Indoor Positioning

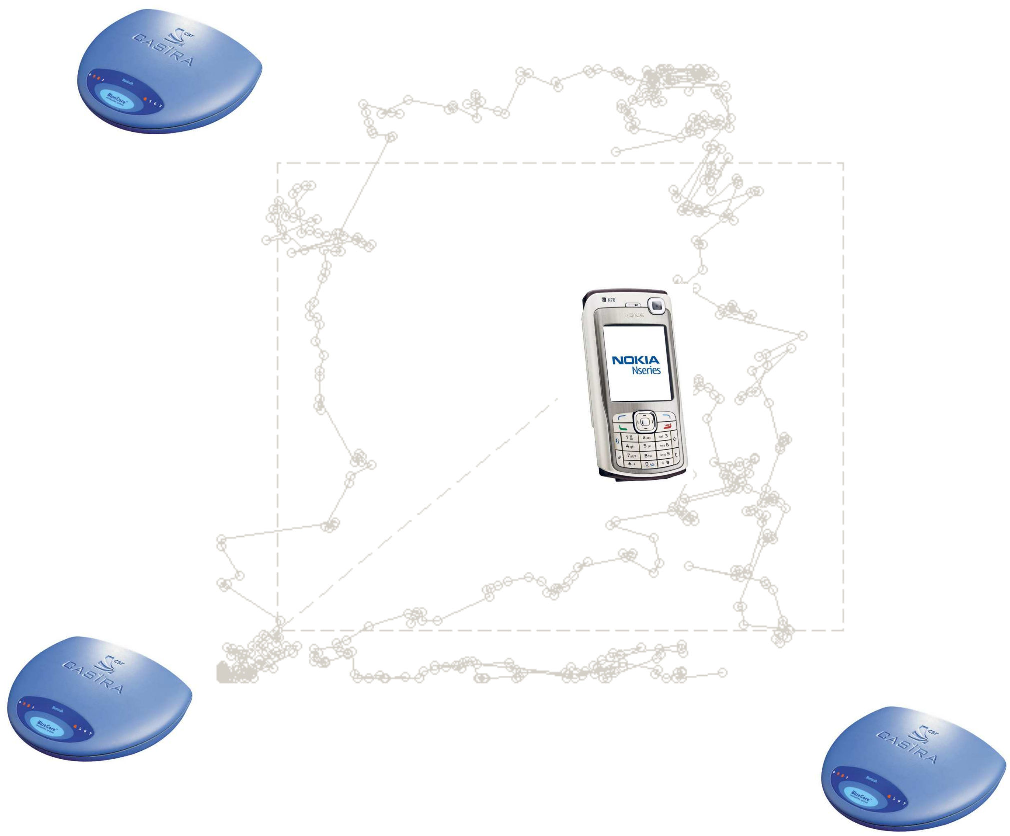
Based on Bluetooth

Group 872

Janne Dahl Rasmussen, Achuthan Paramanathan

Yassine Nassili, Anders Grauballe, Mikkel Gade Jensen

Supervisors: Henrik Schiøler, Jimmy J. Nielsen



8th Semester

Spring 2008

Institute for Electronic Systems

Networks and Distributed Systems

AALBORG UNIVERSITY

Department of Electronic Systems

Networks and Distributed Systems

Fredrik Bajers Vej 7C

Telephone +45 96 35 87 00

Fax +45 98 15 17 39

http://www.kom.aau.dk

Title:

Indoor Positioning Based on Bluetooth

Theme:

Distributed Systems Design

Project period:
8th semester,
February - June 2008

Project group:
NDS, Group 872

Participants:
Yassine Nassili
Anders Grauballe
Mikkel Gade Jensen
Janne Dahl Rasmussen
Achuthan Paramanathan

Supervisors:
Henrik Schiøler
Jimmy J. Nielsen

Number of prints: 9

Number of pages: 106

Number of appendices and character:
1 pcs. CD-ROM

Finished: June 3rd, 2008

Synopsis:

The aim of this project is to investigate how to do indoor positioning based on Bluetooth. This is done by outlining different possible scenarios and the actions needed to obtain the position of a Mobile Device. A prototype system is designed to track a Mobile Device in a room in two dimensions by measuring Received Signal Strength at three Base Stations. Two different methods are tested on the obtained measurements: A simple trilateration based on measurements from all three Base Stations without any history of the position, and a particle filter implementing a mobility model of the user.
The methods are compared and it is concluded that the measurements are too noisy to obtain a clear trend using trilateration. The Particle Filter using the Less Drunk model performs better and is able to produce a track with a trend similar to the correct path.
It is concluded that Bluetooth with use of Received Signal Strength measurements is not an optimal technology for indoor positioning, but it gives good estimation.

Preface

This project has been carried out on the 8th semester, spring 2008 at Networks and Distributed Systems, Aalborg University by group 872. The theme for this semester was Distributed Systems Design. The focus group for this report is people with an interest in indoor positioning and the use of Bluetooth for other purposes than data exchange. In this report figures, pictures and tables are labeled with chapter and figure number for easy reference, e.g. Figure 4.2 is the second figure in Chapter 4. References for literature are shown as e.g. [Fig07]

A CD is enclosed with this report and it contains the following:

- This report in PDF format
- The data from the tests
- MATLAB code used to process the test-data

We would like to thank our supervisors Henrik Schiøler and Jimmy Nielsen who have guided us in our work. Their advice, idea and support, throughout the project has been very helpful. A special thanks to Casper Madsen from Cambridge Silicon Radio (CSR) for his support with the provided Bluetooth equipment and CSR for providing it.

