Function Device (RFD). The services offered by these devices are limited. It can be battery-powered and in Idle stays can be off.

There are three different device types in ZigBee [zig08a] [XP09]

1. **Coordinator:** It is a FFD. There is only one in the network and it sets up a channel, a Personal Area Network (PAN) ID of 16-bits and the extended PAN-ID of 64-bits to start the network. It can allow routers and end devices to join the network and assist in routing data. The Coordinator cannot sleep, should be always powered on and it may buffer RF data packets for sleeping end devices or devices that it has allowed to join the network.
2. **Router:** It should be a FFD and it must join a ZigBee PAN before it can transmit, receive, or route data. After joining, it can allow routers and end devices to join the network and it may assist in routing data. Routers cannot sleep (should be mains powered). They can buffer RF data packets for sleeping end devices. Once a router has joined a network, it retains the following information through power cycle or reset events:

- PAN ID
- Operating channel
- Security policy and frame counter values
- Child table (end device children that are joined to the coordinator)

The router will retain this information indefinitely until it leaves the network. When the router leaves a network, its PAN ID, operating channel, and child table data are lost.

3. **End device:** It is a FFD or RFD. It must join a ZigBee PAN before it can transmit or receive data. It cannot allow devices to join the network, must always transmit and receive RF data through its parent (device which allowed it to join the network) and cannot route data. End devices can enter low power modes to conserve power and can be battery-powered.

2.2.2 PAN ID

ZigBee networks are called Personal Area Networks or PANs [zig08b]. Each network is defined with a unique PAN identifier (PAN ID). This identifier is common among all devices of the same network. ZigBee devices are either pre-configured with a PAN ID to join, or they can discover nearby networks and select a PAN ID to join.

The PAN ID of 16-bit is used as a MAC layer addressing field in all RF data transmissions between devices in a network. But there is a possibility that multiple ZigBee networks (within range of each other) could use the same 16-bit PAN ID. To resolve potential 16-bit PAN ID conflicts, the ZigBee Alliance [all11] created a 64-bit extended PAN ID.

ZigBee supports both a 64-bit and a 16-bit PAN ID. When a coordinator starts a network, it can either start a network on a pre-configured 64-bit PAN ID at manufacturing or set by the user, or it can select a random one. ZigBee routers and end devices should be configured with the 64-bit PAN ID of the network they want to join. They typically acquire the 16-bit PAN ID when they join a network.

2.2.3 ZigBee Stack

The software implementation of a protocol is known as protocol stack software. The ZigBee stack architecture, shown in Figure 2.5, is based on the OSI model. The IEEE 802.15.4-2003 [80203] standard defines the physical (PHY) layer and the medium access control (MAC) sub-layer of the ZigBee stack. It was developed

independently of the ZigBee Standard. The MAC layer validates the frames, maintains the synchronization of the network, controls the association or disassociation and schedules the time slots. It is based on the carrier sense multiple-access/collision avoidance mechanism (CSMA/CA) [XP09].

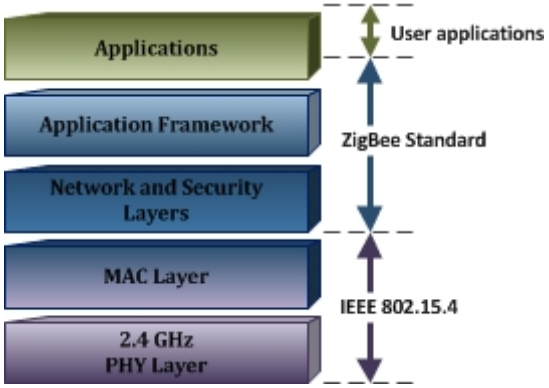

Figure 2.5: ZigBee architecture

The ZigBee Alliance[4] provides the network layer and the framework for the application layer, giving procedures and mechanisms for security, delivery mode, binding[5], messages' fragmentation etc.

Frequency

The IEEE 802.15.4-2003 standard defines the following frequencies [zig08b] [XP09]:

- 868.0-868.6 MHz for Europe. It has one communication channel
- 902-928 MHz for North America with up to thirty channels
- 2400-2483.5 MHz is worldwide use and has up to sixteen channels

The center frequency (MHz) of the channels for the band of 2.4 GHz is defined:

$$Fc = 2405 + 5(k - 11) \tag{2.7}$$

[4]The ZigBee Alliance is an association of companies working together to enable reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard [all11]

[5]Binding is a mechanism for the control flow of messages form one application to another application

k is the channel number and it can be a value between 11 and 16.

Network Topologies

The ZigBee network layer (NWK) supports star, tree, and mesh topologies (Figure 2.6).

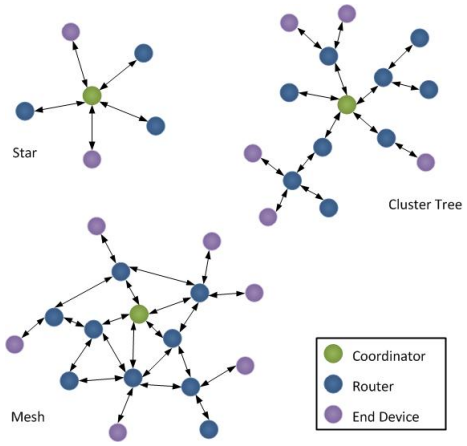

Figure 2.6: ZigBee topologies

The star topology is the simplest and most limited of the topologies. In this topology, each node is connected directly to the coordinator and therefore, all messages pass through it.

The cluster tree consists of a number of star networks connected whose central nodes (routers) are also in direct communications with the coordinator. An end device can only directly communicate with its parent (router or coordinator); routers can only directly communicate with its end devices and its own parent. The messages travel up and then down the tree, until they reach the destination node.

In a mesh topology the coordinator and every router are connected to other nodes on the network, therefore, the data packets can traverse multiple nodes (hops) from their source to the destination using a process called route discovery. With this type of networking, the connections are dynamic in terms of reconfiguration around broken or blocked paths by "hopping" from node to node and the routing of messages is decentralized and cooperative process and independent of the coordinator.

Delivery mode

In ZigBee application layer, there are different modes to send a frame [zig08b]:

- Unicast: The frame will be delivered to a specific device
- Broadcast: The message is delivered to all devices within the network
- Multicast: The frame will only be delivered to the devices which are part of a specific group

The implementation of this project uses unicast and broadcast delivery modes.

2.2.4 Establishing a ZigBee Network

The procedure to establish a network is initiated only for devices which are coordinators, and which are not currently joined to a network [zig08b]. This procedure starts with an energy detection scan, sending beacons over either a specified set of channels or a pre-configured one. The scan searches for possible interferences in order to detect the channels with best energy level, on which to establish the network.

The scan returns a list of found PANs and finds the channels with the lowest number of existing networks. If a suitable channel is found, the 16-bit PAN ID is chosen. This can be done choosing a random PAN identifier less than 0xffff that is not already in use on the selected channel or using the pre-configured value. Once a PAN identifier is selected; the 16-bit network address of the module is set to 0x0000 indicating that this device is the coordinator. When a network address is set, the 64-bit PAN ID is initialized with a constant. In this moment the PAN starts up and the ZigBee coordinator initialization is performed.

When a device attempts to join a network, it performs a network discovery procedure in its personal operating space. The first step is to scan the channels sending beacons and wait for the answer of either a coordinator or a router for a permission of joining a network.

The procedure for permitting devices to join a network can be done only for devices that are either the ZigBee coordinator or a ZigBee router. Once a router or coordinator receive a scan beacon, they send back a message with a description of the network, containing the ZigBee version, stack profile, extended PAN ID, PAN ID, logical channel, and information on whether it is permitting joining or not. This information is checked by the scanning device, if the protocol version matches with

its own, save the information in a table with the other received beacons; in the other case, the message is ignored.

Then the device selects a suitable parent device, i.e. a device that fulfills the following conditions:

- The device belongs to a conformed network
- It has a good link quality[6]
- The device is open to join requests

If there are no suitable parents on the list, the device stays in a state of looking for a network to join. If there are more than one suitable parents, the device which is at a minimum depth[7] from the ZigBee coordinator may be chosen. If more than one device has a minimum depth, the selection is at random. Once the parent is selected, the device sends requests to associate itself to the network. In response, it receives a unique 16-bit logical address which the device can use in its transmissions. The relationship between *Parent-child* is established and the configuration device, such as PAN ID, network state, logical address, is set.

2.2.5 Path loss theory

Path loss is the reduction in power density or attenuation of an electromagnetic wave as it propagates through space. When calculating the path loss, several effects must be considered, such as the free space loss, attenuation and scattering.

Signal power is diminished by geometric spreading of the wave front, commonly known as free space loss. Ignoring everything else, the further away the two communication devices are, the smaller the received signal is due to free space loss. This is independent from the environment, and depends only on the distance. This loss happens because the radiated signal energy expands as a function of the distance from the transmitter. This function is described by the Friis' transmission equation for free space propagation [CALT10]:

$$P_r = P_t + G_r + G_t + 20log(\frac{\lambda}{4\pi}) + 20log(d) \tag{2.8}$$

[6]Determinated for the ZigBee specifications, sub-clause 3.6.3.1[zig08b]
[7]Number of hops

Where:

$$
\begin{aligned}
P_r &= Receiver\ power\ in\ dBm \\
P_t &= Transmitter\ power\ in\ dBm \\
G_r &= Receiver\ antenna\ gain\ in\ dB \\
G_t &= Transmitter\ antenna\ gain\ in\ dB \\
d &= Distance\ between\ the\ transmitter\ and\ receiver\ in\ meter
\end{aligned}
$$

$$\lambda = \frac{c}{f} = \frac{speed\ of\ light}{frequency} \tag{2.9}$$

The free space path loss is defined as [CP96]

$$L = (\frac{4\pi d}{\lambda})^2 = (\frac{4\pi df}{c})^2 \tag{2.10}$$

And in dB

$$
\begin{aligned}
L(dB) &= 20log(d) + 20log(f) + 20log(\frac{4\pi}{c}) \\
&= 20log(d) + 20log(\frac{4\pi}{\lambda})
\end{aligned}
\tag{2.11}
$$

The second contribution to the path loss is given by attenuation. This takes place as some of the signal power is absorbed when the wave passes through solid objects such as walls, windows or floors of buildings, for instance. Attenuation can greatly vary depending upon the structure of the object the signal is passing through, and it is very difficult to quantify. The most convenient way to express its contribution to the total loss is by adding an "allowed loss" to the free space.

Along the link path, the RF energy leaves the transmitting antenna and energy spreads out. Some of the RF energy reaches the receiving antenna directly, while some bounces off the ground. Part of the RF energy which does this, reaches the receiving antenna. Since the reflected signal has a longer way to travel, it arrives at the receiving antenna later than the direct signal. This effect is called multi-path, fading or signal dispersion. In some cases reflected signals add together and do not cause problems. When they add together out of phase, the received signal is almost worthless. In some cases, the signal at the receiving antenna can be canceled by the reflected signals [AWC+05] [EGT+99].

When free space loss, attenuation, scattering are combined and the "allowed losses" considered, the path loss equation (2.11) is transformed to:

$$L(dB) = 20log(d) + 20log(\frac{4\pi}{\lambda}) + Losses \tag{2.12}$$

The environment can bring further signal loss, and should be considered for an exact evaluation of the link. The environment is in fact a very important factor, and should never be neglected. For example, a simple way of applying the effects of scattering in the calculation of the path loss is to change the exponent of the distance factor of the free space loss formula (2.12). The exponent tends to increase with the range in an environment with a lot of scattering. An exponent of 3 can be used in an outdoor environment with trees, while one of 4 can be used for an indoor environment [SLA+03].

$$L(dB) = 10Nlog(d) + 20log(\frac{4\pi}{\lambda}) + Losses \tag{2.13}$$

Link budget

The maximum reliable data transmission range on a wireless system depends directly on the link budget, which can be characterized by the following factors:

- Transmit power. It is expressed in milliwatts or in dBm. Transmit Power ranges from 30mW to 200mW or more. TX power is often dependent on the transmission rate.

- Antenna gain. Antennas are passive devices that create the effect of amplification by virtue of their physical shape. Antennas have the same characteristics when receiving and transmitting. So a 12 dBi antenna is simply a 12 dBi antenna, without specifying if it is in transmission or reception mode [Mes06].

- Sensitivity of the receiver. The minimum sensitivity is always expressed as a negative dBm (-dBm) and is the lowest power of signal the radio can distinguish. The minimum value is dependent upon rate, and as a general rule the lowest rate (1 Mbps) has the greatest sensitivity. The minimum will be typically in the range of -75 to -95 dBm [Mes06].

- Cable Losses. Some of the signal's energy is lost in the cables, the connectors and other components, going from the radios to the antennas. The loss depends on the type of cable used and on its length. Signal loss for short coaxial cables including connectors is quite low, in the range of 2-3 dB. It is advisable to use shorter cables as possible [Mes06] [FBR+94].

The link budget is defined as [Rao07]:

$$LB(dB) = P_t\,(dB) + G_r + G_t - receive\ sensitivity \tag{2.14}$$

And if the total path-loss between the transmitter and the receiver is greater than the link budget, there is "loss" of data and communications cannot be achieved. To evaluate the link quality; one must know the characteristics of the equipment being used and evaluate the path loss. It should only be added the transmitter power of one side of the link. If it is using different radios on either side of the link, it should be calculated the path loss twice, once for each direction (using the appropriate transmitter power for each calculation). Adding up all the gains and subtracting all the losses gives:

$$Gain = P_t + G_r + G_t - losses_r - losses_t \tag{2.15}$$

Signal level

The signal level at one side of the link is:

$$signal\ level = Gain - path\ loss \tag{2.16}$$

The resulting signal level should be greater than the minimum sensitivity of the receiver level for the link feasibility. It means that the received signal is powerful enough for the radios to use it. On a given path, the variation in path loss, attenuation and multi-path over a period of time can be large, so a certain margin (difference between the signal level and the minimum sensitivity of the receiver level) should be considered. This margin is the amount of signal above the sensitivity of radio that should be received in order to ensure a stable, high quality radio link. A margin of error of 20dB is recommended [Mes06].

The relationship between the signal level and the distance is given for the combination between equation (2.13) and (2.16):

$$signal\ level = Gain - 10Nlog(d) - 20log(\frac{4\pi}{\lambda}) - Losses \tag{2.17}$$

Where *Losses* are the margin error of 20dB plus the losses due to connections, cables, etc. Equation (2.17) in terms of the *transmission power*, equation (2.15) is:

$$signal\ level = P_t + G_r + G_t - 10Nlog(d) - 20log(\frac{4\pi}{\lambda}) - Total\ losses \tag{2.18}$$

2.3 Digital filters

Filtering is defined as the processing of a time-domain signal that results in the reduction of some unwanted input components of the signal. Specifically, a digital filter processes a sequence of discrete sample values, while analog filters transform continuous signals [Lyo97]. As the nature of the signals in this work are digital, this section is focus on digital filters.

There are typically two kinds of digital filters in terms of feedback: Finite Impulse Response (FIR) filters, which use only the current and past input samples; and Infinite Impulse Response (IIR) filters, where the output depends on the previous input samples and previous filter outputs.

The basic block diagram for an FIR filter of length M is shown in Figure 2.7. The delays represent the operations on prior input samples, the coefficients h_k are used for multiplication, and the output at time n is the result of the sum of all the delayed samples multiplied by the appropriate coefficients. If all the coefficients are equal to the inverse of the filter's length $\frac{1}{M}$, the FIR is an averager, a filter that takes the current and some past input samples and then calculates the average, one of the most simplest FIR filters. This FIR behaves like a low-pass filter and smoothing sudden changes in the input using only input samples. Such filter is considered in Section 2.3.1 as Smooth processing Algorithm.

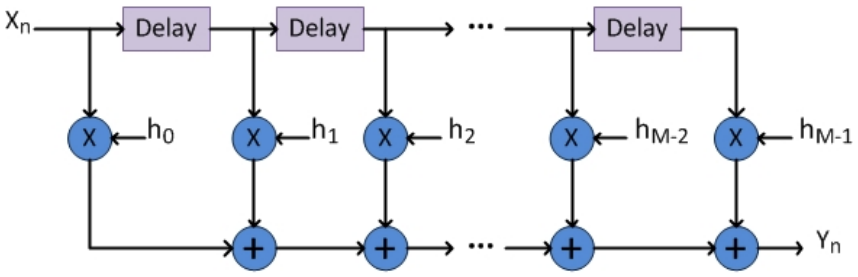

Figure 2.7: FIR block diagram

The IIR filter are very efficient and faster than FIR because they use fewer multiplications per filter output sample to achieve a given frequency magnitude response [Lyo97]. But the feedback in these filters can cause instability and indefinitely oscillations. The IIR is introduced in Section 2.3.2

2.3.1 Smooth processing algorithm

When a signal presents random and fast changes in amplitude due to noise, this behavior can be reduced by processing the signal with FIR filters such as smoothing algorithms. The nth output of a general M-tap FIR filter, see Figure 2.7, is described in equation (2.19) with the coefficients equal to h_k and $x(n-k)$ as the input samples.

$$y(n) = \sum_{k=0}^{M-1} h_k x(n-k) \tag{2.3.19}$$

Specifically in smoothing, the signal's points that are higher than the immediately adjacent points are reduced, and the lowers than the adjacent points are increased [O'H09]. The simplest smoothing algorithm is the rectangular or unweighted sliding-average smooth. This algorithm replaces each point in the signal with the average of w adjacent points, where w is the width or in terms of the FIR filters is *w-tap*. The equation for a 3 point smooth, i.e. $w = 3$ is given by equation (2.3.20).

$$S_i = \frac{Y_{i-2} + Y_{i-1} + Y_i}{3} \tag{2.3.20}$$

This rectangular smooth filter cannot reduce the *white noise*[8] presented in the signal [O'H09]. The coefficients of this filter h_k are equal to $\frac{1}{3}$ and corresponds to a FIR filter of 3-tap.

The second smoothing algorithm is named triangular smooth. This type of smoothing algorithm is like the rectangular smooth, except it implements a weighted smoothing function. This is equivalent to two passes of a 3-point rectangular smooth that results in a more effective reduction of the high-frequency noise [GD10]. The equation (2.3.21) shows the smooth function with a width $w = 5$.

$$S_i = \frac{Y_{i-4} + 2Y_{i-3} + 3Y_{i-2} + 2Y_{i-11} + Y_i}{9} \tag{2.3.21}$$

The smooth coefficients h_k are symmetrically balanced around the central point. The white noise is reduced by a factor of 0.8 with this algorithm [O'H09].