Yassine Nassili

Anders Grauballe

Mikkel Gade Jensen

Janne Dahl Rasmussen

Achuthan Paramanathan

Contents

Chapter 1

Introduction

Positioning of Mobile Device (MD)s is becoming more and more common in consumer products these days. The knowledge of where a specific MD is located can be used to offer new services for the users. The most well known form of application is navigation driven by the Global Positioning System (GPS). Today a GPS navigation device is available for everyone at a low cost. These devices are typically to be installed in cars to show location and route to the driver, but lately a number of mobile phones is also equipped with a GPS receiver e.g. Nokia N95 and Sony Ericsson K530.

In GPS it is necessary to have line of sight (LOS) to the GPS satellites to obtain accurate positions. I.e. GPS does not perform properly in a number of scenarios e.g. in urban areas, indoor or under ground. If a location based service is needed in one of these scenarios, other technologies must be considered. Many different solutions can be thought of depending on which requirements exist for the system. Examples of this could be a pressure sensitive floor to monitor which rooms there are people inside or ultra sonic receivers detecting sound waves from emitters carried by people.

The most common choice of technology is short range wireless RF signals which are used in Bluetooth or IEEE 802.11. This is due to the huge market penetration of these technologies. Today most mobile phones or even laptops support it, thus applications using RF signals should be easy to implement at a low cost.

An example of a real life implementation of indoor positioning is a system of intelligent shopping carts (MediaCarts) in a supermarket in Singapore [AFP08]. The carts which are being deployed later in 2008, can sense their location within the supermarket and guide the shopper to find the products he needs by showing the position in a map on a small display. The display is also used for showing promotional messages about specific products in the proximity of the cart. The system is based on RFID chips to detect proximity around the store. [Med08]

This project aims at developing a positioning system of MDs based on existing wireless technology. Bluetooth has been chosen as technology because of its huge market penetration in MDs. Also the project is done in collaboration with CSR, a leading manufacturer of Bluetooth chips, who will be providing development kits and support for these.

Based on this, the initial problems of the project are as follows:

- **What are the possibilities for performing indoor positioning based on Bluetooth and existing methods for determining distance and direction to obtain location?**
- **How it is possible to minimize location errors when performing indoor positioning using a mobility model and filter?**

Indoor positioning can be used in various scenarios. In order to determine the project scenario, some of these scenarios will be described in the following section. The project scenario is used further to determine the system requirements.

1.1 Scenarios

The scenarios can be divided into different categories. The first one contains the scenarios where the indoor positioning is used for path finding. Basically this category applies to any larger buildings which are hard to navigate in. The second category contains the scenarios where the user needs to find something or someone. It can be equipment in a hospital or residents at a retirement home. The last category contains the scenarios where the location information is transparent to the user. I.e. location based services where the user's MD can react on its location to provide new services or optimize existing ones. In the following the different categories will be described more thoroughly as well as the different scenarios.

1.1.1 Path Finding

Common for the scenarios in this category is that the user needs to be guided to a specific location. This would only be relevant in places the user is not familiar with or places which are big and hard to navigate in. Examples of such places are a shopping mall or an airport.

Shopping Mall
In this scenario the user is inside a shopping mall. When the user arrives he should receive a message on his mobile phone with a program he can install. When the program is installed he can search for a specific store or a specific type of store, e.g. a shoe store. The program will then guide the user to the location of the store using indoor positioning.

The application could also be preinstalled in MDs which the user could loan at the entrance to the shopping mall. This would make it possible to make the user interface more user-friendly because the screen could be bigger than on a mobile phone and the buttons could be more specific to finding stores in the mall.

Another approach would be to have a framework for the location software similar to GPS where the only thing the user needs to download is a specific map. The user would have the possibility to download the map for the shopping mall when he arrives but the program would be the same for any location.

Airport
In this scenario some users would be familiar with the airport, e.g. business men who are traveling every day or several times a month. Some other users would not be familiar with the airport at all, e.g. people who are on vacation etc. The last group of users would obviously have more use of the system than the first group. When the user purchases his ticket he should also receive a MD which has an application installed or be able to download the application to his mobile phone. The application can be used to guide the user to the correct check-in and gate. When the user gets to his final destination he returns the MD so it can be reused. In this scenario it would be important to have a way of updating the destination if e.g. the gate number is changed.

1.1.2 Tracking

In this category, the user needs to find or monitor a given subject in large buildings. The subject can either be portable electronic devices or people moving around. The tracked subject may be missing from its usual position and needs to be located. Also a subject could be monitored to ensure that it does not leave the building unauthorized, e.g. in case of theft.

Office
In large offices it is often difficult to keep track of various equipment such as projectors, laptops or other MDs. Especially if this equipment is shared among employees or available for loan. In most cases, there would be a record of who is responsible for a given MD at the time, but records does not contain information on where the MD is located at a given time. Also the responsible person might not know exactly where a specific MD is if he is responsible for several of them.

The problem could be solved by installing a wireless transceiver in every item potential for future localization or using build in technology in laptops or mobile phones. The ID of all MDs could be stored in the record to make it possible for the employees to track every MD using the wireless Access Point (AP)s in the office.

Retirement Home
Senior citizens at a retirement home sometimes need help taking the correct medicine at the correct time. The citizens of such a retirement home may be in many places. They can visit each other, be outdoor in gardens or in one of more common rooms. Therefore it would be helpful for the staff if they were able to see where a specific person was located at a specific time.

A tracking system in this scenario requires all citizens to carry a device which could be tracked. This could be a small object specially made for this purpose like a bracelet. The staff would need a computer where they could see the position of the citizens which needs there medicine so they could be found quickly. The precision of this system only needs to be room based i.e. which room is the citizen in and not if they are in a specific part of the room.

A similar system could be used in theme parks or museums where the parents may want to track there children if they were walking around on their own.

1.1.3 Location Based Services

Positioning can also be utilized for other purposes which is transparent to the user. I.e. a MD can provide different services based on its location. This way the user will experience a better quality of a known service or discover a new kind of application for his MD.

Cellular Handover

In future generations of mobile communications, it is likely that various wireless technologies can be used for the the same kind of traffic within the same session. I.e. handover can be performed between two different kinds of networks e.g. from GSM to WLAN. (This is an example of Cross-Layer protocol design). Switching between such different networks during a session can often result in a huge rerouting of traffic. The handover from cell 1 to cell 2 should only be performed if e.g. the user stays within cell 2 for a longer amount of time or if cell 1 is out of range. An example of a situation where handover could be avoided is shown in Figure 1.1. The coverage time of cell 2 can be estimated/predicted based on the known coverage area of cell 2, and the current position and velocity of the MD. Combined with the current handover procedures based on signal strength this will in some cases maintain the same traffic path and the user will not experience any lag.

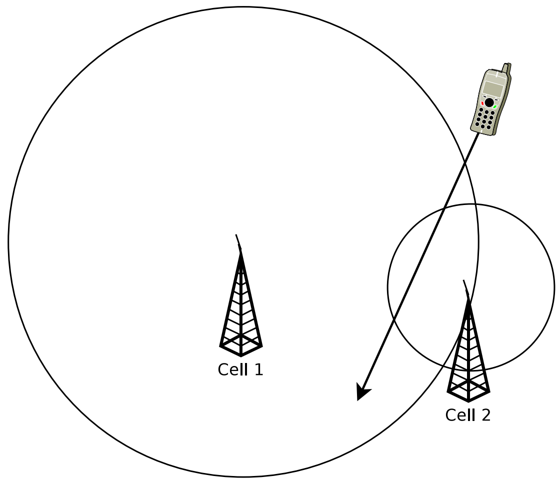


Figure 1.1: *The MD starts in cell 1 and moves briefly through cell 2. If the estimated coverage time of cell 2 is small, the MD can stay connected in cell 1 and a handover can be avoided.*

Proximity Information
Location information can also be used as a base for providing information to the user at the right time and place. A master thesis from Aalborg University [RO07] uses a museum as project scenario. In this scenario the user will receive visual or audible information on his MD whenever it is inside a specific room or in the proximity of a work of art. This service can be provided to the user without him knowing that it is location based. It can also be combined with a path finding application to find the object the user is interested in and get information about this object.

1.2 Choice of Scenario

In the previous section, three different kinds of scenarios have been presented:

- Path finding
  - Requires calculation of path and determination of current position
- Tracking
  - Requires determination of current position
- Location based services
  - Requires determination of current position and maybe estimation of moving pattern.

In this report it is chosen to work with the tracking system because it does not require calculation of a path or estimation of moving pattern which is more complex than tracking alone. This means that it will be possible to focus only on the positioning system. Furthermore it has been decided to choose an office or shopping mall like environment. This is an environment where the user would move in straight lines mostly along walls. This choice has made it possible to set up a simple test example where the MD is either standing still in a location or moving on straight lines.

Chapter 2

The Location Stack

This section is based on [Hig03] and [Hig02].

To structure the mechanisms of positioning, it is chosen to use a layered model to split the different activities into logical blocks of functionality. For this purpose The Location Stack is chosen as it provides abstraction layers for basic measurements and calculations, but also for higher layer enhancements such as mobility models.

In 2002, Jeffery Hightower et al. [Hig02] proposed the Location Stack. It is a seven layer reference model for location in ubiquitous computing similar to the well known OSI reference model for computer networks. The motivation for this model is that nowadays location systems use individual design principles which are often incomparable or incompatible. The Location Stack should serve as a common vocabulary for future location systems to make them based on a standard infrastructure. This makes it easier to replace parts of a given system and for other developers to continue ongoing work.

The Location Stack is based on five design principles:

Fundamental measurement types from sensors such as distance, angle or proximity can be used for location purposes. Other measurements like light intensity or electromagnetic characteristics can also be used for location, but mostly for contextual information for other purposes.

A combination of measurements yield the location of an object. The combined measurements can be of various types e.g. distance and samples from onboard compass results in a position and a direction.

Objects can be related to each other based on proximity, inside a specific region or geometric form.

Uncertainty must be preserved to make more correct decisions at higher abstraction levels.

Applications are concerned with activities based on location information. Typically, applications use a location to make a decision whether to execute a certain activity or not.