The pseudo-Gaussian smoothing algorithm is the result of passing a third time the rectangular smooth operations. In this case, the high-frequency noises can be reduced and also the white noise by a factor of approximately 0.7 [O'H09]. This

[8] evenly distributed over all frequencies

process can be performed more times but the noise reduction for white noise is every time lower. The equation for a 7 point smooth, i.e. $w = 7$ is given by equation (2.3.22).

$$S_i = \frac{Y_{i-3} + 3Y_{i-2} + 6Y_{i-1} + 7Y_i + 6Y_{i+1} + 3_{i+2} + Y_{i+3}}{27} \tag{2.3.22}$$

2.3.2 Infinite Impulse Response Filter

IIR filters are recursive filters, which in addition to input values also use previous output values. As it was described before, to achieve a given frequency response characteristic using a IIR filter requires a much lower order filter, than the equivalent FIR filter.

The order of a IIR filter is the largest number of previous input samples or output values required to compute the current output. In practice, recursive filters usually require the same number of previous inputs and outputs.

The structure of the IIR is based on the FIR. From equation (2.19) a 3-tap FIR filter will be equal to:

$$y(n) = h_0x(n) + h_1x(n-1) + h_2x(n-2) \tag{2.3.23}$$

The equation 2.3.23 is computationally identical to the *y(n)* in Figure 2.8 of the IIR.

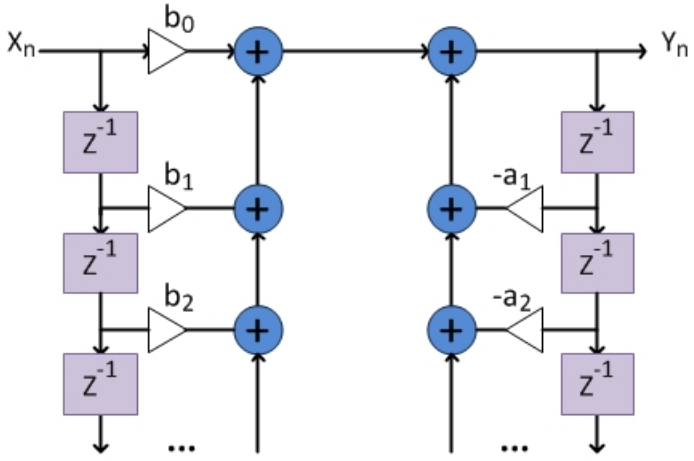

Figure 2.8: IIR bock diagram

The complete equation for a 2th order IIR filter is:

$$\begin{aligned} y(n) &= h_0x(n) + h_1x(n-1) + h_2x(n-2) \\ &\quad + a_1y(n-1) + a_2y(n-2) \end{aligned} \tag{2.3.24}$$

The coefficients of the filter h_k and a_k can be calculated by using one of the several available designing IIR filter's methods. These techniques use the z-transform and can be classified into three classes: The optimization methods, impulse variance and bilinear transform [Lyo97].

One option for calculating the coefficients is with the bilinear transformation Butterworth function [SB98], which is a mathematical function used to produce maximally flat magnitude responses. The filters designed based on this function have no amplitude ripple. One advantage of this technique is the existence of formulas for this filter.

Any multi-order Butterworth filter can be designed by using normalized Butterworth polynomials shown in table 2.1 in the *s-domain*: For example, the transfer

Order	Normalized Denominator Polynomials in Factored Form
1	$(1+s)$
2	$(1+1.414s+s^2)$
3	$(1+s)(1+s+s^2)$
4	$(1+0.765s+s^2)(1+1.618s+s^2)$

Table 2.1: Butterworth polynomials

function of a second-order Butterworth filter has a denominator the polynomial of order 2, it is described in equation (2.3.25):

$$H(s) = \frac{1}{(1+1.414s+s^2)} \tag{2.3.25}$$

The design an N order IIR digital filter can be done through a mathematical software tool, like MATLAB and the Butterworth filter, which returns the filter coefficients h_k and a_k.

3 Localization system implementation with a dynamic ZigBee coordinator selection

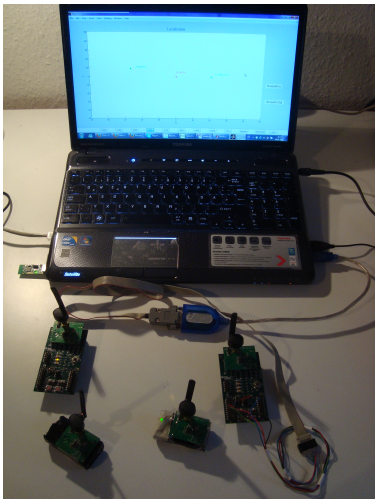

Figure 3.1: Implementation

In a robotic system, whose organism can physically evolve into a symbiotic swarm system of connected robots, it is required that each device knows the existence of the others, their role in the system (i.e. what type of organism: ”scout”, ”Backbone” or ”active wheel”), and their location.

In addition, as it was described before, the organisms should be non-complex, and have a robust, power and programming low cost, wireless communication system. For these reasons, the ZigBee standard was selected not only as communication protocol but as the localization resource as well. The details of the hardware and software platforms are presented in Section 3.1.

The theoretical function of the distance in terms of the RSSI, losses and features of the hardware used in this work, is derived from the path loss model described

in chapter 2 in Section 3.1.2. The expected distance with different RSSI and losses values and the expected RSSI values at various distances are calculated. These calculations allows the understanding of the distance behavior presented in the subsequent chapter 4.

The implementation and the main components of the localization system are described in four parts; each one is discussed in Section 3.2. The first part concerns the RSSI data processing, described in Section 3.2.1. It includes the function which weights the strength of the signal and its posterior processing, which can be either filtering the signal with an Infinite Impulse Response filter (IIR) or using a smooth algorithm.

The distance estimation is the second part, focus of Section 3.2.2. It covers the module characterization for the obtaining of the distance equations as function of the RSSI, and their implementation. Finally, the determination of the position's coordinates and the data processing is described in 3.2.4.

A calibration function was implemented, which is used to decrease the errors produced by the variations of the RSSI and their propagation in the distance estimation and coordinates calculation. The details of this function are presented in Section 3.2.3

The final Section of 3.2 presents the visualization and the description of the system's user interface. MATLAB was the development platform used for the localization implementation and the visualization.

The ZigBee standard necessitates a single coordinator per network. Therefore, the device type is normally pre-configured in this system. However, this symbiotic system cannot depend on any device for the wireless communications and network creation. In fact, the system may change at any moment and a device may or may not be part of the system. Thus, a dynamic coordinator selection feature was designed and implemented, where any device can be either coordinator or router, depending on a given priority and its time of initialization. This implementation is expounded in Section 3.3.

Once the network is conformed, an embedded application, under *ZigBee Stack*, gets the RSSI of all devices in the network and communicates these values serially to the MATLAB application, allowing the distance calculation. The use of MATLAB is due to the need of fast assessment of the algorithms, but as a future work, this application may be embedded, simplifying the overall system. This chapter

provides the description of the RSSI propagation algorithm and also the MATLAB application.

3.1 System design

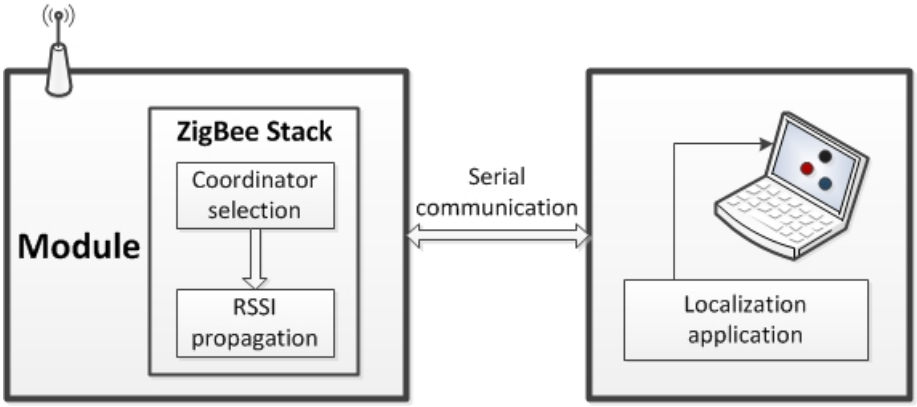

Figure 3.2: System architecture

The system implemented consists of two main parts: The embedded system and the MATLAB application. The first one was developed with ZigBee modules which have a System On Chip (SoC). The embedded software was programmed under the ZigBee standard stack and is composed of the coordinator selection, the application to obtain the strength of the signals and the application used to serially perform the communications with the MATLAB application. The system architecture is shown in Figure 3.2

3.1.1 Hardware description

ZigBee at 2.4GHz has three different configurations; they are depicted in Figure 3.4. The configuration of the Figure *A* consists of a transceiver for the physical and MAC layer, and a micro-controller for the stack and the application. This configuration is mainly used when a proprietary stack will be used and therefore, it can be embedded in the same micro-component with the application. The architecture *B* uses just one SoC for all layers and the stack, consequently, it has low power consumption, lower cost in space and number of components. Finally, the configuration *C* has a SoC for physical - MAC layer and the application; and an additional micro-controller for the application. This configuration is more suitable when the application is too big for the program memory space of the SoC or when the control of additional interfaces, which the SoC does not provide, is required.

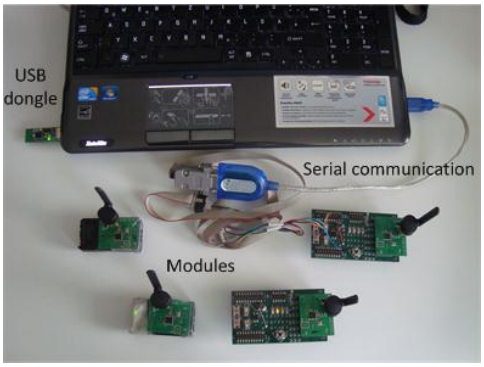

Figure 3.3: Hardware

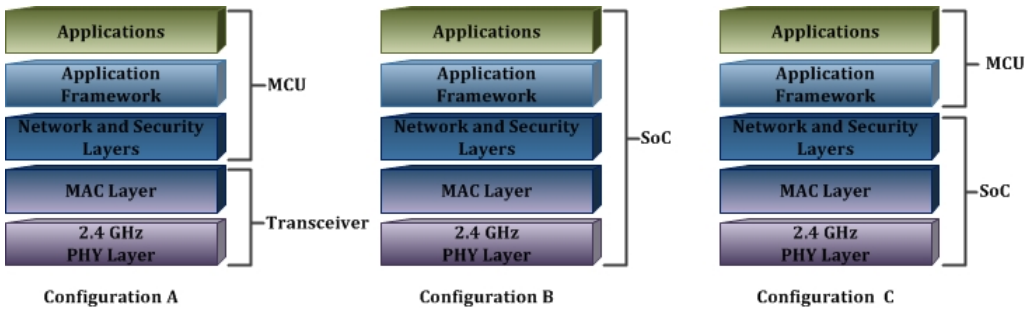

Figure 3.4: Hardware configuration

The architecture *B* was selected because it allows a fast implementation, the design of the system's radio frequency components is not required (which is a time consuming and nontrivial task), and it permits a rapid assessment of the technology, stack and design.

There are various SoC alternatives on the market. The Texas Instrument module CC2530em, see Figure 3.5, was chosen for its low cost, which includes modules and programming tools; program memory for the stack and applications. Furthermore, simplicity in terms of programming, debugging and hardware; and an open source license for its stack are additional advantages[1].

[1] At the beginning of the project, the wireless development tool eZ430-RF2500 was used at a cost of 48 Euros; it includes 2 USB modules but it does not support the ZigBee stack, because the memory space is not sufficient. Simple development kits (2 modules and programmer) from other manufactures, can cost more than 340 Euros, but their modules include more peripherals, such as LCD or USB port, which are not required for this project. Other important features, like power consumption and link budget, from different modules are quite similar

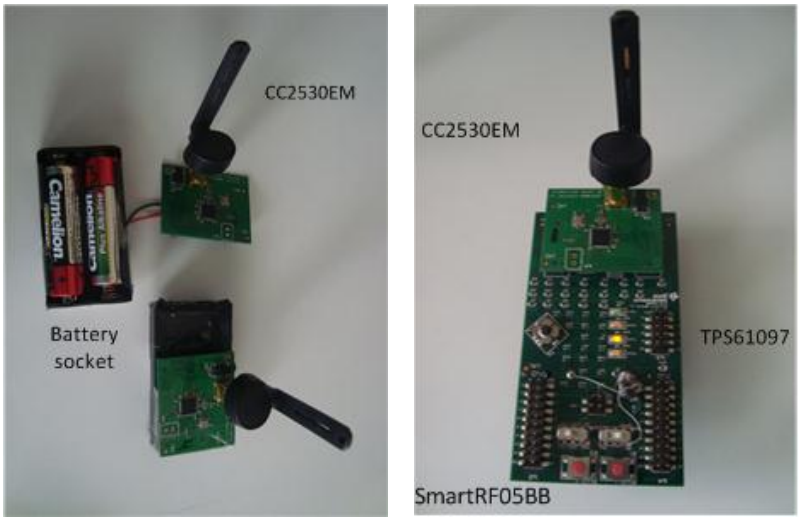

Figure 3.5: CC2530em modules

The modules CC2530em have a 32.768 kHz crystal, general IO headers/connectors, external passive components for the balun and antenna match filter and an SMA connector for the antenna or any other RF instrument connection. The principal features of the modules are summarized in table 3.1 [Ins09].

Feature	Description
Link budget	101.5dB
Max. Power consumption	37.5mA
Operating Supply Voltage	2-3.6V
Flash memory	256KB
Stack size	75KB
USARTs	2
ADC	12-Bit with 8 Channels

Table 3.1: CC2530em features

The system consists of 4 modules as is shown in Figure 3.3. Two of the modules are plugged into the SmartRF05BB (Battery Board) board, Figure 3.5 right image, which provides a standalone node when the CC2530EM is connected. The SmartRF05BB powers the modules with 2 AA batteries whose sockets are on the underneath of the board. In addition, it connects the I/O module's pins to some peripherals, such as LEDs, push buttons, joystick, debug connector connection for

debugging and programming tools [Tex09a]. The modules can be connected to a computer using a serial-USB cable and a RS232 transceiver, as is shown in Figure 3.3. The other two modules are connected to the battery socket for powering, as it can be observed in Figure 3.5 left image.

The modules use a 2.4GHz swivel SMA antenna with linear polarization. Its electrical characteristics are shown in table 3.2 [Ant11]. The type of the antenna is *whip*, which radiates equally in all azimuthal directions (perpendicular to the antenna's axis) but varies with elevation [Sey05]. The theoretical pattern is depicted in Figure 3.6[2] where the azimuthal plane is XY.

Feature	Typical performance
Peak gain	2.2 dBi
Average gain	-1.0 dBi
Maximum Return Loss	-13 dB

Table 3.2: Antenna electrical characteristics

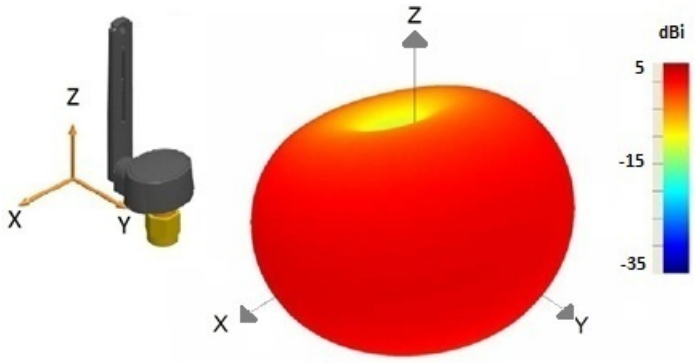

Figure 3.6: Antenna pattern

The RSSI is a signal that depends on several conditions, for example transmission power (see Section 2.2.5). Consequently, it depends on the power supply's charge. As the modules are connected to two batteries, it was observed a change in the strength of the signal up to 30dB produced by the voltage deterioration. Therefore, the use of a boost converter *TPD61097* was seen as appropriate solution. This converter provides a power supply solution for battery-powered devices, giving a constant programmed output voltage while the input has a voltage range lower

[2]Image taken from [Ant11]

than the output and higher than 0.9V [Tex09b]. The output voltage was set to 3.3V for this application.

A USB dongle *CC2531 USB Dongle* was used to capture packets being sent over the air. This dongle, see Figure 3.3 is already pre-programmed and it was used in conjunction with the SmartRF Packet Sniffer to observe the ZigBee frames. More detailed information about the Packet Sniffer and these frames is given in Section 3.3.2.

3.1.2 Path loss effect

The modules used in this project have a maximum transmission power of 4.5dB [Ins09]; At 2.4GHz, λ is equal to 0.12491m. Assuming the gain on the antennas is 0 dBm equation (2.18) is transformed into:

$$\begin{aligned} signal\ level &= 4.5dBm - 10Nlog(d) - 20log(\frac{4\pi}{\lambda}) - Total\ losses \\ &= 4.5dBm - 10Nlog(d) - 40dB - Total\ losses \end{aligned} \tag{3.1.1}$$

In free space where N is 3 and with different distances and assuming the total losses (L) equal to 20dB, 10dB and the ideal case 0dB, the signal levels expected at the receiver are collected in table 3.3

$$\begin{aligned} signal\ level &= 4.5dBm - 10Nlog(d) - 40dB - Total\ losses \\ &= 4.5dBm - 30log(d) - Total\ losses \end{aligned} \tag{3.1.2}$$

Distance	RSSI (L = 20dB)	RSSI (L = 10dB)	RSSI (L = 0dB)
10m	-85.5	-75.5	-65.5
5m	-76.47	-66.47	-56.47
2m	-64.53	-54.53	-44.53
1.5m	-60,78	-50,78	-40,78
1m	-55.5	-45.5	-55.5
75cm	-51.75	-41.75	-51.75
50cm	-46.47	-36.47	-46.47
25cm	-37,44	-27,44	-37,44
10cm	-25.5	-15.5	-25.5
5cm	-16.47	-6.47	3.53

Table 3.3: Signal level expected

The expected distance as a function of the strength of the signal is presented in table 3.4. The values correspond to different signals levels, assuming the total losses equal to 20dB, 10dB and the ideal case 0dB, like in the case before, and using the equation (3.1.3) based on (3.1.2).

$$\begin{aligned} signal\ level &= -35.5dB - 30log(d) - Total\ losses \\ log(d) &= \frac{-35.5dB - Total\ losses - signal\ level}{30} \\ ln(d) &= \frac{-35.5dB - Total\ losses - signal\ level}{13.03} \\ d &= e^{\frac{-35.5dB - Total\ losses - signal\ level}{13.03}} \end{aligned} \tag{3.1.3}$$

RSSI (dB)	Distance (L = 20dB)	Distance (L = 10dB)	Distance (L = 0dB)
-10	3.04	6.56	14.13
-20	6.56	14.13	30.43
-30	14.13	30.43	65.56
-40	30.43	65.56	141.25
-50	65.56	141.25	304.32
-60	141.25	304.32	655.64
-70	304.32	655.64	1412.54
-80	655.64	1412.54	3043.22
-90	1412.54	3043.22	6556.42

Table 3.4: Expected distance in cm

3.1.3 Software platforms

The software was developed on two different platforms: The embedded software is based on the ZigBee standard; specifically, the ZigBee stack by Texas Instrument: *Z-Stack*^{TM} was used. The localization algorithms were implemented in MATLAB.