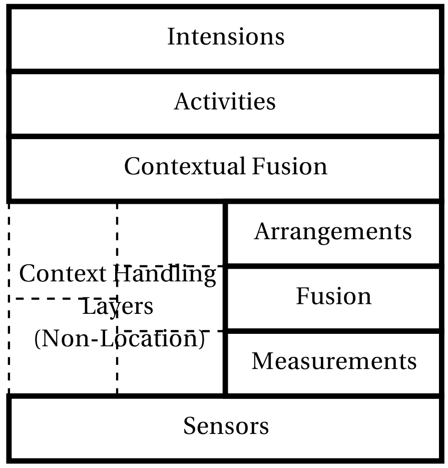


Figure 2.1: *The location stack [Hig02].*

Figure 2.1 shows the seven layer Location Stack. The following describes the contents and tasks of the layers bottom up:

Sensors

The sensors are hardware which are sensing conditions of the physical world e.g. voltage, watt or frequency. Different examples of such sensors can be seen in Section 2.1. The detected raw data are sent to the above layers.

Measurements

In this layer the raw data, received by the sensors, are converted into useful measurement for use of higher layers. These data can be sorted into different classifications for example; Received Signal Strength (RSS), Time of Arrival (ToA) and Angle of Arrival (AoA). In Section 2.2 a more detailed description of these methods can be found.

Fusion

The measurements received from the Measurement Layer, are used to achieve an estimated representation of the location of an object. These representations can be e.g. triangulation, trilateration and multilateration. These methods are further described in Section 2.3. Furthermore different mobility models can be used which will ensure a more correct positioning of the object.

Arrangements

This layer handles the relation between two or more objects which have been individually located by the Fusion Layer. These locations are then categorized e.g. in a database based on some predefined rules. For example, if the located objects are within 1 meter of one another then those objects will be grouped in one category. The above layer uses this information, from the database, for different purpose or applications.

Contextual Fusion

This layer merges the position of a given object with non-location information. This non-location information could be for example a map representing a particular place e.g. an airport, shopping mall etc. and in this case the current position of the object will be attached with a set of coordinates, which relate to a spot on a map. Variety of non-location information can be considered merged with the positioning information and this will depend on the application of use.

Activities

By using the knowledge achieved from the Arrangement and the Contextual Fusion layer, the Activity Layer provides semantic state defined applications, this means that the application triggers an event based on a set of rules. These could be e.g in a prison where the system registers that the prisoners are not all in their cell, and the time is 10.00 PM, and in this case the application will activate the alarm.

Intentions

This layer adjusts the system according to the user based on his needs, his behavior or his intention. For example, a user uses a path finding system in a shopping mall to find the way to a specific location. If the user suddenly changes the direction because he saw some interesting electrical equipment, then this layer will identify his intention, according to the position and the amount of time the user stays in this position. Based on this the Intention Layer will present some information e.g. regarding those electrical equipments that have a discount of 25%.

As stated in the initial problem, the focus in this project is on the actual positioning. Also, for simplicity, it has been chosen to locate MDs independently of others. Thus, only the following layers will be considered further: Sensors, Measurements, Fusion and Contextual Fusion. In the following sections the different parts that may be a part of this project are described with the corresponding layers.

2.1 Sensors

In this project Bluetooth is used for positioning, thus the sensing capabilities for obtaining location information must be based on a radio transceiver. The definition of the sensor itself can be the radio transceiver, but there may be other abstraction levels for sensors within the radio transceiver. Examples of these are described in this section.

2.1.1 Radio Transceiver

A radio transceiver is an intergrated circuit which is capable of receiving (and transmitting) radio signals in a specified frequency band. To receive the best signal possible it is connected to an antenna designed to receive the particular frequency or frequency range. This means that it must have a length or size that is equal or a ratio of the wave length. The radio transceiver can sense different physical characteristics in the signal e.g. angle (in case of multiple antennas), frequency, RSS etc.

2.1.2 Crystal Oscillator - Clock

The crystal oscillator is a device which creates a signal with a precise frequency that can be used as clock. This provides the system with a clock where the number of ticks can be measured e.g. to measure the time between two detected events. A clock is a sensor on a higher level as it needs another sensor to trigger the measuring. E.g. the Time Difference of Arrival (TDoA) can be measured, triggered by a radio receiver.

2.2 Measurements

In this section different techniques of converting raw data from the sensors into some useful measurements are described. These measurements can however be influced by several factors, escpecially in an indoor scenaro. This is due to the nature of radio signals and therfore these factors are also described in this section.

2.2.1 Radio Signals in Indoor Scenarios.

This section is based on [JA95] and [Par01].

In the initial problem it is stated that part of the goal of this project is to find the possibilities of indoor positioning with Bluetooth. Because Bluetooth is a wireless technology it is necessary to investigate how wireless signals are behaving in an indoor environment. This section will be explaining the basic principles of wireless signals in an indoor scenario.

If a wireless signal propagates in an ideal space with no obstacles or anything else to interfere with the signal, the received part will have traveled in a straight line between the transmitter and the receiver. No scenarios will involve a completely ideal space, but outdoor scenarios will be a lot closer than indoor scenarios. At indoor scenarios there are walls, doors, windows and maybe some movable objects like chairs and tables. The obstacles will have different effect on the signal depending on their size, material and surface.

Penetration

The signal will be able to penetrate some obstacles, but depending on the frequency of the signal and the material of the obstacles some or all of the signal's energy will be absorbed by the obstacle. The space behind the obstacles is called a shadow region. In this region there can be some or no part of the signal's energy left.

Diffraction

Obstacles with sharp edges will cause the signal to diffract over the edge. This can be seen in Figure 2.2. As the figure shows, some of the signal will be bend around the edge and propagate in a new direction.

Reflection

When the signal hits a wall or the floor in a building the signal will be reflected. This causes the signal to continue in a new direction. The obstacle will, depending on the surface, absorb some of the signal causing it to be weaker after reflection than before. Examples of reflected signals can be seen in Figure 2.3 (B) and (C).

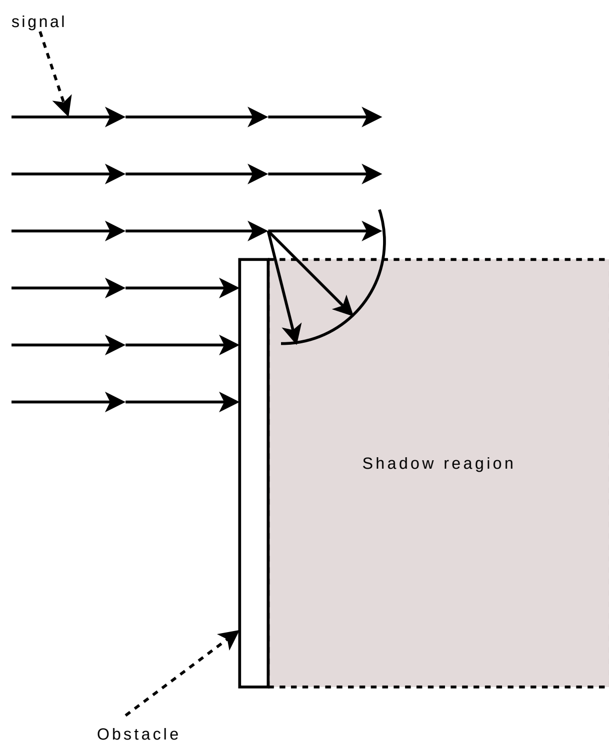


Figure 2.2: *The obstacle causes the signal to diffract around the edge.*

Multipath Propagation

This phenomenon is the consequence of reflection and diffraction of the radio signals. When a signal is received from multiple paths, e.g. from reflections on walls etc., it may be phase shifted from the original signal received at LOS. An example of multipath can be seen in Figure 2.3.

At the receiver, the signals from the multipath is added to the LOS signal resulting in either constructive or destructive addition. Figure 2.4 shows the extreme cases of this effect. The first path (A) represents the LOS and the echo paths (B) and (C) represents other signal paths.

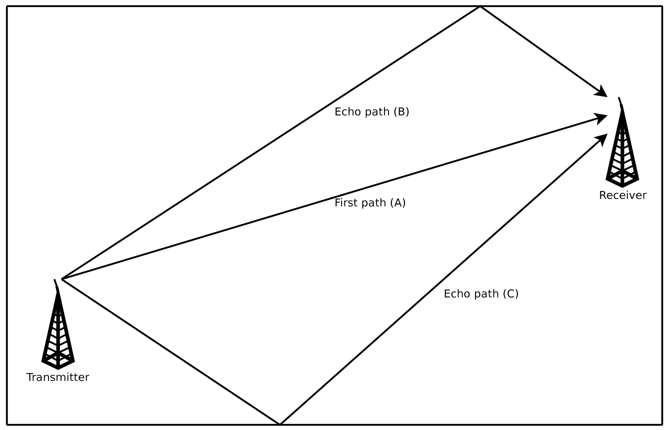


Figure 2.3: *An example of multipath. [Par01]*

In Case 1 in Figure 2.4 the echo signal is not phase shifted, which leads to constructive addition at the receiver i.e. the amplitude of the received signal is the amplitude of signal (A) plus the amplitude of signal (B). In Case 2 the echo signal is shifted π (180°or one half of a wavelength) which leads to destructive addition i.e. the amplitude of the received signal is the amplitude of signal (A) minus the amplitude of signal (C). In worst case a phase shifted signal can annihilate the LOS signal completely. [Par01]

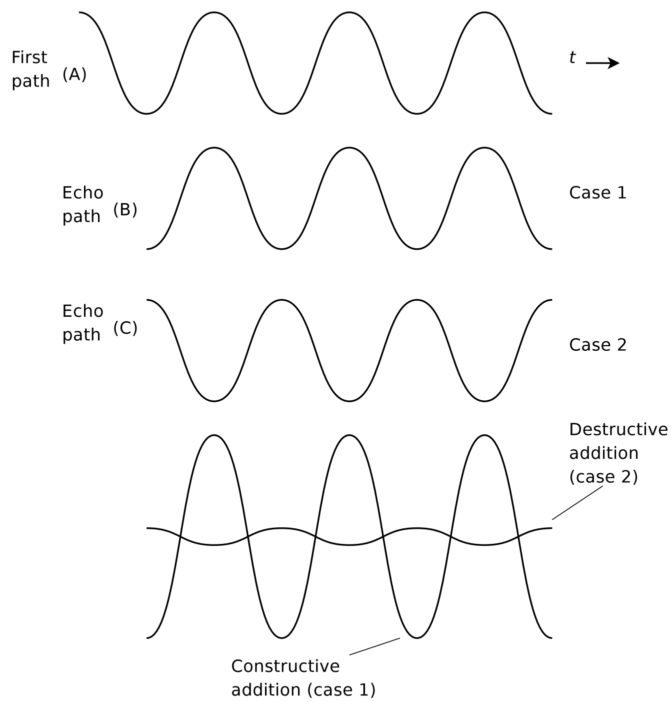


Figure 2.4: *The effect of constructive and destructive addition in multipath conditions. [Par01]*

2.2.2 Time of Arrival

For measuring the distance between a Base Station (BS) and a MD, the ToA can be used. The signal from the transmitter contains a time stamp indicating the moment when the signal was transmitted. The receiver obtains another time stamp instantly when receiving the signal from the transmitter. Given that the transmitter and receiver are time synchronized, and both time stamps are known, it is possible to calculate the traveling time of the signal between transmitter and receiver. This is also known as the signal propagation delay. Assuming that the speed of a radio signal is the speed of light in vacuum, the distance can be obtained as:

$$d = c \cdot t \tag{2.1}$$

Where c is the speed of light and t is the propagation delay.