Z-Stack

Z-Stack^{TM}is the *Texas Instrument* protocol stack which is compliant with the ZigBee®2007 (ZigBee and ZigBee PRO[3]). Consequently, Z-Stack features [ZS11]:

- Low power consumption and low data rate (approximately 100Kb/s)

[3]ZigBee and ZigBee PRO are two different stack specifications defined by ZigBee Alliance specification

- Self forming and repairing mesh networking
- Interoperability
- MAC and application level acknowledgments
- Fragmentation
- Frequency Agility
- Security

In addition, it has an operating system with task scheduling and a Hardware Abstraction Layer (HAL) which modularly drives peripherals such as LEDs, ADC, keys UART and the PA/LNA. There is an Application Programming Interface (API) that allows to separate the operating system and the stack from the user software. It shields the stack and makes the application portable. The main functions of the API are [Tex09c]:

- Task Management. There are functions for adding and managing tasks in the operating system. Each task consists of an *initialization function* and an *event processing function*.
- Task Synchronization. It enables a task to wait for events to happen and to return control while waiting; to set events for a task and to notify the task once a event is set.
- Message management and exchange between tasks or processing elements, such as, interrupt service routines. It includes message buffer allocation and de-allocation, sending command messages to other tasks and receiving response messages.
- Timer Management. It enables the use of timers (start, stop a timer and set an event when the time is reached)
- Interrupt Management. With this function the tasks can interface with external interrupts.
- Memory Management, dynamic memory allocation.
- Non-Volatile Memory use

The stack and operating system occupy approximately 75Kb of memory space. IAR Embedded Workbench was used as compiler and debugger with a free student license.

MATLAB

MATLAB is a high-level language for technical computing and a development environment for data management, visualization and analysis. It supports programming languages such as C and C++ and comes with several toolboxes[4], mathematical functions and integration with external applications. These features were very useful in the implementation phase of this project.

3.2 Localization system realization

The distance estimation is based on the strength of the signal. As mentioned above before, this method has low power consumption, does not require additional hardware and has low programming costs. But it has the drawback of poor accuracy. The distance error range using this method, depends on the use of anchors, node's and anchors (if used) density, connectivity and refinement process [Pal10]. Applications which use several nodes (more than 100) [CSL02] [ASS02], or have large connectivity (more than 7 nodes) [SRB01] or use combination of different sensors, such as [ASS02], present reductions in the errors, but this is associated with an increase in hardware or software cost.

The localization solution presented in this thesis, has to fulfill the following requirements: the topology of the network is undetermined, there are no anchors, the devices may be mobile, the node's density is low (minimum 3 up to 16), it should be low computing cost; and finally, all robots in the network should be covered and located.

The algorithm proposed is a combination of different kinds of solutions described in Section 2.1. The implementation is a distributed, anchor-free, self-configuring and concurrent algorithm, whose nodes cooperate with each other to estimate their location based on the RSSI. The employed localization method is Eucledian multilateration, which requires at least 3 nodes and the distance among them to estimate their positions. These positions are iteratively recalculated and updated. The refinement is an implicit and immerse process in the localization algorithm that consists of iterative multi-lateration, Heuristic, Weighted Least Square and two kinds of filters, FIR and IIR.

The solution proposed has 4 main phases:

1. RSSI data processing

[4]Toolboxes are a collection of special-purpose MATLAB functions, which are available separately

2. Distance estimation
3. Calibration
4. Coordinate calculation

The strength of the signal is transmitted serially from the nodes to a computer with the MATLAB localization application. The configuration of the serial port is illustrated in table 3.5.

Feature	Description
COM number	5
Baud rate	115200
Data bits	8
Parity	none
stop bits	1
Flow control	none

Table 3.5: Serial port configuration

The localization application opens the serial port and reads the messages with the RSSI information. The Figure 3.7 shows the format of these messages. The first byte is the identification number of the device, which information is contained in this frame. In the example of the Figure, this ID is ***s***. The second byte corresponds to the ID's number of the device connected to the serial port, *source device*. The next two bytes are the short ZigBee address of device ***s***, followed by a byte with its battery value. The number of devices in the network is the next byte and then appear the RSSI between device ***s*** and the other devices, that is, $[RSSI_{s0}, RSSI_{s1}, ..., RSSI_{sm}]$ being m the maximum number of devices that can be connected in the network. Each RSSI value has a length of one byte.

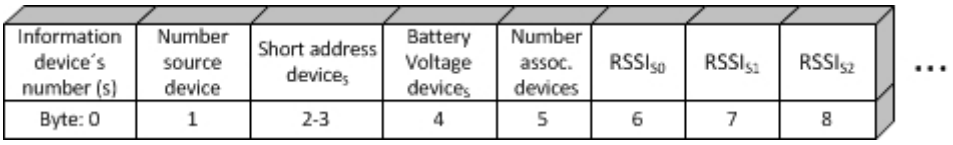

Figure 3.7: Serial message format

A device ***a***, as an example, may have a $RSSI_{ab}$ different to the $RSSI_{ba}$ from device ***b***. Therefore, a weighted function based on [CSL02] and [ASS02] algorithms was

implemented, that gives more weight to the signal measurement with the better link, that is the lower RSSI value. The purpose of this weighted function is to reduce the influence of wrong data due to interference, mismatch of the antenna and other facts. Moreover, it was necessary to filter the weighted RSSI, due to the fact that RSSI measurement errors have a strong influence on the localization [SRB01] and because the RSSI is a very noisy signal. Two alternatives for the filtering, an Infinite Impulse Response filter, which is described in Section 2.3.2 and the a pseudo-Gaussian smoothing algorithm, presented in Section 2.3.1, were implemented to alleviate these problems.

The second phase was the distance estimation. It consists of obtaining the distance equations as a function of the RSSI and its utilization. These equations depend on the environment, radio transceiver, receiver sensitivity, transmission power and antenna used, see Section 2.2.5. Therefore, an interpolation process was used to provide an optimal fit function[5]. The interpolated information is the result of the RSSI measured at several distance positions, in the range of 1cm - 10m. As expected, the distance equation is an exponential function of the RSSI. The RSSI sample space was divided into 6 regions, each one with a different distance function. The details of phase two are discussed in Section 3.2.2.

The distance values obtained from the RSSI measurements may vary from the ones obtained in the interpolation process. Thus, a calibration function was implemented that searches for this difference and calculates a correction factor for the received RSSI. This correction factor is used before the distance estimation in the localization application, in order to avoid error propagation in the estimated distance and in the coordinates calculation. This function is described in 3.2.3.

After the distance estimation is performed, the coordinates of each robot are calculated. This phase is explained in Section 3.2.4. This estimation process uses the multi-lateration approach, used in [SRB01] and described in Section 2.1.2. Finally, the visualization of the nodes is described and the user interface is explained in Section 3.2.5.

3.2.1 RSSI data processing

The strength of the signal is a measurement that heavily depends on the environment, radios, antenna, position etc. Consequently the RSSI between two nodes may be different, i.e. the value measured by node a from node b $(RSSI_{ab})$ is not necessary the same as measured by node b from node a $(RSSI_{ba})$. In order to avoid

[5] I.e. correlation closes to 1

errors produced by this difference, a weighted function based on the applications presented in [CSL02] and [ASS02] was implemented

This function gives more weight to the signals with better link, that is the lower RSSI value. The weighted function considers 3 ranges:

1. If the difference between the measurement signals is lower than 3dB
2. If the RSSI values differ more than 3 dB and lower than 6dB
3. Finally, if the difference is bigger than 6dB

The purpose of the weight is to reduce the influence of the worst measurement. In the first case, an average of the signals is calculated. For the second, the value with the best link has a weight 83%. And in the third case, the difference of weights is 90% to 10%.

The result of the weighted RSSI can either be filtered with an IIR or smoothed with a pseudo-Gaussian smooth algorithm. The coefficients of the IIR were calculated with a second order low-pass Butterworth filter.

The smoothing algorithm has a weight of 7. The values used for the distance estimation are the means, where is located the peak of a Gaussian function. The goal of using these algorithms is to compare the behavior of the distance when it is filtered with IIR or simply smoothed.

3.2.2 Distance estimation description

A distance function in term of the RSSI is a function which is strongly dependent on the line-of-sight between the radios, antennas, environment conditions, power transmission etc. Therefore, an interpolation process was used.

For this purpose, several RSSI measurements were taken from the 4 modules at 80 different distances in a range of $[1cm - 10m]$. The average of the measured values was calculated with a weighted error. Then, MATLAB was used to find the equation with an interpolation method. The Figure 3.8 depict the results of the measurements with their errors.

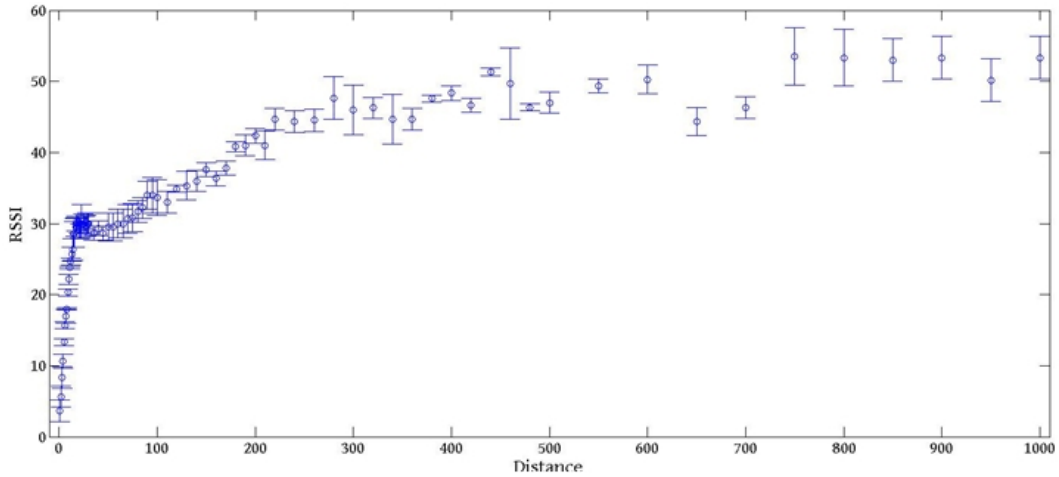

Figure 3.8: RSSI vs. Distance

The results of the measurements were divided into 6 intervals for calculating the distance functions. In most of the cases, the equation with best correlation is logarithmic as expected. The logarithmic equations with best correlation and those which match better with the next interval, were chosen, i.e. the values in the limits have a difference lower than the 10% of the distance in this point. The 6 resulting curves are presented as follows:

For a range between $[0dB - 25.5dB]$ Figure 3.9 and the logarithmic equation utilized is (3.2.2) that has a correlation of 0.96.

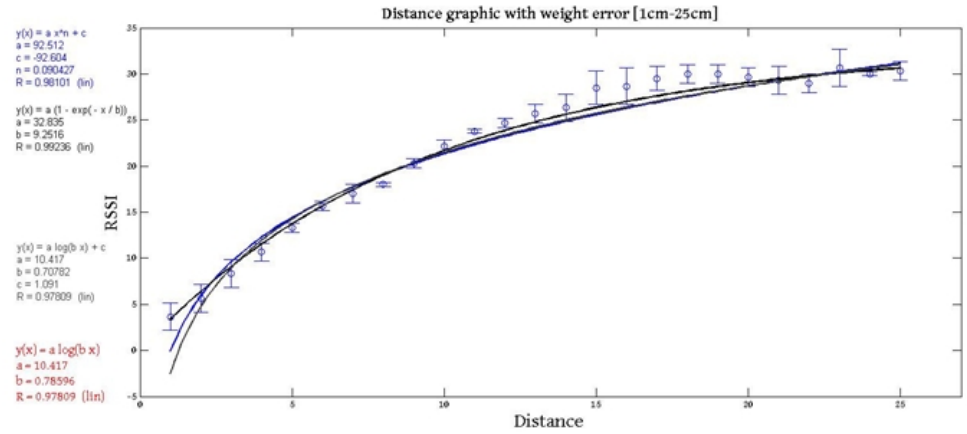

Figure 3.9: Distance fitting between 0dB - 25.5dB

$$RSSI = 10.417Log(0.78596X)$$
$$RSSI = 4.524Log_{10}(0.78596X) \tag{3.2.1}$$

As a function of the RSSI:

$$X = \frac{e^{\frac{RSSI}{10.417}}}{0.78596} \tag{3.2.2}$$

The next equation (3.2.4) is between 25.5dB - 36dB. The correlation is 0.9677 and the corresponding Figure is 3.10.

$$RSSI = 7.4608Log(0.89346x) \tag{3.2.3}$$
$$X = \frac{e^{\frac{RSSI}{7.4608}}}{0.89346} \tag{3.2.4}$$

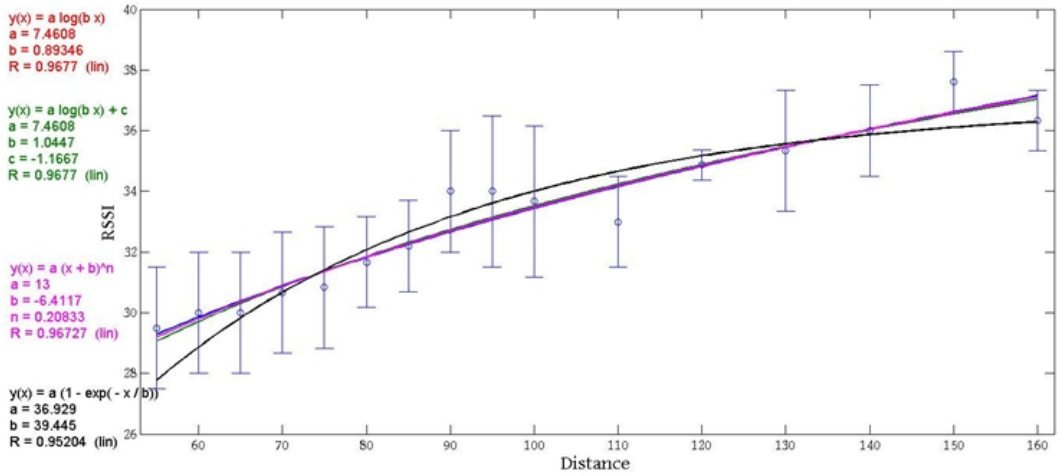

Figure 3.10: Distance fitting between 25.5dB - 36dB

The interval between 36dB - 46dB is based on equation (3.2.6) with a correlation of 0.95045, Figure 3.11:

$$RSSI = 14.357Log(0.089336x) \tag{3.2.5}$$
$$X = \frac{e^{\frac{RSSI}{14.35}}}{0.089336} \tag{3.2.6}$$

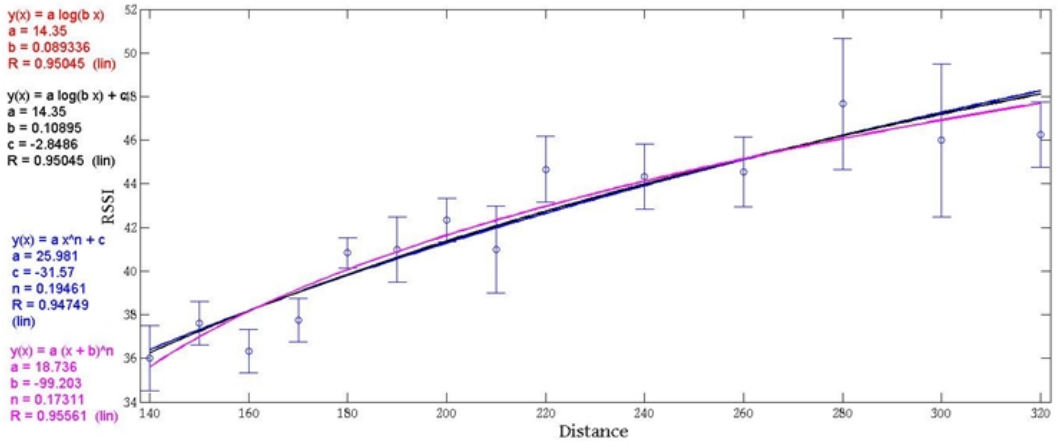

Figure 3.11: Distance fitting between 36dB - 46dB

The equation (3.2.8) represents the range $[46 - 49.5]$; it has a correlation of 0.47218. In this short interval there were several variations in the RSSI values that may represent distance from 2.4m to 7m. Therefore, it small sample was taken and the correlation is not good. See graphic 3.12.

$$RSSI = 4.395Log(3.6046x) + 15.601 \tag{3.2.7}$$

$$X = \frac{e^{\frac{RSSI-15.601}{4.395}}}{3.6046} \tag{3.2.8}$$

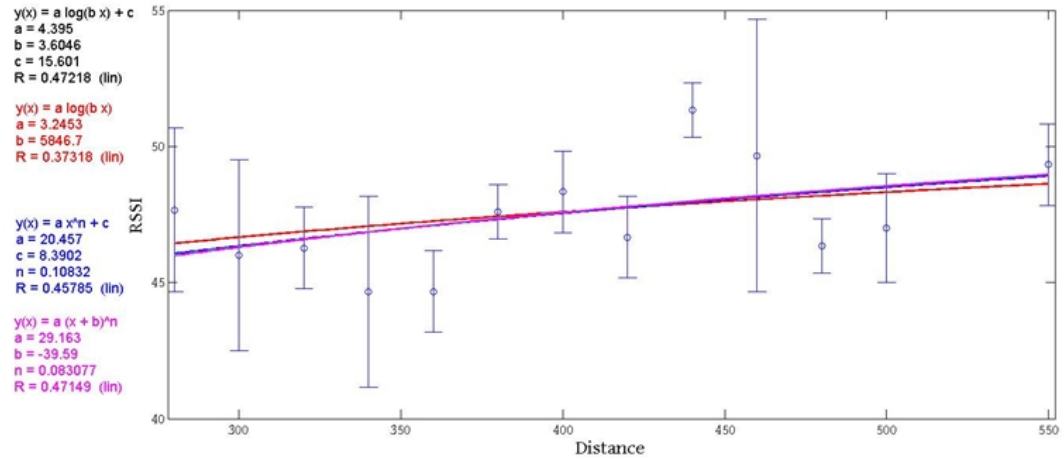

Figure 3.12: Distance fitting between 46dB - 49.5dB

The range between $[49.5dB - 53dB]$ corresponds to Figure 3.13. The equation (3.2.10) utilized has a correlation 0.54824. Again this interval is highly sensitive to noise and any change in the distance.

$$RSSI = 6.3607Log(3.86226x) \tag{3.2.9}$$

$$X = \frac{e^{\frac{RSSI}{6.3607}}}{3.8622} \tag{3.2.10}$$

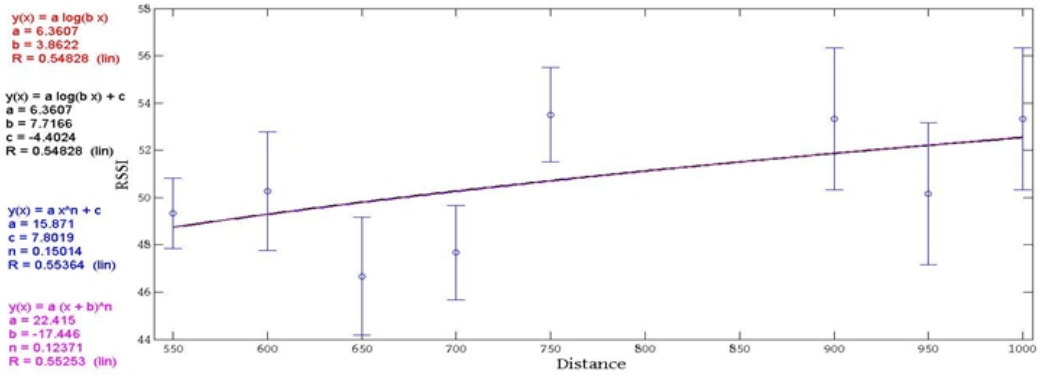

Figure 3.13: Distance fitting between 49.5dB - 53dB

The last part corresponds to the values greater than 53dB, Figure 3.14. The equation (3.2.12) has a correlation of 0.78 and was used in combination with equation (3.2.14).

$$RSSI = 7.4458Log(1.4831x) \tag{3.2.11}$$

$$X = \frac{e^{\frac{RSSI}{7.4458}}}{1.4831} \tag{3.2.12}$$

$$RSSI = 6.9759Log(2.1969x) \tag{3.2.13}$$

$$X = \frac{e^{\frac{RSSI}{6.9759}}}{2.1969} \tag{3.2.14}$$

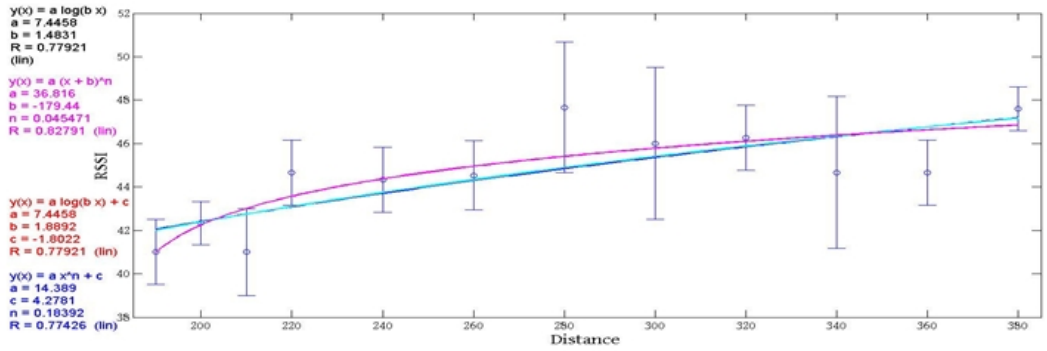

Figure 3.14: Distance fitting values bigger than 53dB

3.2.3 Calibration

The calibration function aims to reduce the variations between the RSSI measurement and the RSSI expected. These differences are caused by the irregular behavior of the signal and its dependency on several factors.

The calibration requires that the four nodes are located in the positions shown in Figure 3.15. Then, the functions starts to calculate the mean value of each RSSI for a time determined by the user.

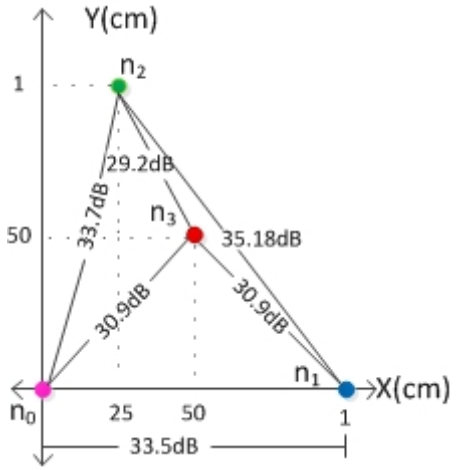

Figure 3.15: Calibration configuration and expected RSSI

The final mean values are compared with the expected ones, which are indicated in Figure 3.15, and a correction factor for each RSSI is determined. These correction factors are used after the RSSI filtering results. If the filtered RSSI is bigger than a determined value and the correction factor is bigger than zero, then RSSI is corrected. These limits attempts to avoid incorrect negative values.

3.2.4 Coordinate determination

The final localization phase is the coordinate determination. Its implementation is based on the algorithm ABC [SRB01] and consists of a multi-lateration calculation of the distances. Every node performs the coordinate calculation in parallel and independent of the other devices. The nodes are located in a Cartesian plane, which dimensions are for X and $\mathbf{Y}$ $[-10m, 10m]$.

It is assumed that each node is its own reference point, I.e. from the point of view of a node n_0, its position in the plane is $(0, 0)$. Then node n_0 orders a list of the other devices in the network, being the first device the nearest to it. This node n_1 will be the Y=*0* reference and its position is assumed at $(d_{01}, 0)$, where d_{01} is the estimated distance between n_0 and n_1. The next node n_2 in the list is located at (X_2, Y_2) with Y_2 positive. The values of (X_2, Y_2) are calculated with the equations (2.1):

$$x_2 = \frac{r_{01}^2 + r_{02}^2 - r_{12}^2}{2r_{01}}, \quad y_2 = \sqrt{r_{02}^2 - x_2^2} \quad \textit{equations (2.1)}$$

The next nodes $[n_3, n_4...]$ are located at the coordinates (X_m, Y_m). For the calculation of the position, either equation (2.3) or (2.1).

$$\begin{aligned} x_m &= \frac{r_{01}^2 + r_{0m}^2 - r_{1m}^2}{2r_{01}} \\ y_m &= \frac{r_{0m}^2 - r_{2m}^2 + x_2^2 + y_2^2 - 2x_2 y_m^2}{2y_2} \quad \textit{equations (2.3) canbeused.} \\ z_m &= \sqrt{r_{0m}^2 - x_m^2 - y_m^2} \end{aligned}$$

If node n_2 has any of its coordinates in the limits of the plane, node n_m will use equations (2.1), otherwise, it uses equations (2.3). The use of equations (2.1) by node n_m attempts to avoid an error propagation produced by the n_2 localization and due to the fact that longer distances represent larger errors.

If any coordinate $[X_1...X_{max}]$ or $[Y_2...Y_{max}]$ are out of the Cartesian plane, that means if they are $(X, Y > 10m)$ or $(X, Y < -10m)$, then this value is fixed to its nearest point of the plane (10m or -10m) for the visualization.

Finally, all coordinates within the plane and $\neq 0$ pass through a pseudo-Gaussian smooth algorithm with a history of 10 points. The purpose of the smoothing is to avoid fast and steep changes in the position of the devices. The localization process is performed every time there is new information, i.e every 300ms.

3.2.5 Visualization and user interface

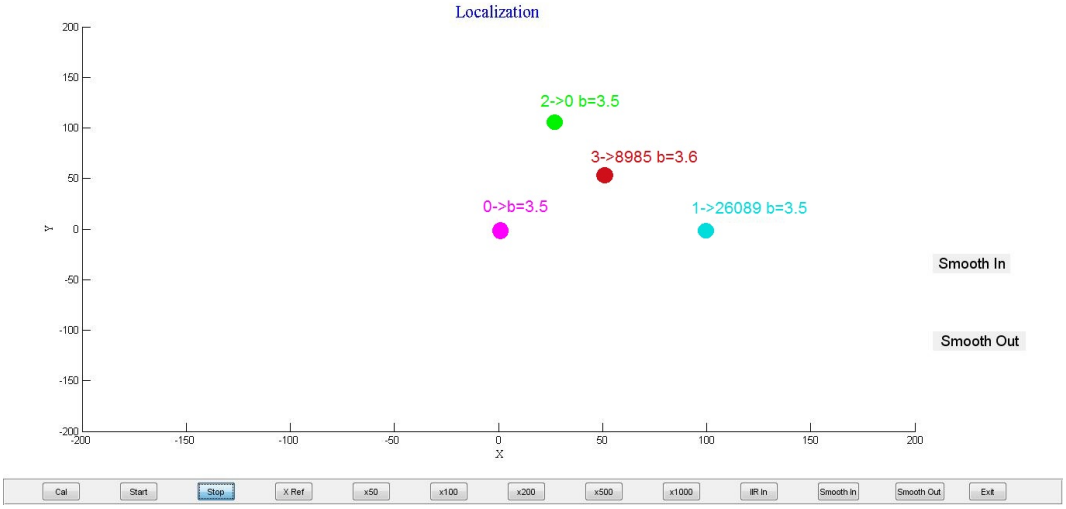

Figure 3.16: Visualization and user interface

Figure 3.16 illustrates the visualization and user interface, which consist of a Cartesian plane, buttons for the user interaction, the localization of the devices with relevant information of each one and message boxes for visualization of the processes that are used.

User interface

At the start of the application, the Cartesian plane has dimensions of $X, Y = [-10m, 10m]$. There is a menu with five toggle buttons, which can be used to change the scale into: $X, Y = [-50cm, 50cm]$, $X, Y = [-1m, 1m]$, $X, Y = [-2m, 2m]$, $X, Y = [-5m, 5m]$ and $X, Y = [-10m, 10m]$. It is shown in Figure 3.17.

Figure 3.17: Scale buttons

There are 2 buttons for stopping and starting the localization updates, see Figure 3.18. The application starts showing and updating the position of the nodes, if the stop button is pressed, then the current positions stays until the start button is pressed.

As described before, the references of the node are selected at the start of the application. If the user wants to change the reference, there is a toggle button *X Ref* for that. When *X Ref* is pressed, the order of the nodes' list is updated, being node n_1 the nearest node to n_0 and the *Y=0* reference. In Figure 3.18 appears the calibration toggle button *Cal*. When the user wants to calibrate the system, the nodes should be located in the calibration position, then the user presses this button and check how many measurements are taken. To finish the process the button should be press again, but as *Cal* is a toggle button, it does not work if a different button has been selected before. The calibration process can be activated just once during runtime of the application.

Figure 3.18: Calibration, Start, stop and reference buttons

It is possible to change the kind of input filter and the state of the output smoothing, see Figure 3.19. When the IIR is activated and the smoothing input button is pressed, the IIR is deactivated and the smoothing activated and vice versa. The output smoothing toggle button activates or deactivates this filter. A message on the screen shows which filter is running in the current state.

The button *exit* ends the program. With this button the application closes the serial communication, the Figure and all the files that are open.

The most important information is saved in text files, in order to be able to follow the application. This information is: the serial frames *frames.txt*, RSSI matrix

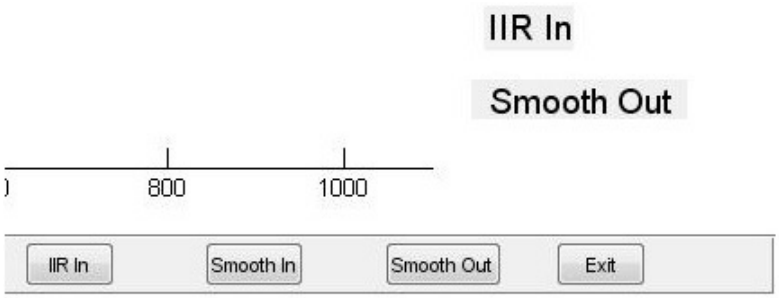

Figure 3.19: Input / Output data processing

rssimatrix.txt, weighted RSSI matrix *rssiweightmatrix.txt*, filtered RSSI matrix *RSSIfiltermatrix.txt*, history of the input smoothing or input IIR *historyfilt.txt*, estimated distance matrix *distancematrix.txt*, the RSSI history in the calibration phase *correction.txt* and the calculated coordinates *coordinates.txt*. If there is a change in the application, such as activation of a filter, this event is written in each file.

Nodes visualization

The node n_0, which is located in $(0, 0)$, is colored in magenta. It is identified with the number *0* and the value of its battery value appears above its localization, as it is illustrated in Figure 3.20. Node n_1 has the color cyan and it is identified with the number *1*; Node n_2 is the number *2* and it is green; the rest of the nodes are red and they are identified with sequential numbers. From node 2 to the number of devices in the network, their identification appear above their position together with their short ZigBee address and battery state.

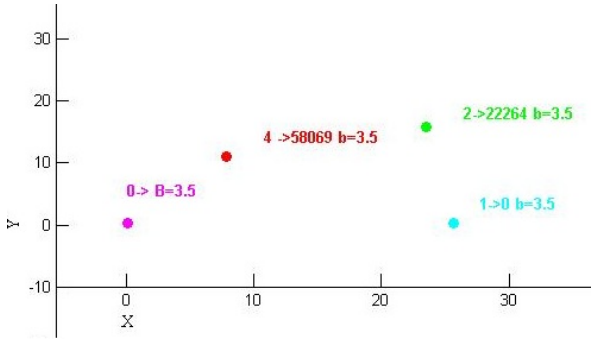

Figure 3.20: Node localization

3.3 ZigBee application

The application is restricted into channel 15, because it is one of the four channels (15, 16, 21, 22), which falls between the often-used and non-overlapping 802.11b/g channels, reducing future interference problems [GTC08].

As any robot can be turned off at any time and as they are mobile, an independent, flexible decentralized communication network in terms of device's location is necessary. Therefore, a mesh topology is configured for this project [Net10].

For the implementation of the dynamic coordinator selection each device has different priority. This selection task corresponds to the *coordSelection.c* application. When a device is switched on, it starts as a coordinator to look for other networks/coordinators in the channel. The device with the lowest PANID has the highest priority. In addition, the number of scans and the interval of time between each one, is also given for the PANID. This means that a device with higher PANID performs more scans and its time interval is longer than a device with lower PANID. The details of this process are described in Section 3.3.1.

After the coordination selection, a ZigBee network is started. All devices which were not selected as coordinator become routers, that means the network is composed of one coordinator and several routers. There are no end devices; thus, the mesh networking can be used. Once the coordinator associates a device, it starts to poll the associated devices for the propagation of the RSSI. This process is described in Section 3.3.2. The propagation task was developed in the LocalCooApp.c file .

3.3.1 Coordinator selection

Every device is programmed with the capability of being either coordinator or router. The purpose of this is to change the role of the device without the necessity of reprogramming, adding hardware or turning off the devices, methods normally used in the case of changing a feature in the stack.

The PAN ID is pre-configured with a number lower than 0xFFF0 and different one in each device. When a device changes the device type[6], i.e. from coordinator to router, it is necessary to perform a software reset. Thus the state of the network is saved in the non-volatile memory. There are two main network states: *finding coordinators* and *starting normally* the network.

[6]Device type is described in Section 2.2.1

At the beginning of the application, all devices are coordinators and check the state of the network by reading their memory. The normal initial state is *finding coordinators*. Once the coordinator selection is finished, the devices change their network state to *starting normally*, save in memory the type of device that will be used in this state and restart. When the saved network state is *starting normally* the coordination selection application is ignored and devices start the normal ZigBee process. That is, the coordinator establishes the network and routers attempt to join a network.

Device priority

The priority of the devices is given by the PAN ID, the lower the ID the higher the priority. A device with high priority will perform less scans and the interval between each other is lower. As a result, the device will become faster a coordinator compared with the devices with lower priority.

Coordinator selection algorithm

The devices start sending an active scan (Section 2.2.4) looking for other coordinators in the network. A timer is activated, waiting for a response. If it does not receive an answer, it sends the beacon again until the maximum number of re-transmissions is reached. Then, the device saves its future device type (coordinator) and the network state *starting normally* and restarts to establish the network as the coordinator with the new values.

If after a scan a device receives a response, that means that there are other coordinators or routers in the channel. Then, it compares its PAN ID with the devices which answered. If any PAN ID is lower than the own, the device changes its device type into router, the state of the network and executes a software reset. When all the PAN IDs are bigger, it continues searching.

The sending of the answer to the scans is performed by the stack and it is invisible to the application, as well as the reception of the scans responses. Thus, it was necessary to change a part of the stack and activate an event to inform the application about the scan's results[7].

The flow chart of the coordinator selection algorithm is depicted in Figure 3.21.