The accuracy of the measurement is dependent on the granularity of the internal of the receiver i.e. the time elapsed between two consecutive clock ticks. This means that if the signal arrives between the two ticks, it will get the time stamp of the latter of the ticks. Using Equation 2.1 the maximum accuracy can be found as:

$$d = c \cdot t = \frac{c}{f_{clock}} \tag{2.2}$$

In the case of a clock granularity like the ones found in an average home computer (2 GHz), the accuracy is:

$$d = \frac{c}{f_c lock} = \frac{3 \cdot 10^8 m/s}{2 \cdot 10^9 Hz} = 0,150m \tag{2.3}$$

The downside of ToA is that the transmitter clock must be synchronized with the receiver clock. Synchronization may be very complex and expensive to deploy. Atomic clocks may also be necessary to use to minimize the clock drift in the system. This proves practically impossible when one of the two devices in the system is a mobile phone or a similar hand held low-cost MD. [Fig07] and [Dye01]

2.2.3 Time Difference of Arrival

Synchronization between sender and receiver can be avoided by introducing the concept of TDoA. In this approach, time synchronized BSs are needed and the measurements can be based on both uplink and downlink. In uplink, the MD is sending the signal and the BSs are comparing the ToA to determine the distance to the MD. I.e. the calculations are done in the network of the BSs. In downlink, the BSs are sending a synchronized signal which is received and compared by the MD. I.e. the calculations are done in the user equipment. [Fig07] and [Dye01]

2.2.4 Received Signal Strength

The Received Signal Strength (RSS) can be used to estimate the distance between two network nodes based on measuring the RSS and the Transmitted Signal Strength (TSS). This is achieved by modeling a system that gives a RSS value based on TSS, path loss and shadowing effect according to a given distance. The path loss and shadow effects have an impact on the transmitted signal because the wave of the transmitted signal propagates through the air and obstacles encountered along the path. Due to this, the energy of the transmitted signal will be distorted as described in Section 2.2.1. In this section a mathematical model is described. Its starting point is achieving RSS based on the path loss in free space where the transmitting and receiving nodes are in LOS of each other.
The following model describes the RSS for an indoor wireless system [Gol05]:

$$P_r dBm = P_t dBm + KdB - 10\beta log(\frac{d}{d_0}) \quad (2.4)$$

where P_r is the received signal strength and P_t is a component which depends on the transmitted signal energy, K is a constant which describes the characteristics of the antenna, β is the path loss exponent above the reference distance d_0.

The distance between the transmitting- and receiving node is denoted as d and d_0 is a reference distance for the antenna far field (at this distance the received power equals the transmitted). Assuming omnidirectional antennas, the value of K can be obtained by computing:

$$KdB = 20log(\frac{\lambda}{4\pi d_0}) \quad (2.5)$$

Where λ is the signal wavelength.

In the following an example is given regarding calculating the RSS using Equation 2.4. Consider a 2.4 GHz indoor wireless system transmitting a signal with power level equal to 2dBm. The receiving device is in LOS of the transmitting device but with varying distance d. The objective of this example is to find those different RSS value for a given d between these two devices. Based on the given information, the value of K can be calculated (for $d_0 = 1m$):

$$KdB = 20log(\frac{0.125}{4\pi}) = -40.05dB \quad (2.6)$$

The component β is assumed to have the value 2 which is proposed as a typical component for an office environment in [Gol05].

Inserting K and β in equation (2.4) gives:

$$P_r dBm = 2 - 40.05 - 10 \cdot 2log(\frac{d}{d_0}) \quad (2.7)$$

The result of this example is illustrated on Figure 2.5.

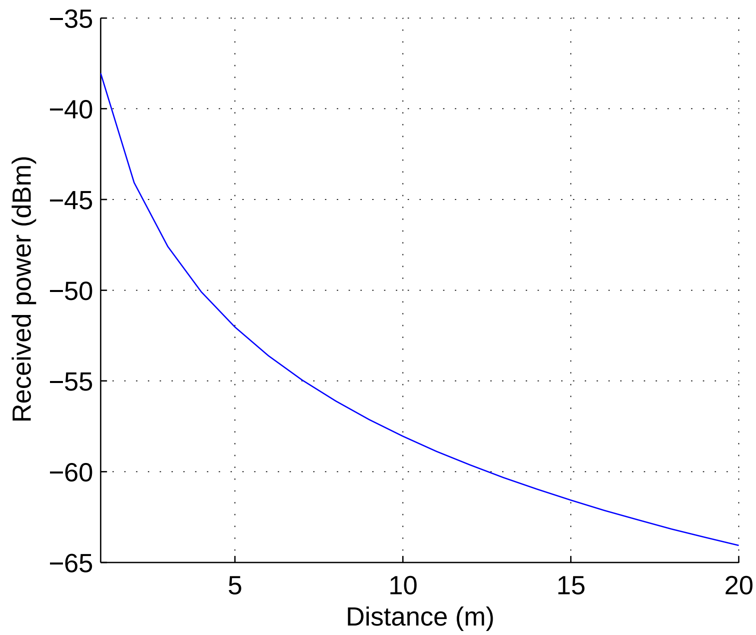


Figure 2.5: *Received signal strength as a function of the distance. This figure illustrates that the RSS value decreases as the distance between the transmitting and receiving node increases*

Equation (2.4) takes only account of measuring the RSS when there are no obstacles between the transmitting and receiving nodes.