[7] These changes are documented in the history of the application

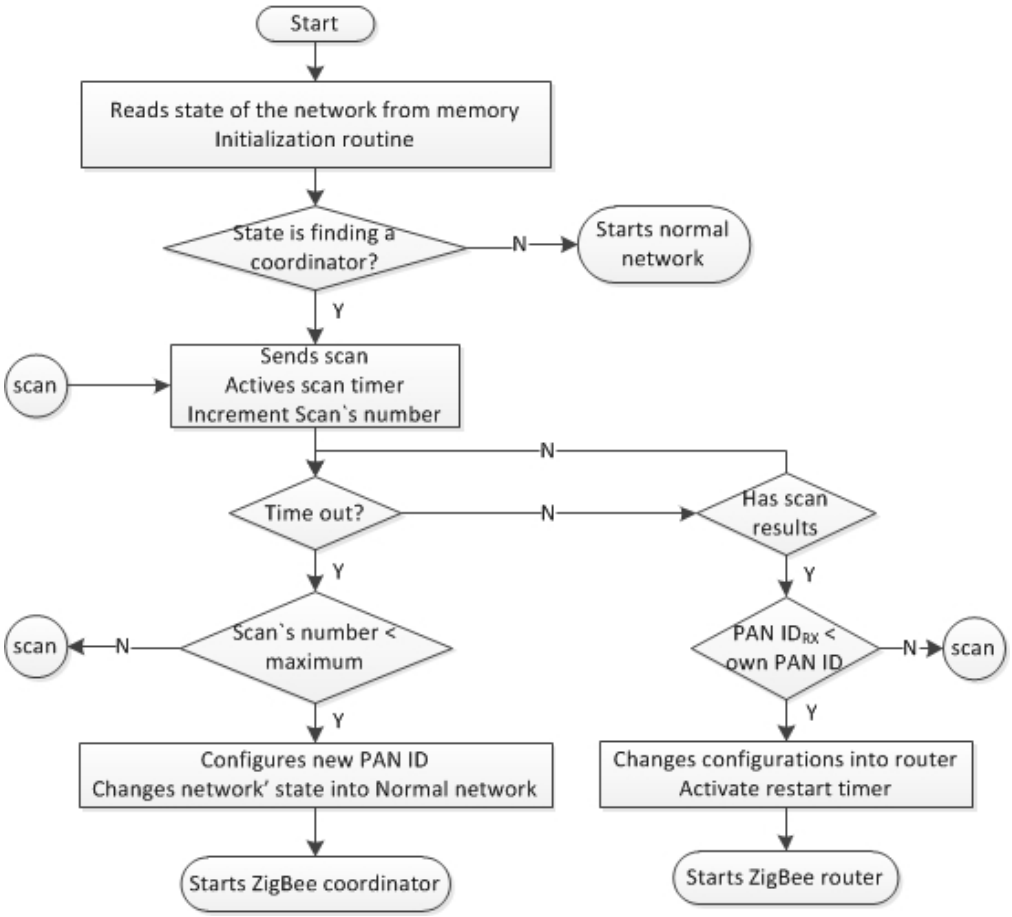

Figure 3.21: Coordinator selection flow chart

3.3.2 RSSI propagation algorithm

Once the coordinator has been selected, all devices restart with their respective device type and the ZigBee network is established. Then, the algorithms for the RSSI propagation start, one algorithm in the coordinator and a different in the routers. The information communicated using ZigBee is, the RSSI among all devices, the battery state, short address of each node and the number of devices in the network.

Coordinator

The coordinator is the device, which allows others to join the network. In this application, it is also in charge of the communication control for the RSSI propagation. Every time a device joins, the coordinator updates a list with the short

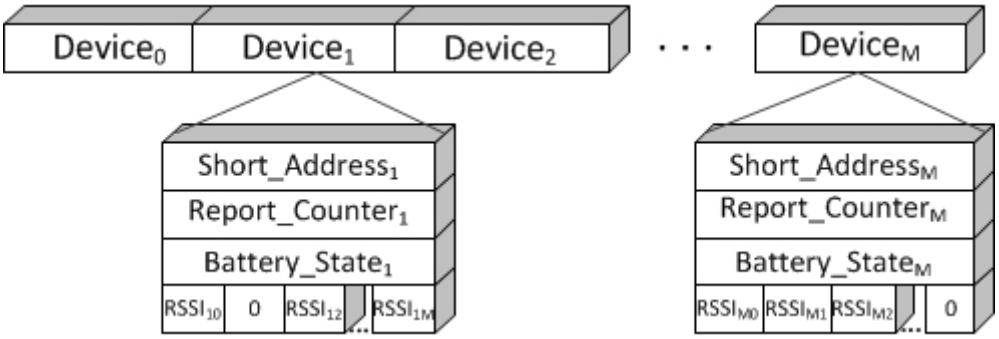

Figure 3.22: Data structure

address of the devices already associated. This list is useful for managing the data of all devices. That is, there is a vector with length equal to the maximum number of devices[8]. The vector is the type of a structure, which has the information collected in Table 3.6. The data structure of a system with M devices is represented in Figure 3.22. The $RSSI_{12}$ means the strength of the signal that $Device_1$ detected when a frame was received from $Device_2$. $RSSI_{12}$ and $RSSI_{21}$ may be different, due to effects explained in the Sections of *Path loss theory* 2.2.5 and *Path loss effects* 3.1.2, and the RSSI between a device and itself, for example $RSSI_{MM}$, is zero.

Data	Type
Short address	Unsigned 16Bit
Counter of reports	Unsigned 8Bit
Battery state	Unsigned 8Bit
RSSI vector	Unsigned 8Bit

Table 3.6: RSSI device structure

The position 0 of the vector is always the coordinator, the other positions are in accordance with the order of joining the network and is given by the coordinator. This order is the same for the *RSSI vector*. The length of this vector is equal to the maximum number of devices. Every time the coordinator polls a device, it decrements the device's *counter of report*, to control if the device is still in the network.

The coordinator polls each device with a unicast, $CMessage_n$. In this message, it sends its own information, such as battery state and the RSSI between itself and the

[8]This value is a constant that can be changed easily, in the current application it has a value of 16

routers. The time between the messages of the coordinator is 300ms. The format of a message addressed to $device_n$ is shown in Figure 3.23.

The strength of the signals between the coordinator and each device in the network, bytes 6,9,12, etc. where there are D devices associated, is a set defined as $RSSI_0 = [RSSI_{00}, RSSI_{01}, RSSI_{02}, ..., RSSI_{0D}]$. It means that the length of $CMessage_n$ is not constant and depends on the number of devices in the network.

Destination Short address: $device_n$	Number Associated devices	Coordinator Battery voltage	Coordinator Short address	$RSSI_{00}$	$Device_1$ Short address	$RSSI_{01}$	$Device_2$ Short address	$RSSI_{02}$
Bytes:0-1	2	3	4-5	6	7-8	9	10-11	12

Figure 3.23: *CMessage* frame format

The *CMessage* is the *frame payload* field of a ZigBee unicast message. The headers and other fields are built by the stack.

After sending the OTA[9] message, the coordinator decrements the device's *counter of report* and transmits serially, $SerialMessage_{n-1}$, the information of one of the devices in the network. For example, if there are two devices associated, the coordinator updates its battery level and sends at first its own data (*coordinator*), 300ms later the data from $router_1$, then $router_2$ and the cycle starts again. The Figure 3.7 illustrates a *SerialMessage* format with the information of $device_s$. The length of the *SerialMessage*, in contrast to the OTA frames, is constant and equal to $length = 6 + the\ maximum\ number\ of\ devices$. The RSSI of the devices that are not in the network is filled with 255.

As a response, the router which was polled sends its own information in a broadcast, $RMessage_p$. Each device including the coordinator saves this information in its data structure, see Figure 3.22. With this response, the coordinator increments the device's *counter of report*. The flow chart of the coordinator RSSI propagation algorithm is depicted in Figure 3.24.

Routers

As soon as a router restarts, it attempts to join the network. Then, it waits for the coordinator's message or a message from another router. If it receives the *CMessage* message, it updates the list of associated devices and the coordinator's

[9]Over The Air

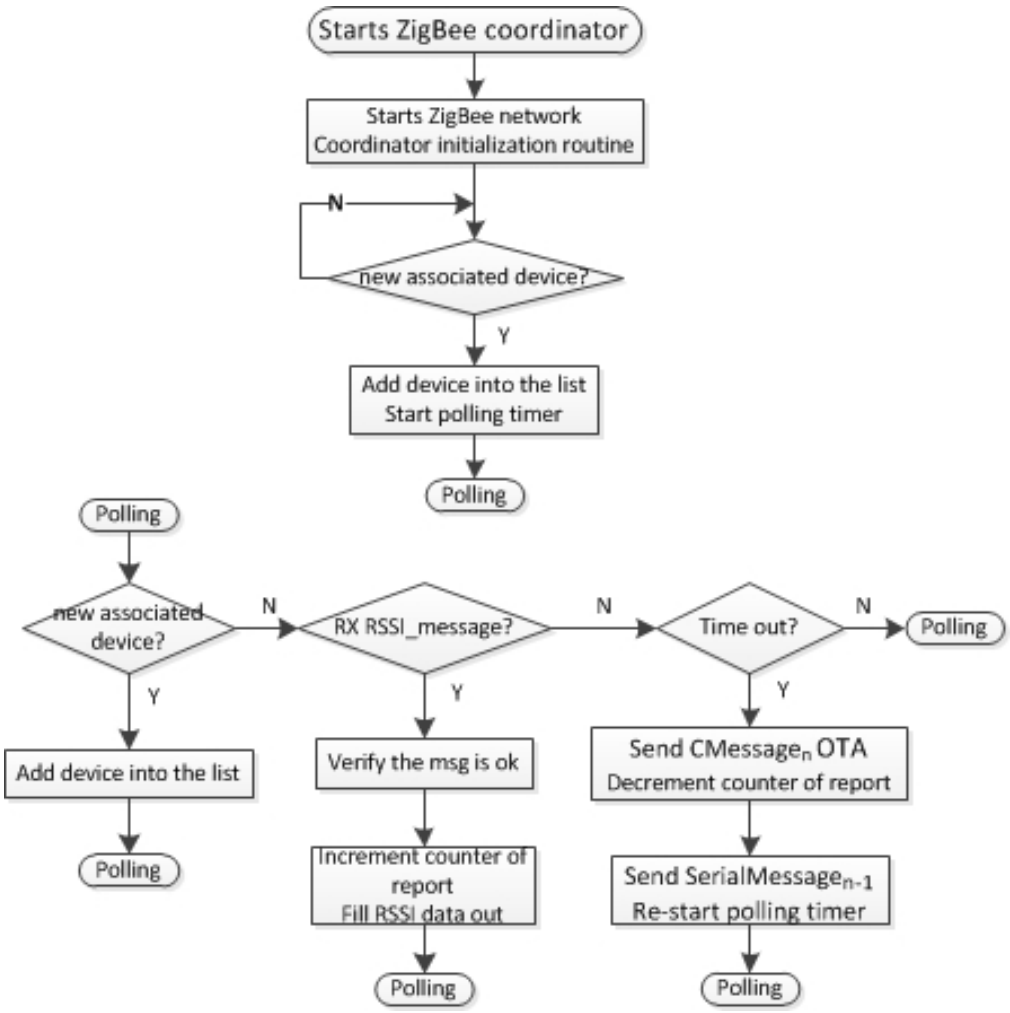

Figure 3.24: Coordinator RSSI propagation algorithm

information, in a vector of structures alike the coordinator's one, represented in Figure 3.22. If the message is from another router, the device updates this router data.

After the data actualization, the router reads its battery level and sends a broadcast with the frame structure of Figure 3.25. The short address of the coordinator, i.e. 0x0000, is the first field, and it is used to check the frame. If the device which sends the message OTA is the number r in a network with D devices associated, the set of strength of the signal is:

$$RSSI_r = [RSSI_{r0}, RSSI_{r1}, RSSI_{r2}, ..., RSSI_{rD}]$$

The frame's length is a function of D.

Coordinator Short address	Number Associated devices	$Device_r$ Battery voltage	Coordinator Short address	$RSSI_{r0}$	$Device_1$ Short address	$RSSI_{r1}$	$Device_2$ Short address	$RSSI_{r2}$
Bytes:0-1	2	3	4-5	6	7-8	9	10-11	12

Figure 3.25: Router frame

After sending the OTA message, the router transmits a serial message with the data from one associated device, like the coordinator does and with the same frame format, see 3.7. The Figure 3.26 illustrates the flow chart of the routers.

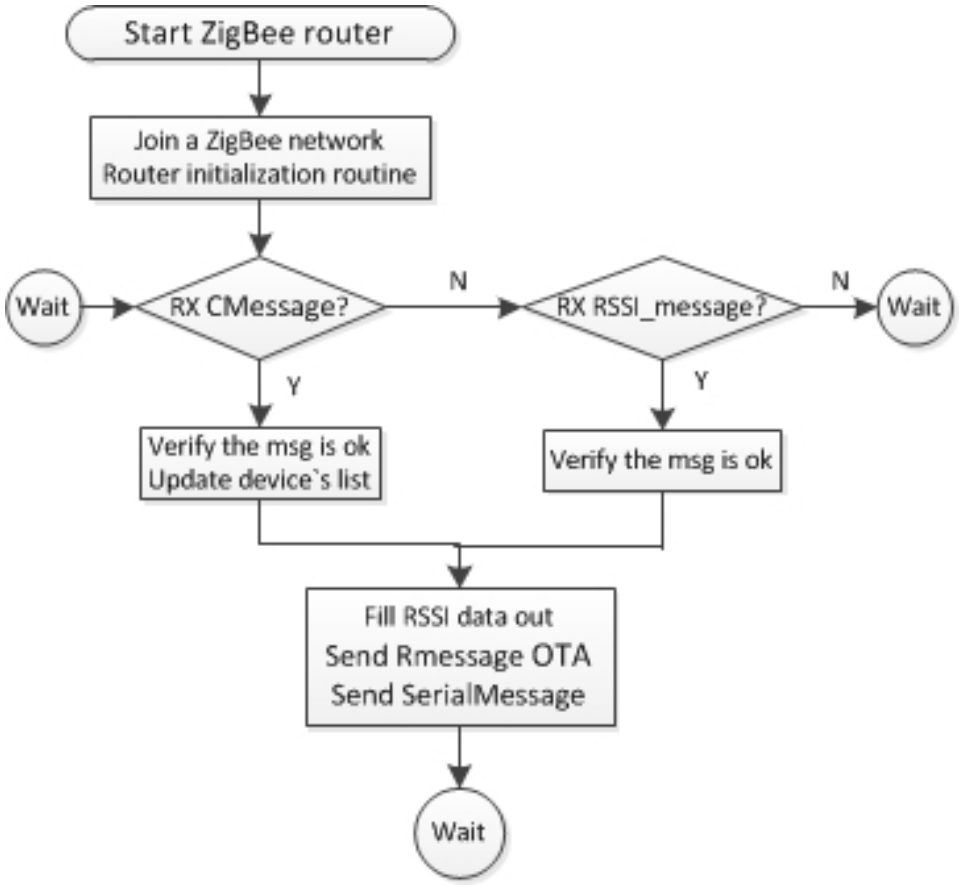

Figure 3.26: Router algorithm

Communication flow

The Figure 3.28 provides an illustration of the communication's flow and the interaction among the different devices. The system consists of the coordinator C, and D routers $\{R_1, R_2, ..., R_D\}$. The coordinator, R_2 and R_D are connected to a computer through their serial port. The messages from the *coordinator* are named $CMessage_n$ with n as the destination number's device. The routers' broadcasts are labeled RM_r where r is the number of the router source. And the serial messages (colored in green) are called $sm_{r,m}$ with r as the number of the router source and m the number of the device which information is contained in the message.

The cycle starts at *T=0* with the $CMessage_1$, then router R_1 sends the broadcast RM_1. In *T=300ms* the coordinator polls router R_2 and this replies with RM_2. The process continues until *T=300ms x (D-1)* when R_D is polled. Then the cycle starts again in *T=300ms x D*.

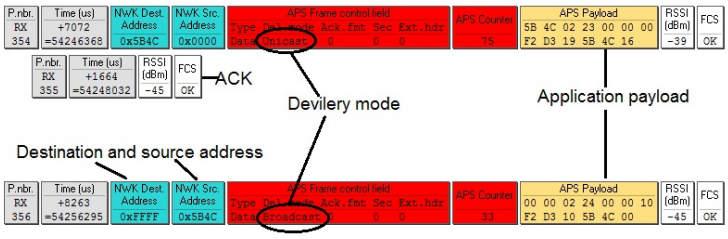

Figure 3.27: ZigBee frames

The coordinator unicast messages and the router broadcast messages are the application payload of a ZigBee frame. The Figure 3.27 shown three types of frame, as it is depicted by SmartRF Packet Sniffer. The first one is a coordinator unicast to the device with short address 0x5B4C (hexadecimal). Thus, the source address is 0x0000 (hexadecimal). The device 0x5B4C replies with an *acknowledgement* frame, second frame in the Figure, according to the ZigBee protocol standard.

And the last frame is the response of the router application to the coordinator message. This frame is a broadcast, i.e. the destination address is 0xFFFF. In this example, there are two associated devices, byte 3 in the application payload, the 0xF2D3 (bytes 8-9) and the 0x5B4C (bytes 11-12). Specifically in the last frame, the battery state of 0x5B4C is the byte 4^{10}, and the RSSI between this device and the coordinator is byte 7, and between the other router is byte 10.

[10]The value is multiplied by 10 and transmitted, in this case is 0x24 hexadecimal or 36 decimal. i.e. 3.6V

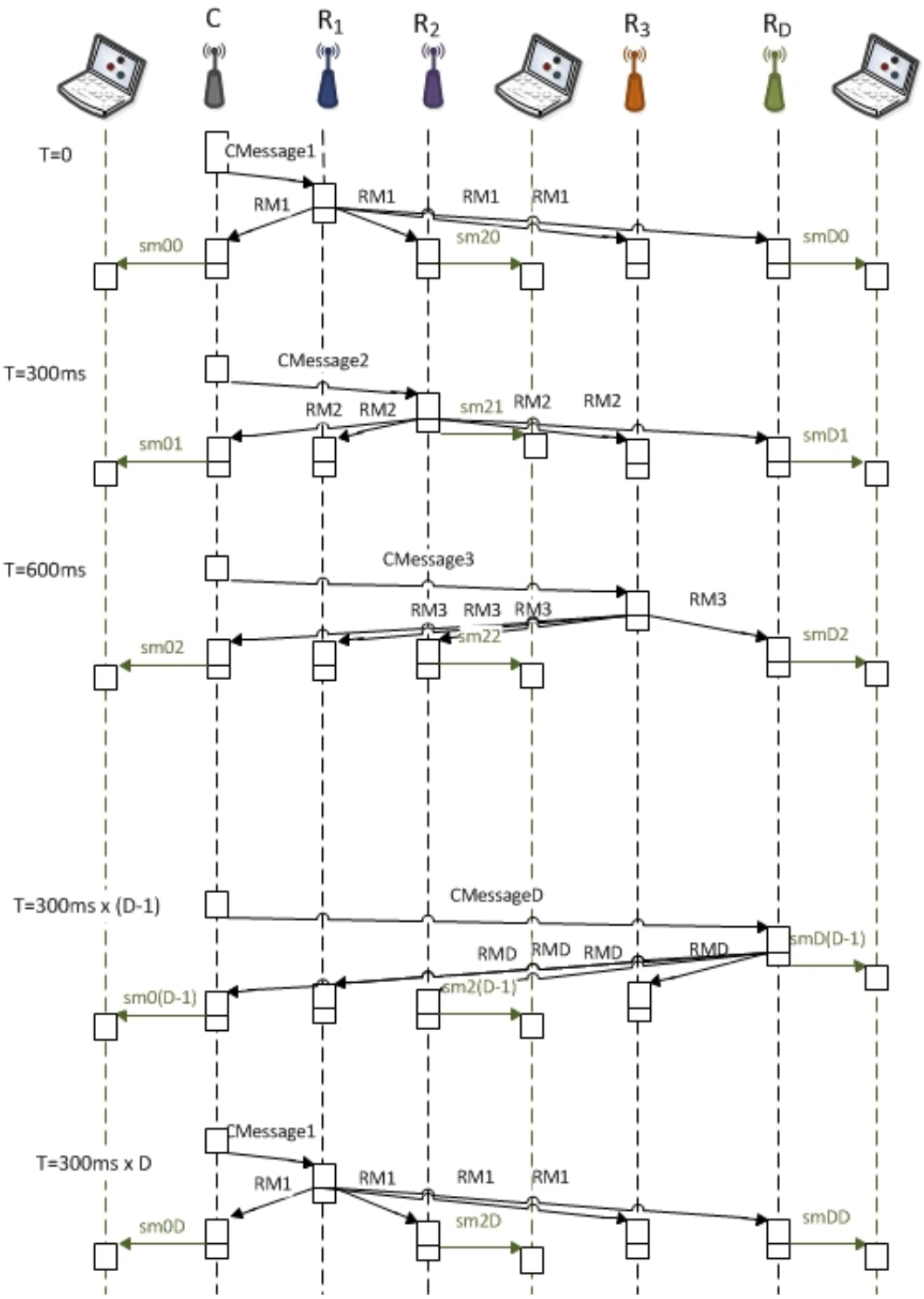

Figure 3.28: Data flow and interaction among devices

4 Results

Several experiments were executed to assess the performance of the developed system. These experiments are grouped in two main clusters. The first concerns to the networking localization solution, shown in Section 4.1, and and the second relates to the dynamic coordinator selection, shown in Section 4.2. This Chapter describes in detail all the experiments performed to the system, and the analysis of the gathered results.

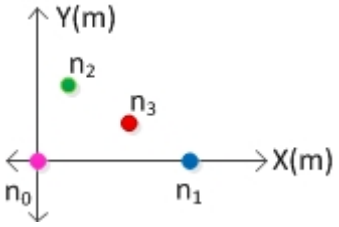

Figure 4.1: Basic configuration

The basic system's configuration utilized in the experiments used to test the localization system is depicted in Figure 4.1. The node n_0, (colored in magenta), is the node connected to the computer and therefore the reference with coordinates $(X, Y) = (0, 0)$. The device n_1, in blue, has always its Y coordinate equal to zero and its coordinate X is positive. The node 2 n_2, which colored in green, is located in the positive side of the Y Cartesian axis, and the third, n_3, in red, may be located in any place. In all further experiments, the nodes are represented using the same color convention.

Six different positional cases were considered:

1. In the first one, described in Section 4.1.1, the devices 0, 1 and 2 are in a constant position while the node 3 moves along the plane.

2. In the case shown in Section 4.1.2, nodes 0, 1 and 3 are fixed in the same position described in the previous case, while n_2 moves through the Y positive plane.

3. In this case the node n_1 changes its position while the others remain in the aforementioned locations. This case is described in Section 4.1.3

4. In the case, expounded in Section 4.1.4, all modules are in a constant position but node n_1 rotates around its axle.

5. This case is similar to case 4, though node 2 rotates while the others have a constant position. See Section 4.1.5

6. Case six evaluates the system's collinearity (explained in Section 4.1.6): All modules are located in different points in the line y=0.

To evaluate the coordinator selection three experiments were performed:

1. This case, shown in Section 4.2.1, considers the situation when the devices are started up in the same order as it is their priority, that is, the device with highest priority is the first one to be initiated.

2. In the case described in Section 4.2.2, a device with high priority starts up when a network is already conformed by a device with lower priority.

3. Finally, Section 4.2.3 presents the situation when the modules are turned on in a random order and before any device completed its coordinator selection process.

4.1 Localization System: tests and results

The first step performed by the localization system is a system's calibration described in Section 3.2.3. In this step, the devices are located according to Figure 4.2:

As observed from table 4.1, the highest correction factor that was obtained had a value of 16.1 dB for d_{03}. That translates in a difference of 855.6cm between the real and the expected, or calculated distance[1]. This difference is due to the RSSI's dependence on the characteristics of the environment, losses caused by the RF component connections and reflections, see Section 3.1.2, the antenna's radiation pattern, and the antenna's position. In addition, since to the relation between the RSSI and the distance is exponential, small changes in the value of the RSSI signal represent large increments in the distance, as it was demonstrated in Section 3.1.2.

[1]The expected values are taken from the equations described in Section 3.2.1

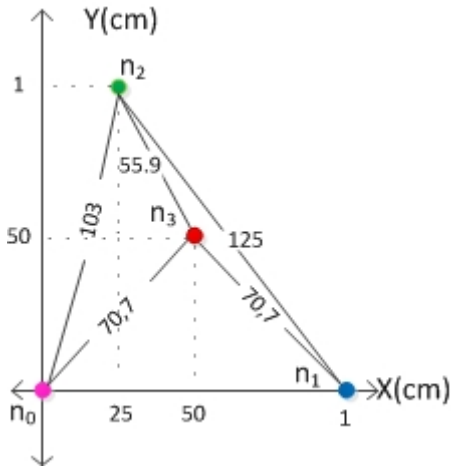

Figure 4.2: Positional calibration

	real distance (cm)	estimated distance (cm)	RSSI expected	correction factor (dB)
d_{01}	100	241.9	33.5	10.6
d_{02}	103	958.6	33.7	20
d_{03}	70.7	351.44	30.9	16.1
d_{12}	125	222.18	35.18	7.7
d_{12}	70.7	266.7	30.9	14.6
d_{23}	55.9	259.36	29.2	15.9

Table 4.1: Positions and correction factor

The lowest correction factor corresponds to d_{12}. This factor (7.7dB) represents a difference of nearly 100cm between the distances. The real and estimated positions for each node in this experiment and their error obtained after the calibration process are presented in table 4.2. It is evident not just the usefulness but the necessity as well, of the process of calibration in order to reduce high errors and to make the system more stable.

	X_1	X_2	Y_2	X_3	Y_3
Mean estimated value (cm)	104.93	52.13	97.98	56.3	42.31
Real value (cm)	100	25	100	50	50
Real error (cm)	4.93	27.13	2.02	6.31	7.69
Percentage error	4.93	108.52	2.02	12.62	15.38

Table 4.2: Post-calibration errors

The post-calibration position of the nodes n_1 and n_3 is constant when is compared with the position of the node n_2. Due to the location of node 2 and that the calculation of its coordinates depend on the distance measured to device 0 and 1; it is not simple to find an antenna position that covers both nodes. After the calibration and without moving n_2, the mean of the X_2 error between the real and the calculated distance is 27.13cm, or 108.52%. These results show the high dependency on the angle between the node and n_0 and n_1. This phenomenon is more evident in cases 2 and 5 described in Sections 4.1.2 and 4.1.5.

The measurements of the node's positions were performed after the system was calibrated. The description of these processes and their results are summarized in the following sections.

4.1.1 Localization case 1

The first set of experiments correspond to case where nodes n_0, n_1 and n_2 are located in a constant position, that is $(X_0, Y_0) = (0, 0)$, $(X_1, Y_1) = (100, 0)$ and $(X_2, Y_2) = (25, 100)$; while n_3 moves along the plane. Figure 4.3 shows the error of the localization of node n_3 in the plane. The areas where the error was lower than 50cm are shown in dark blue, the area filled in cyan has an error of approximately 100cm; the area in yellow has an error of about 150cm and the area in red, errors of more than 200cm.

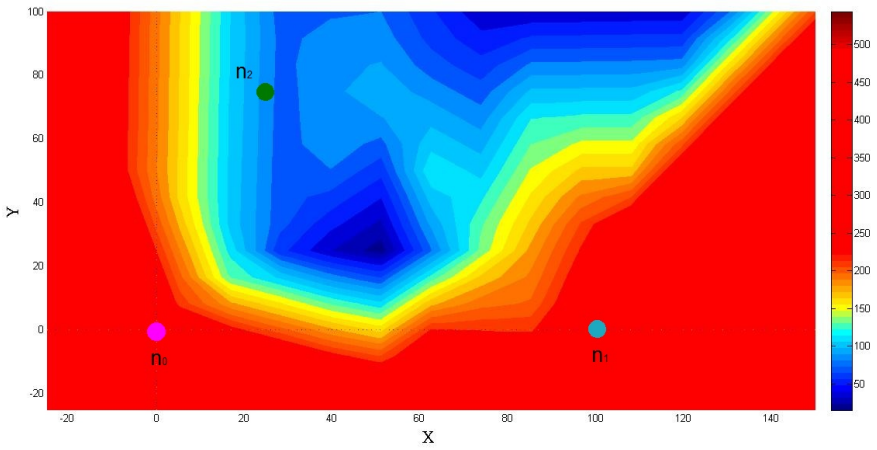

Figure 4.3: Localization error configuration 1

Figure 4.4 the aforementioned error expressed in percentage. The areas shown in dark blue are those where distance found to node n_3 has errors lower than 50%, the

light blue and green areas are the zones with errors between 100% and 175%, the yellow zone shows errors of 200% and the red zone has errors bigger than 250%.

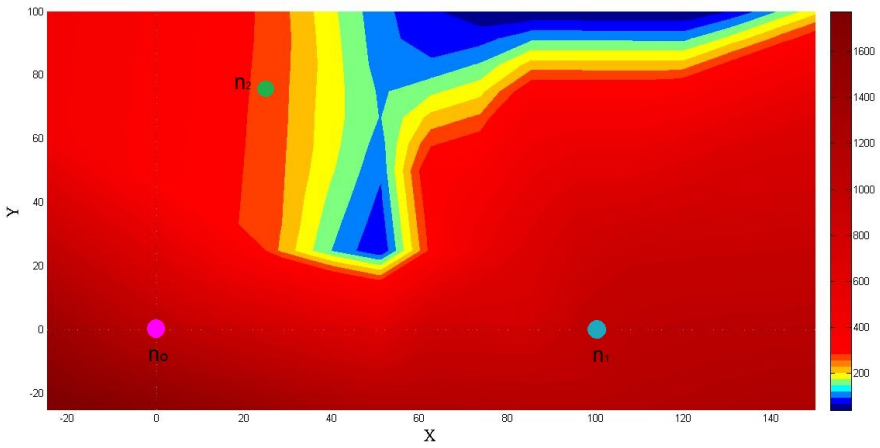

Figure 4.4: Percentage error for device 3

It can be concluded that when the localization system requires good precision, the node n_3 should be located in, or cannot be outside of the blue areas shown in Figure 4.4. In these areas, the calculated position has an error of maximum 50%, and one of the radio frequency signals between n_3 and each of the other nodes has a good coverage (for example, $RSSI_{13}$ or $RSSI_{31}$ should be have a strong signal, or in other words, one node should be in the line of sight of the other device's antenna).

If it is acceptable for a system to have an approximate localization of the nodes, the range of the system's coverage can be extended to the cases where the node n_3 is in the yellow and blue areas shown in Figure 4.3. In these zones, the devices may not be in the line of sight of the others but they are still being covered by the spread of the communications signal.

4.1.2 Localization case 2

The situation presented in this case, is when nodes 0, 1 and 3 stay in a fixed position while node 2 moves across the positive portion of plane Y. The real coordinates of node 1 and 2 are $(X_1, Y_1) = (0, 100)$ and $(X_3, Y_3) = (25, 25)$ or $(X_3, Y_3) = (50, 50)$. Two different results are considered, one is the effect of the movement of n_2 on the localization of node 2 and the other is the variations in the localization of node 3.

The reason of the later is because the position of node 3 depends on the localization of node 1 and 2, and the distance among all the devices as described in Section 3.2.4.

Effects on node 2 localization

Figure 4.5 provides an illustration of node 2's localization error. Their color convention is shown in Table 4.3

Color	real error range (cm)	percentage error range %
Dark blue	0-50	0-40
Cyan	50-100	40-130
Yellow	100-150	130-170
Red	150-200	170-200
Dark red	bigger than 200	bigger than 200

Table 4.3: Color-map error ranges

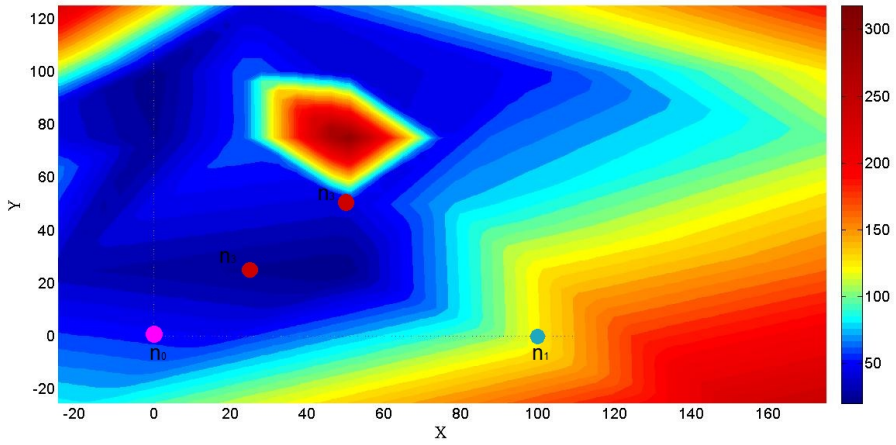

Figure 4.5: Case 2. Localization error for device 2

It was found that there is a red zone between $X = [30 - 75]$ and $Y = [60 - 90]$. It is believed that this zone is due to the antenna's position and the signal coverage between node 2, and node 0 and 1. As it was described before, it is not feasible than in this zone three devices are in the line of sight of their antennas or at least, having a good coverage. When n_2 moves away, the precision of the localization improves because the coverage zone is wide at longer distances. The position of n_2

near to this red zone, during the calibration process, explains the difference of the estimated distance after the calibration without moving the node.

It was found that the signal strength of the RSSI on node 2 increases with the distance. This effect is expected due to the direct relationship between this indicator and the distance. In addition, this variation produces an increment in the localization error due to the fact that the distance is an exponential function of the RSSI. Therefore, as the distance increases, smaller changes in the RSSI cause larger changes in the calculated distance and thus bigger errors. See Sections 2.2.5, 3.1.2 and 3.2.2 for an explanation of Path Loss Theory and a description of Distance Estimation.

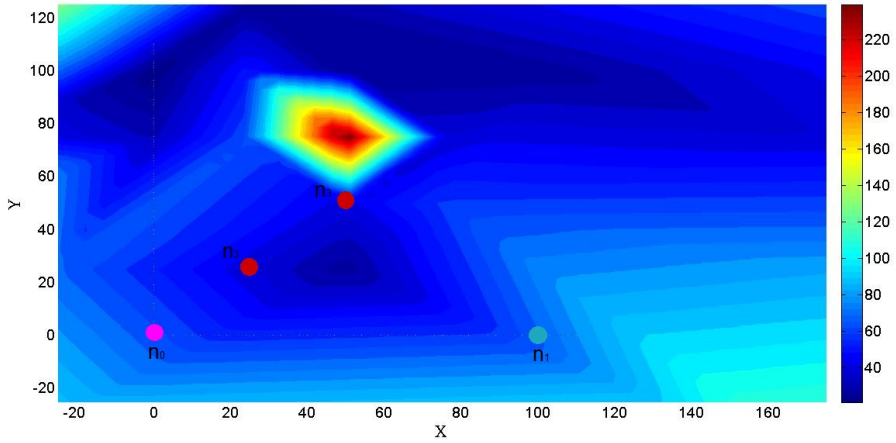

Figure 4.6: Case 2. Percentage errors for device 2

The error between the real and the measure positions is shown in Figure 4.6 and its conventions further explained in table 4.3. As it can be observed, the regions with errors lower than 60% are broader than those in Figure 4.5. This shows that in this case the system exhibit large areas with small errors, and that the localization of node n_2 from the zones shown in cyan until the zones shown in red is not reliable.

Comparing Figure 4.6 and Figure 4.4, it is found that is evident that the behavior of the localization system for node 2, when it moves, is more stable than the one obtained when node 3 moves. This can be explained due to the fact that the localization of node 2 depend just on the three distances calculated to n_0, n_1 and n_2 while the n_3's location depend on these distance and three more, computed between it

and the other nodes. This fact makes that errors in the signal measurements of n_0, n_1 and n_2 will be reflected in the n_3's location estimation, incrementing its error.