In an indoor environment there will be a destructive propagation on the RSS from e.g. moving persons, different type of reflective materials etc. When taken this contribution into consideration a more complex model is required to accommodate to the stochastic behavior of their nature.

An approach to simplify this complexity is to consider the contribution from this stochastic behavior as a Gaussian distributed noise with standard deviation σ and thereafter add this noise to Equation 2.4 which yield to:

$$P_r dBm = P_t dBm + KdB - 10\beta log(\frac{d}{d_0}) + \sigma \tag{2.8}$$

The received signal strength can be measured in different units. In this project the RSS will be measured in dBm (decibel - milli watts) and Received Signal Strength Indicator (RSSI). The dBm is a logarithmic measurement of signal strength and can directly be converted into mW or vice versa. An example of this could be:

$$100mW- > 10 \cdot log(100) = 20dBm \tag{2.9}$$
$$50mW- > 10 \cdot log(50) = 16.98dBm \tag{2.10}$$
$$25mW- > 10 \cdot log(25) = 13.97dBm \tag{2.11}$$

The RSSI is a IEEE 802.11 standard that defines a measurement unit where the RF energy is measured and converted to a numeric value(1-byte) [BS08]. For example the RSSI range of Bluetooth is from -128 to 127. This value represents 255 different signal level of a Bluetooth device. For more details, see Section 6.2.2.

2.2.5 Angle of Arrival

The AoA is a method of detecting the direction of a radio signal by measuring this signal on an antenna array. This is done by measuring the phase difference of the signal on the different arrays. The phase difference will give the AoA which is the direction of the incoming signal.

In Figure 2.6 a radio wave is emitted from a transmitter and is received by a receiver with two array elements which receive the signal at different times. The element close to the transmitter will receive the signal indicated by the blue color first and the other element which is more distant from the transmitter will receive it later. This corresponds to a difference in the phase which is the AoA indicated by the black line.

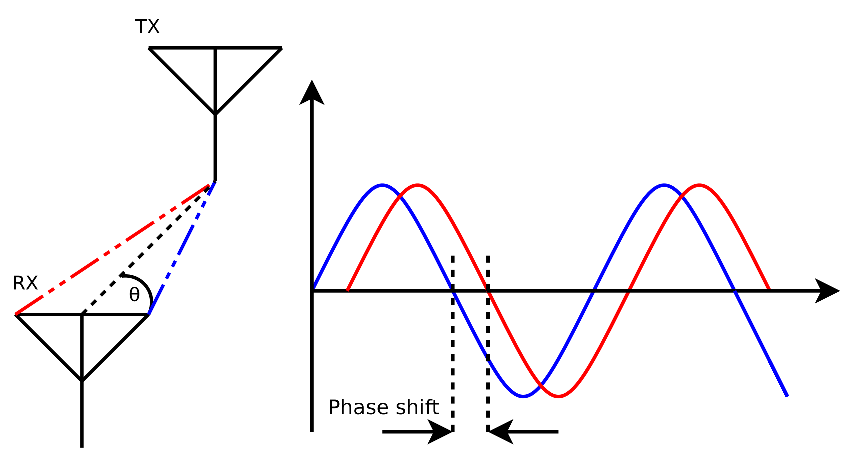


Figure 2.6: *In the figure to the left a transmitter is sending a signal that is received by to antenna array elements. The right figure shows the corresponding signals received and the difference in phase, in this example the phase difference is 45 degrees and this is the AoA.*

2.3 Fusion

In order to determine the position of the MD, many techniques can be performed. These techniques are based on geometrical calculation given some measurements. In this section, three different positioning techniques will be described and a theoretical solution of the ideal case will be given. This will only be appropriate if the measurements used are perfectly correct.

2.3.1 Triangulation

Triangulation can be used to find the location of a user in a two or three dimensional space. In a 2D space, the triangle is formed by the user(MD) and two other known points(BSs), see Figure 2.7. Given the position of these known points, the distance between them can be calculated. This gives the length of one side of the triangle. And by measuring the AoA of the signal from the BSs to the MD the values α and β are found.

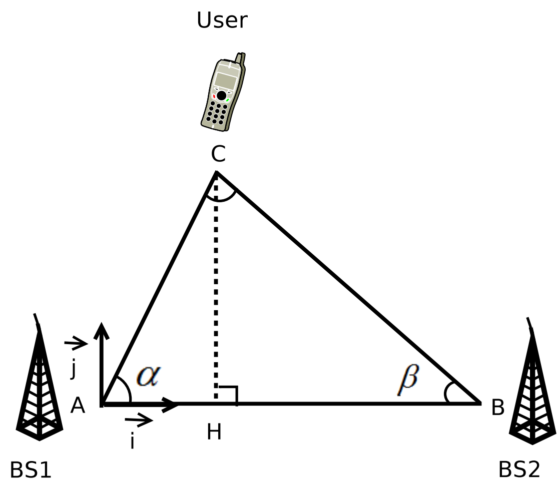


Figure 2.7: *The user is localized by calculating the distance HC given the distance AB (the position of the bases) and the angles α and β.*