Variations in node 3's localization

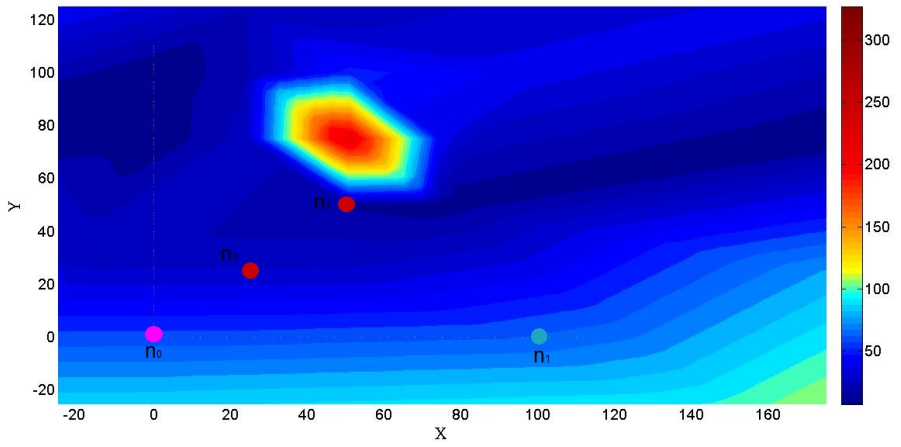

Figure 4.7: Case 2. Localization error for device 3

The incremental error of n_3 is better observed in Figures 4.7 and 4.8. For this configuration, node 3 is located in a fixed position, prone to the errors caused by the movement of node 2. The high errors in the localization of n_2, shown in Figure 4.5, are directly reflected in Figure 4.7, where it can be seen that they cause errors higher than 80cm in n_3's positioning.

According to equation (2.3), the Y_3 coordinate depends directly on the distance between nodes 2 and 3, and the coordinates X_2 and Y_2. The influence of node 2 localization's error is clearly evident in Figure 4.8, where there is a wide red zone with errors of more than 200% caused by node 2's localization errors.

It can be concluded than for an optimization of the localization of node 3, this red zone should be considered, and the its positioning should be made independent of node 2, or the system should be directed to select another node to be n_2 if there are more available nodes in the system.

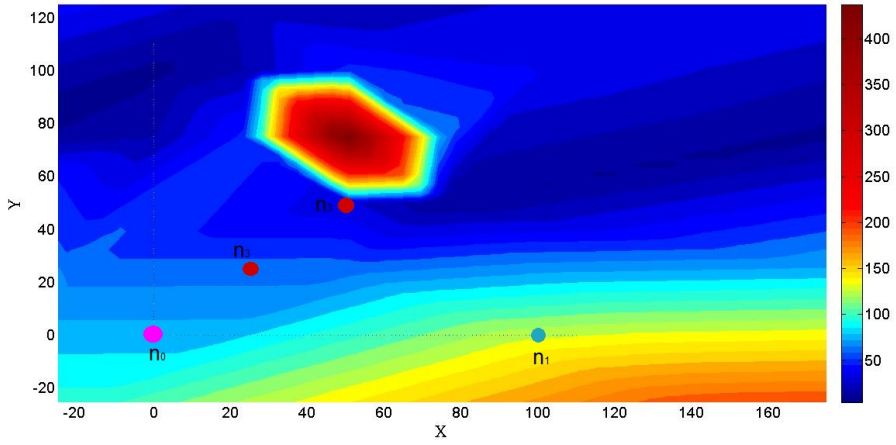

Figure 4.8: Case 2. Percentage error for device 3

4.1.3 Localization case 3

In the case described in this section, n_2 and n_3 have a constant position, $X_2, Y_2 = (25, 75)$ and $X_3, Y_3 = (50, 50)$; and n_1 moves along axis X. Once again, the increment on n_1's errors are reflected in n_2 and n_3 errors.

X_1	n_1 %error	n_2 error	n_3 error
125	5.1	24.9	8.04
100	19.8	19.36	11.48
75	19.8	21.95	16.81
50	21.1	62.23	19.34
25	37.1	121.2	50.82

Table 4.4: Case 3. Node 2 and node 3 errors

From table 4.4 can be concluded that the best results of the positioning system for n_1 are obtained when it is far from n_0, instead of when it is close to it as it was estimated at the beginning of this project. This is because the range coverage of the transition signal strength between n_1 and the other nodes is wider, but also because n_1 is located near to its calibration point. It means that in distances between n_1 and n_0 longer than 125cm the n_1's error increases and therefore the localization of the whole system will be more imprecise. For these reasons, the position of node 1 in the calibration process should be in a middle point between the reference node and the maximum length allowed in the X axis of the plane.

The Figure 4.9 shows the effects of node 1's movement on the localization of n_1, n_2 and n_3. In this color-map figure, the regions with errors lower than 40% are shaded in blue, those in the ranges of 40% and 50% are in cyan, the areas with errors between 60% and 90% are shaded in yellow and those higher than 90% in red. The node n3 has a good response to n_1 movements. This is due to the fact that n_3 is located in the blue zone where the errors are still low. Nevertheless, the location of n_1 has a higher impact in n_2 localization precision than in n_3's errors, caused for signal changes associated to its antenna position and location, as it was mentioned before.

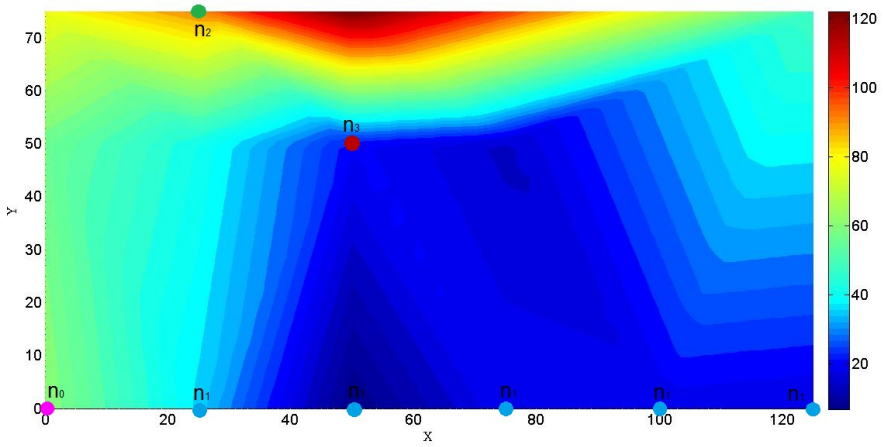

Figure 4.9: Case 3. Errors in the localization system

4.1.4 Localization case 4

Figure 4.10 represents the case 4, where all modules are in a constant position, $(X_1, Y_1) = (0, 100)$ $(X_2, Y_2) = (25, 75)$ and $(X_3, Y_3) = (50, 50)$; but n_1 rotates around its axle.

Table 4.5 collects the best *(E3)* and the worst *(E6)* cases, for the localization error in all nodes in this set of experiments. The best case is found when the n_1's antenna is parallel to axis X, and therefore its radiation patter covers the other devices. In this case there is not a problem with node 0, which antenna is perpendicular to node 1, because n_1 is in the line of sight of n_0's antenna and thus, the implemented *weighting function* compensates the discrepancies between $RSSI_{01}$ and $RSSI_{10}$, as it was explained in Section 3.2.1.

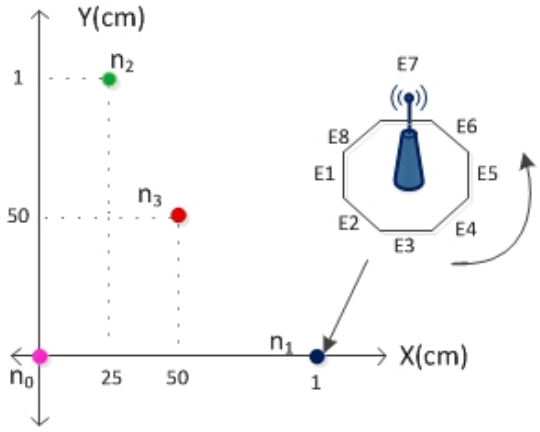

Figure 4.10: Configuration case 4

The worst case is found with experiment *E6*. In this case, n_1 and n_2 antennas are parallel and therefore, there is not a good RSSI than can be compensated. This effect is reflected by a 285% localization error obtained in the X_2 coordinate.

case	X_1 % error	X_2 % error	Y_2 % error	X_3 % error	Y_3 % error
E3	21.6	13.8	13.4	12.4	16.2
E6	26.86	284.6	16.6	19.6	14.9

Table 4.5: Case 4. Node's localization errors in the best and worst cases

From this case, it can be concluded that the best position for n_1's antenna is parallel to axis X, when n_0's antenna is perpendicular to the same axis.

4.1.5 Localization case 5

This set of experiments is devoted to find the effects on the localization system's error when node 2 rotated around its axle, while the other nodes remained fixed in their positions, as it is shown in Figure 4.11. TThe coordinates of the nodes are the same used in case 4, that is $(X_1, Y_1) = (0, 100)$, $(X_2, Y_2) = (25, 75)$ and $(X_3, Y_3) = (50, 50)$.

As it was expected, the best position for n_2's antenna is *E1*, where there is a better coverage of the other devices. In contrast the worst case is experiment *E3*, where the antenna of node 2 is parallel to axis Y and no device is in its line of sight. The localization error of cases *E1* and *E3* is summarized in Table 4.6.

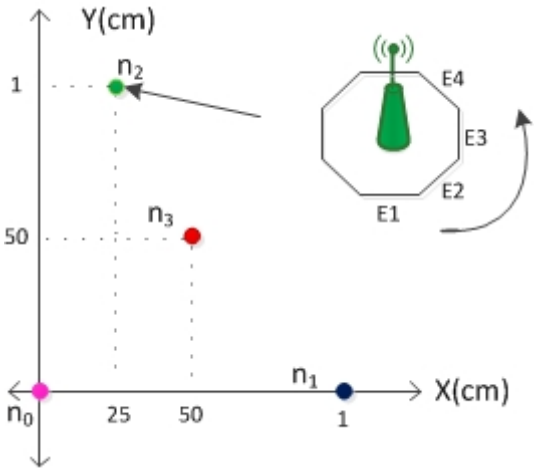

Figure 4.11: Configuration case 5

case	X_2% error	Y_2 % error	X_3 % error	Y_3 % error
E1	201.9	3.7	26.5	23.9
E3	493.3	23.3	14.5	54.6

Table 4.6: Case 5. Node's localization errors in the best and worst cases

This experiment demonstrates not just the importance of the location of node 2 but also the importance of the positioning of its antenna. It was found that even when n_2 is located in the best possible position, case *E1* shows an error of nearly 200% due to small angle changes of its antenna. This experiment highlights that the localization system should not depend on the position of the antennas, and therefore, a fixed antenna, that cannot swivel and that has a homogeneous pattern in the plane *(X,Y)* it is required for achieving a more precise localization. A rough estimated pattern of the antenna used in this localization implementation in the plane *(X,Y)* is shown in Figure 4.12. This pattern is a result from the data of cases 5 and 6.

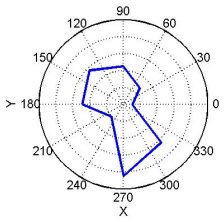

Figure 4.12: Estimated antenna pattern

4.1.6 Localization case 6

This case attempts to assess the behavior of the implementation when collinearity is present. Specifically, 3 cases are taken into account; they are shown in Figure 4.13. For these experiments, all antennas are perpendicular to axis X.

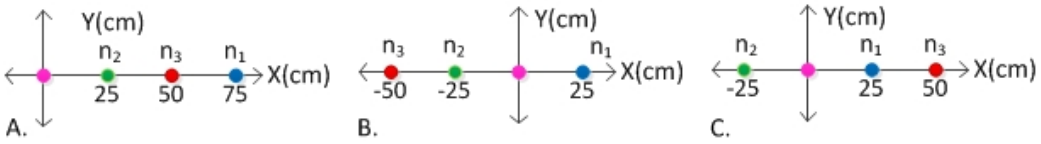

Figure 4.13: Case 6

In this case, the worst RSSI is found to be between devices which distance is longest, See Table 4.7, I.e. in case A node 1 has been located in worst position (75cm from n_0), but its effects are not reflected in its location because n_1 has Y_1 always in zero and X_1 depends only on the strength of the signal between it and n_0, thus there is not accumulation of errors.

Case	measurement	X_1 (cm)	X_2 (cm)	Y_2 (cm)	X_3 (cm)	Y_3 (cm)
A	real position	75	25	0	50	0
	estimated position	42.6	18.3	12.1	28.9	-4.6
B	real position	25	-25	0	-50	0
	estimated position	13.3	-380.3	382	-206.9	174.2
C	real position	25	-25	0	50	0
	estimated position	11.8	-8.8	3.9	29.3	-1070.3

Table 4.7: Case 6. Node's errors in the best and worst cases

Case B has a particular configuration. Node n_2 has the biggest localization errors because the RSSI between it and n_2 is higher compared with $RSSI_{01}$ and $RSSI_{02}$, This causes the coordinate X_2 to be negative and large as it was explained in see Section 3.2.4. The localization error of node 2 and node 3 drastically increment the localization errors in the final position of n_3.

Finally in case C, the worst RSSI is between n_2-n_3 is shown. According to equations (2.3), Y_3 depends on the distance between these 2 devices and when the distance d_{03}, X_2 and Y_2 are small compare with d_{23}, this distance will determine the value of Y_3, which is the situation presented in this case.

For solving the collinearity problem, it is suggested the implementation of a function which detects this situation based on the results shown by the described set of experiments, and performs the correction. This problem and a possible solution are mentioned in [MF04].

4.2 Coordinator selection measurements and results

The system that was used for these testes consisted of 4 modules. The configuration of each module is summarized in table 4.8. For instance, device d_1 has the highest priority and therefore, the lower PAN ID (in hexadecimal format), a lower number of scans[2] for checking (if there are other coordinators) and the lower time between each scan.

Device	Priority	PAN ID (Hex)	Scan's number	Scan time out (s)
d_1	1	0x41	5	2.02
d_2	2	0x42	6	3.04
d_3	3	0x43	7	4.07
d_4	4	0x44	8	5.1

Table 4.8: ZigBee modules configuration

The experiments shown in this section, were performed using the USB dongle with the *SmartRF^{TM} Packet Sniffer* described in Section 3.3.2. Figure 4.14 shows a snapshoot of the Sniffer's interface. This application consists of 2 parts, the first one displays the ZigBee OTA frames and the second shows different options useful for detailed information. For these experiments in particular, it was used the "time line" feature, which shows the received frames in chronological order, and classifies them using colors identifying each type of frame, and using lines identifying the source device.

The results of the 3 different cases are presented as it follows.

4.2.1 Case 1

In this case, the initiation sequence of the devices is the same order as the priority of the devices. That is, first d_1, then d_2, later d_3 and finally d_4. In order to compare the different times between scans, Table 4.9 collects them when each node is initialized in a situation as they are the only device in the network.

[2]In some documents the scans are referred as beacons, but since this functionality is the same, both names can be used

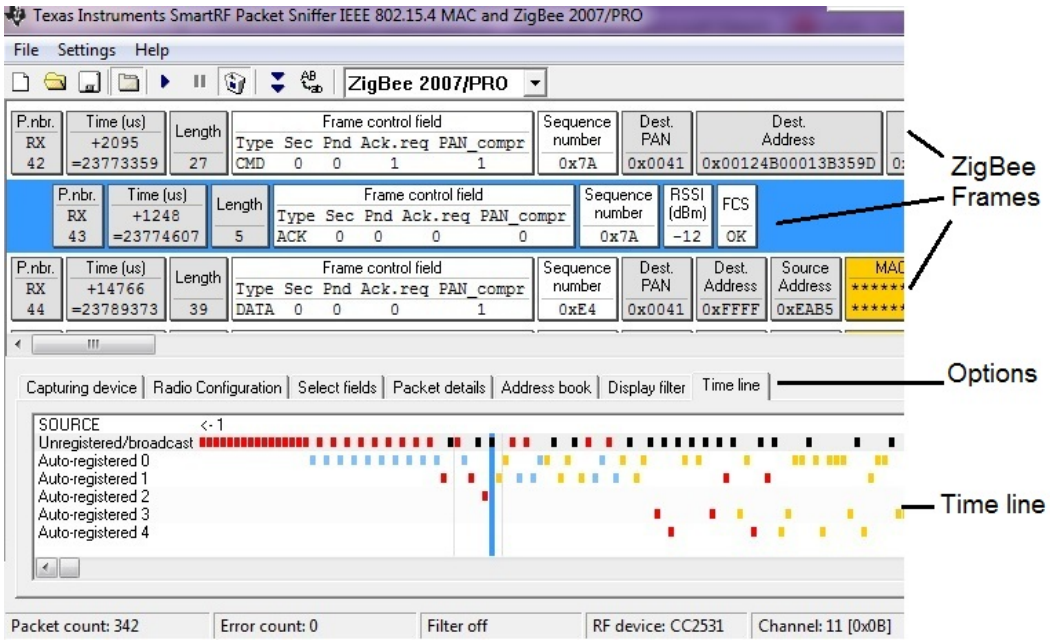

Figure 4.14: Packet Sniffer

d_1	d_2	d_3	d_4
12.5s	20.9s	31.2s	43.4s

Table 4.9: Devices starting time

When all devices are started up in the same order as its priority, and before the device with highest priority finishes its coordinator selection process, the time required for the ZigBee network establishment increases. This is due to time increments caused by longer exchanges of information. The total time that d_1 takes to complete its initialization under these circumstances is 14.6s.

In this case, the coordinator is always device number 1, which is expected due to its high priority, leading to a successful coordinator selection. An example of this' sequence is shown in Figure 4.15. It shows the network's establishment and the devices joining the network as routers.

4.2.2 Case 2

This case shows the situation when any device starts a ZigBee network and later another one, with a higher priority is initiated. The results of the experiments in this case are always successful demonstrating that the devices with priority higher

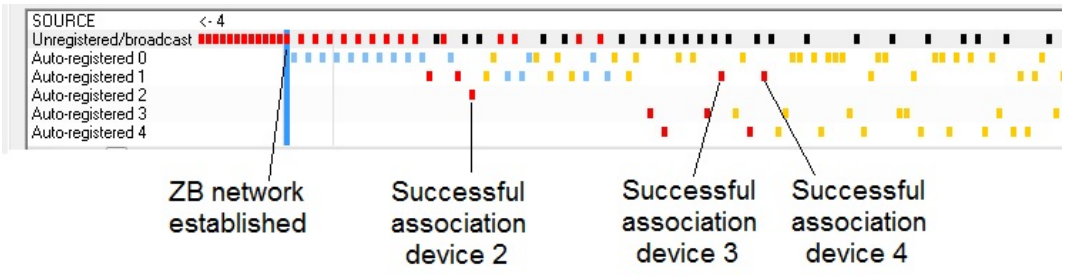

Figure 4.15: Coordination selection case 1

than the one that the current controller has, can successfully join the network and become routers.

Figure 4.16 shows the case when device 4 establishes the network and later device 1 joins it. When d_1 is powered on, it starts the coordinator selection process, which has duration of 12.5s. Then it performs the association request that is successful after 0.5s. This time may vary depending on the tasks that the coordinators is executing, such as performing scans, answering other association requests, localization application, or others.

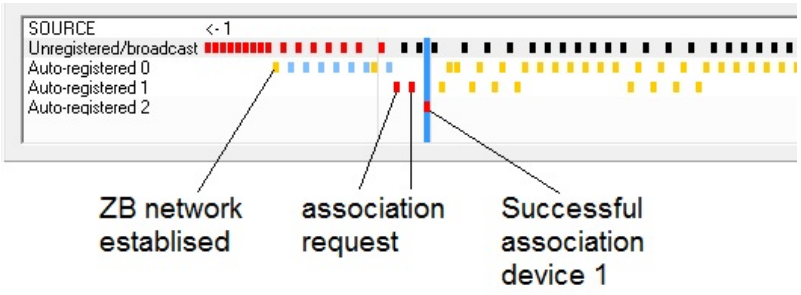

Figure 4.16: Coordination selection case 2

4.2.3 Case 3

In the experiments that compose this case, all modules are powered in a random order before any device completes its coordinator selection process. Unfortunately, the sniffer application does not show the source of the scans or its PAN ID. This causes that debugging the application changes the beacon's sending times and response times, causing delays and making a real time monitoring of the situation not feasible.

There are some situations, where 2 devices with different priority finished the selection process nearly simultaneously. It causes the establishing of 2 different ZigBee networks. Although rare, this phenomenon is produced when a device with high priority starts up after a device with lower priority .

Figure 4.2.2 shows the case when device 1 establishes the network while device 2 is restarting, in order to become coordinator and to create a ZigBee network.

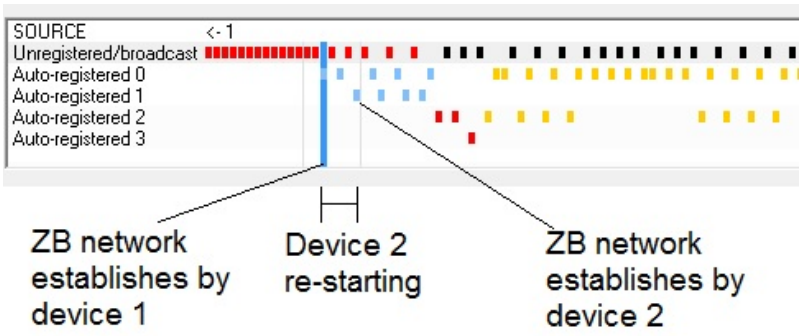

Figure 4.17: Coordination selection case 3

Due to the random nature of this event, and because it depends on the moment each device is turned on, it is suggested that to avoid this problem, the devices should be turned on in the same sequence as in cases 1 or 2 described above. Also two other solutions are described in Section 5.2. The basic idea behind these solutions is to perform scans after the network is established, or to use a different range of PANIDs between the selection process and the network conformation. This allows the devices to recognize a coordinator that already executed the selection process, or to recognize a coordinator, which already started a network.

5 Conclusions and Future Work

5.1 Conclusions

In a collective symbiotic system compounded of multiple stand-alone robots, which are autonomous, self-adaptive, self-assembling, highly capable and sensor-rich, it is required a low cost, low power and non complex localization system. Furthermore, a robust, self-healing and ad hoc wireless communication system is needed among the robots that are part of this symbiotic system. As a result, this master thesis proposed the design and development of a localization system using ZigBee (2.4GHz) not only as a resource to obtain signals used in the localization estimation, but also as a base for the communications system.

Along the realization of this project, an ample theoretical foundation was constructed, providing the necessary bases for the development of the localization and communications systems.

The localization system developed in this thesis is a low cost, low power consumption, distributed, anchor-free, self-configuring, cooperative and concurrent application based on each node's RSSI. Due to several factors, which influence the strength of the communications signal and due to its exponential relationship with the distance, it was implemented a calibration process, FIR and IIR filters in order to reduce the errors produced by these variations. Moreover, the developed algorithm has an implicit refinement process, which is a merge of the iterative multilateration, Heuristic and Weighted Least Square approaches.

Before the implementation of the calibration, filtering and refinement processes, the localization had and error higher than 1m, at any distance in a range of 50cm - 3m. That translates in errors larger than 100%. With the final implementation, it was possible to have localization errors lower than 5% or 10cm. Nevertheless, the results obtained from the localization system present numerous fluctuations produced by several factors, such as changes in the environment, characteristics of the RF component connections, antenna, interferences, among other factors, see Sections 2.2.5 and 3.1.2.

It is concluded from these results, that the calibration process is necessary in order to reduce errors caused for changes in the environment, radio transceiver and antennas. It is suggested the modification of n_2's calibration position to one with a X_2 position farer to the axis Y. Or test different location, for example, X_2 near to X_1 or X_2,Y_2 in the maximum length allowed in the X and Y axes of the plane. The purpose of this change[1] is to stabilize its localization behavior, and therefore of node 3, and to eliminate or reduce areas with high localization errors as shown in the experiments summarized in case 2 of Section 4.1.4.

It was found that the position of node n_1 should be between node n_0 and the maximum length allowed in the X axis of the plane. The experiments in the third case for the localization system described in Section 4.1.3, demonstrated that when n_1 is too close to n_0, nodes 2 and 3 have an increment in their errors; and when node n_1 it is very far from them, the exponential relationship between the RSSI and the distance also induces rises in the localization errors.

The performance and position of the antennas play an important role in the precision of the localization system. As a result, the system should be in such as configuration that at least one of the antennas of nodes 0, 1 and 2 are in the line of sight and node 3, between the region of the coverage of these 3 nodes, or where node 3 can cover them. These results can vary with the use of different antennas.

As for the selection of a network coordinator, when the device with highest priority is powered on before the other devices, or devices with higher priority than the coordinator are started up after the network is established; the application works with a 100% of successful cases. However, if the devices are powered on in a random order and before any device had finished its selection process, it could happen that two devices become the coordinator and therefore, it will cause the creation of two different ZigBee networks. To avoid this situation it is advised to follow the procedure for powering on the devices described in case 1 and 2 of Section 4.2.

5.2 Future work

Different ideas emerged during the development of this thesis, and unfortunately by time constrains did not allow to implement them. This section collects these ideas.

The results gathered in this work show that the performance of the developed localization system deeply depends on the position of nodes n_0, n_1 and n_3. Thus, it

[1] This change also requires a modification in the application developed in Matlab, see Section 3.2.3

is proposed the implementation of these nodes in docking stations[2], and as a result, to reduce the errors produced by their change of position. With this configuration, it is possible to implement an algorithm, based on the implementation of this thesis, which can compute a global map of the positions of the nodes that compose the system.

For a better performance of the system, it is proposed a change of antenna. As it was expounded before, any alteration in the antenna position, in angle or alignment with respect to axis Y represents different distance estimations and therefore different positioning. Thus, it is recommended replace the current antenna with a non-swivel one and with an uniform radiation pattern in axes *(X,Y)* (i.e. a circular radiation pattern). If the antenna is changed, the distance estimation should be altered as well, as it was explained in Section 3.2.2.

Another future functionality can be implementing an iterative feedback of the estimated localization among all devices. This information can be used to recalculate the each node's own estimations and to reduce their errors. This approach may be implemented together with a relative localization system where the reference nodes are selected among those with the best RSSI.

For solving the collinearity problem, it is suggested the implementation of a function, which uses only the best links per node to calculate the distances. Then, these distances can be compared and through geometrical functions, the collinear configuration can be found. An approach to this solution is mentioned in [MF04].

To improve the ZigBee application, a function that detects the absence of the coordinators (orphan state) on the routers can be implemented. Once the routers are in this state, they can restart the coordinator selection process. This solution will guarantee the independency of the systems from the coordinator state.

As a solution for the coordinator selection shown in case 3 of Section 4.2.3, where two coordinators emerge and therefore two ZigBee networks are created, it is suggested to implement a function that scans the channel after the network is established. The scans will be performed only by the coordinator. If it finds another conformed network, the coordinators will decide, through the definition of some rules, which network/coordinator will remain. The other coordinator should execute a disassociation process, in order to finish its network.

[2]Devices located in a fixed position, which function is to recharge the stand-alone robots

Another solution is to use a different range of PAN IDs between the selection process and the network conformation, for example, utilizing a PANID bigger than 0x0f00 (hexadecimal) for the coordinator selection. When a coordinator finishes its selection process, it should change its PAN ID for a number lower than this value. In such a way, the devices searching for other coordinators will know when a coordinator is also running the selection process or when there is a coordinator, which already established a network. In the second case, the device that performs the scans becomes a router, no matter its priority.

Finally, it is recommended the implementation and assessment of this this localization system, as an embedded application based on the results presented in this thesis and new researches; and its utilization as an approximation of the nodes position and as the communication system of the REPLICATOR robots. In order to improve the system's accuracy, it is suggested the use for a short time of other sensors, such as infrared or ultrasound to recalibrate the system, without significantly increasing the system's power consumption.

Bibliography

[80297] Ieee std 802.11-1997 part 11 : Wireless lan medium access control (mac) and physical layer (phy) specifications, 1997.

[80203] Ieee std 802.15.4-2003 part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans), May 2003.

[all11] Zigbee alliance. http://www.zigbee.org/, 2011.

[Ant11] Antenova. *Titanis 2.4 GHz Swivel SMA Antenna*, 2011. Part No. B4844 / B6090. Product specification AE030054-I.

[APA06] N. A. Alsindi, K. Pahlavan, and B. Alavi. An error propagation aware algorithm for precise cooperative indoor localization. In *Proc. IEEE Military Communications Conf. MILCOM 2006*, pages 1–7, 2006.

[ASS01] Chih-chieh Han Andreas Savvides and Mani B. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the seventh annual international conference on Mobile computing and networking, Mobicom 2001*, pages 166–179, 2001.

[ASS02] Heemin Park Andreas Savvides and Mani B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. pages 112–121, 2002.

[ASS05] A. A. Ahmed, H. Shi, and Y. Shang. Sharp: a new approach to relative localization in wireless sensor networks. In *Proc. 25th IEEE Int Distributed Computing Systems Workshops Conf*, pages 892–898, 2005.

[AV06] C. Alippi and G. Vanini. A rssi-based and calibrated centralized localization technique for wireless sensor networks. In *Proc. Fourth Annual IEEE Int. Conf. Pervasive Computing and Communications Workshops PerCom Workshops 2006*, 2006.

[AWC+05] V. S. Abhayawardhana, I. J. Wassell, D. Crosby, M. P. Sellars, and M. G. Brown. Comparison of empirical propagation path loss models for fixed wireless access systems. In *Proc. VTC 2005-Spring Vehicular Technology Conf. 2005 IEEE 61st*, volume 1, pages 73–77, 2005.

[CALT10] Joel T. Johnson Curt A. Levis and Fernando L. Teixeira. *Radiowave propagation, physics and applications*. John Wiley & Sons, Inc., 2010.

[CDGS04] K. K. Chintalapudi, A. Dhariwal, R. Govindan, and G. Sukhatme. Ad-hoc localization using ranging and sectoring. In *Proc. INFOCOM 2004. Twenty-third AnnualJoint Conf. of the IEEE Computer and Communications Societies*, volume 4, pages 2662–2672, 2004.

[CP96] E. Christensen and S. E. Paulsen. Improved coverage and interference predictions using line-of-sight detection and correction. In *Proc. IEEE 46th Vehicular Technology Conf. 'Mobile Technology for the Human Race'*, volume 3, pages 1638–1642, 1996.

[CSL02] Jan Rabaey Chris Savarese and Koen Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks, June 2002. USENIX Annual Technical Conference.

[DMT04] Daniela Rus David Moore, John Leonard and Seth Teller. Robust distributed network localization with noisy range measurements. Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems, November 2004.

[EGT+99] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi. An empirically based path loss model for wireless channels in suburban environments. 17(7):1205–1211, 1999.

[FBR+94] M. J. Feuerstein, K. L. Blackard, T. S. Rappaport, S. Y. Seidel, and H. H. Xia. Path loss, delay spread, and outage models as functions of antenna height for microcellular system design. 43(3):487–498, 1994.

[GD10] Joan Condell Gabriel Deak, Kevin Curran. Device-free passive localisation using rssi-based wireless network nodes. Technical report, University of Ulster, Intelligent System Research Centre, 2010.

[GTC08] Patrick Allard-Jacquin Gilles Thonet and Pierre Colle. Zigbee wifi coexistence. White paper and test report, Schneider Electric, 37 Quai Paul Louis Merlin 38000 Grenoble, France, April 2008.

[Ins09] Texas Instrument. *A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications*, April 2009. SWRS081A.

[JBS03] Michael Salib Jonathan Bachrach, Radhika Nagpal and Howard Shrobe. Experimental results and theoretical analysis of a self-organizing global coordinate system for ad hoc sensor networks, 2003.

[JZS05] J. A. Stankovic J. Zhang, T. Yan and S. H. Son. Thunder: A practical acoustic localization scheme for outdoor wireless sensor networks. Technical Report CS-2005-13, University of Virginia, Department of Computer Science, 2005.

[Kay93] Steven M. Kay. *Fundementals of Statistical Signal Processing: Estimation Theory*. Number 0133457117. Prentice Hall, 1993.

[LK10] P. Levi and S. Kernbach. *Symbiotic Multi-Robot Organisms. Reliability, Adaptability, Evolution*. Springer, 2010.

[Log92] T Logsdon. *The Navstar Global Positioning System*. Kluwer Academic Publishers, 1992. Provided by the SAO/NASA Astrophysics Data System.

[Lyo97] Richard G. Lyons. *Understanding Digital Signal Processing*. Addison Weskey Longman, Inc., 1997.

[Mes06] Adam Messer. *Wireless networking in the developing world*. Limehouse Book Sprint Team, 2006.

[MF04] Lambert Meertens and Stephen Fitzpatric. The distributed construction of a global coordinate system in a network of static computational nodes from inter-node distances. Technical Report KES.U.04.04, Kestrel Institute, March 2004.

[NBPT03] Erik Demaine Nissanka B. Priyantha, Hari Balakrishnan and Seth Teller. Anchor-free distributed localization in sensor networks. Technical report, MIT Laboratory for Computer Science, April 2003.

[Net10] Daintree Networks. Applying mesh networking to wireless lighting control. Technical report, Daintree Networks, Inc., 2010.

[NN01] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps). In *IN GLOBECOM*, volume 5, pages 2926–2931, August 2001.

[NN03] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps) using aoa. In *IN GLOBECOM*, pages 1734–1743, July 2003.

[O'H09] Tom O'Haver. An introduction to signal processing in chemical analysis. University of Maryland, 2009.

[Pal10] Amitangshu Pal. Localization algorithms in wireless sensor networks: Current approaches and future challenges. Technical report, Macrothink Institute, 2010. Network Protocols and Algorithms.

[PBDT05] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller. Mobile-assisted localization in wireless sensor networks. In *Proc. IEEE 24th Annual Joint Conf. of the IEEE Computer and Communications Societies INFOCOM 2005*, volume 1, pages 172–183, 2005.

[PH06] N. Patwari and A. O. Hero. Indirect radio interferometric localization via pairwise distances, May 2006. in Proceedings of 3rd IEEE Workshop on Embedded Networked Sensors.

[Rao07] Shreharsha Rao. Estimating transmission range for zigbee and proprietary short-range wireless devices in 900mhz and 2.4ghz ism band. Technical report, Texas Instruments, May 2007.

[rep11] replicator. http://www.replicators.eu, 2011.

[SB98] I. W. Selesnick and C. S. Burrus. Generalized digital butterworth filter design. 46(6):1688–1694, 1998.

[Sey05] John Seybold. *An introduction to RF propagation that spans all wireless applications*. John Wiley & Sons, Inc., 2005.

[SLA+03] Ghassemzadeh S., Greenstein L., Kavcic A., Sveinsson T., and Tarokh V. Uwb indoor path loss model for residential and commercial buildings. In *Proc. VTC 2003 Fall Vehicular Technology Conf. 2003 IEEE 58th*, volume 5, pages 3115–3119, 2003.

[SRB01] C. Savarese, J. M. Rabaey, and J. Beutel. Location in distributed ad-hoc wireless sensor networks. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '01)*, volume 4, pages 2037–2040, 2001.

[SS02] S. Simic and S. Sastry. Distributed localization in wireless ad hoc networks. Technical Report UCB/ERL M02/26, UC Berkeley, 2002.

[Sug06] Masashi Sugano. Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard. In *Wireless and Optical Communications*, pages 1–6. IASTED/ACTA Press, 2006.

[Tex09a] Texas Instrument. *CC2530 ZigBee Development Kit User's Guide*, 2009. swru209b.

[Tex09b] Texas Instrument. *TPS61097. Low input voltage synchronous boost converter with low quiescent current*, June 2009. SLVS872B.

[Tex09c] Texas Instruments, Inc., San Diego, California USA. *OS Abstraction Layer Application Programming Interface*, 2009. SWRA194.

[XP09] Yang Xiao and Yi Pan. *merging Wireless LANs, Wireless PANs, and Wireless MANs: IEEE 802.11, IEEE 802.15, 802.16 Wireless Standard Family*. Wiley editorial, 2009.

[YYK02] Mario Gerla Yunjung Yi and Taek-Jin Kwon. Efficient flooding in ad hoc networks using on-demand (passive) cluster formation. In Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2002.

[zig08a] zigbee.org. Zigbee-2007 layer pics and stack profiles. Technical Report 08006r03, ZigBee Alliance, June 2008.

[zig08b] zigbee.org. Zigbee specification. Technical Report 053474r17, ZigBee Alliance, January 2008.

[ZS11] Texas Instrument Z-Stack. www.ti.com/z-stack, 2011.

Declaration

All the work contained within this thesis, except where otherwise acknowledged, was solely the effort of the author. At no stage was any collaboration entered into with any other party.

(Jeimy Catherine Millán Ochoa)