As it will be demonstrated later, the localization of the user can be seen as calculating the distance CH (H being the orthogonal projection of C on the straight line (AB)).
Known values :

$$HB = HC \cdot \tan(\beta) \quad and \quad HA = HC \cdot \tan(\alpha) \tag{2.12}$$

Whereas :

$$AH + HB = AB \tag{2.13}$$

Thus :

$$\mathrm{HC} = \frac{\mathrm{AB}}{\tan(\alpha) + \tan(\beta)} \tag{2.14}$$

To give an example of positioning the user, the orthonormal coordinate system $\left(A, \vec{i}, \vec{j}\right)$ is considered, see Figure 2.7. The coordinates of the user are calculated using these relations :

$$x_c = AH = \frac{HC}{\tan(\alpha)} \quad and \quad y_c = HC \tag{2.15}$$

Therefore, it is possible to locate the MD by triangulation using the measurement of the AoA from two BSs. Another measurement of azimuth allows the positioning in a 3D space. The problem of using AoA measurements is that unless the propagation path is LOS the signal can be reflected which causes shadowing giving an improper indication of the signal direction.

2.3.2 Trilateration

Trilateration is a positioning technique which uses the distance of the unknown point from other reference points known and non-collinear. In a 2D space, three reference points are needed. Given the distance of the user from each reference point, it is known that the user is localized on three circles. Each circle has a reference point as a center and the corresponding distance as a radius. The intersection of two circles gives two possible points and the other circle leads to a unique point. This will be demonstrated below. The same principle can be used in a 3D space, but in this case it is a matter of spheres instead of circles and another reference point is needed. In the following an analytic resolution of this problem (in a 3D space) will be given.

Initially, three spheres formed as described above are considered. To simplify the problem, it will be resolved in a coordinate system where the $z = 0$ plane contains the centers of these three spheres, see Figure 2.8. One center will be the origin, another will be on the x-axis. Such a reduction is allowed as far as any set of three disjoint and non-collinear points can fit with these constraints. Therefore the position of the point, in any other coordinate system, can be calculated once it is found in this specified coordinate system.

- Let $M(x, y, z)$ be the position of the user.
- $O_1(0, 0, 0)$ the 1^{st} BS.
- $O_2(x_2, 0, 0)$ the 2^{nd} BS.
- $O_3(x_3, y_3, 0)$ the 3^{rd} BS.

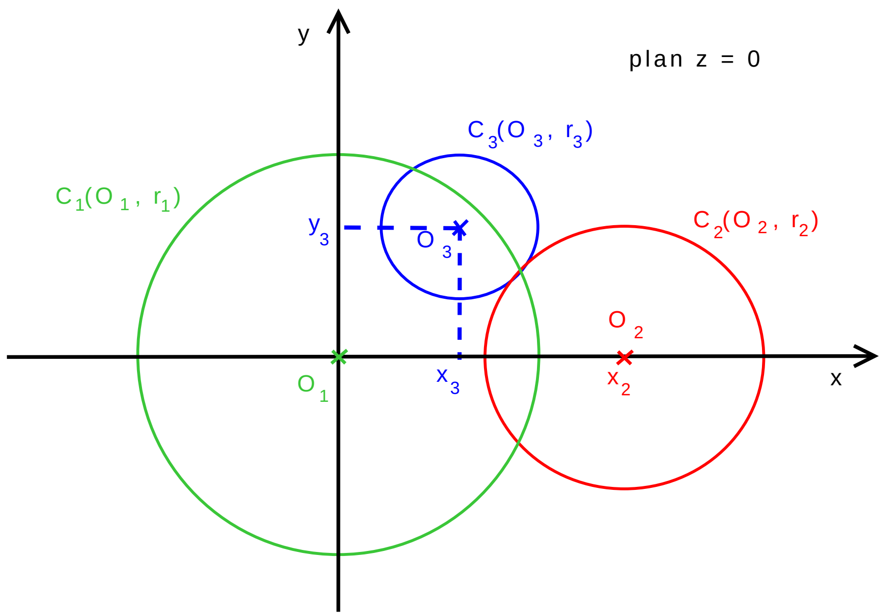


Figure 2.8: *Each center O_i is a BS with the corresponding distance r_i from the user. The intersection of three spheres gives one point in the plane $z = 0$, thus a fourth sphere is needed for a 3D positioning.*

In this specified coordinate system, the fact that : $M \in C_1 \cap C_2 \cap C_3$ implies :

$$
\begin{aligned}
&x^2 + y^2 + z^2 = r_1^2 \\
&(x - x_2)^2 + y^2 + z^2 = r_2^2 \\
&(x - x_3)^2 + (y - y_3)^2 + z^2 = r_3^2
\end{aligned}
\tag{2.16}
$$

By substitution the following is obtained:

$$
\begin{aligned}
&x = \frac{r_1^2 + x_2^2 - r_2^2}{2x_2} \\
&y = \frac{r_1^2 + y_3^2 + x_3^2 - 2xx_3 - r_3^2}{2y_3} \\
&z^2 = r_1^2 - x^2 - y^2
\end{aligned}
\tag{2.17}
$$

The obtained value of z^2 can be either zero, positive or negative. Therefore there is three possible cases:

1. If this value is zero, then there is a unique solution ($z = 0$). The 4^{th} BS is useless in this case.
2. If It is positive, then there is two solutions $z = \sqrt{r_1^2 - x^2 - y^2}$ or $z = -\sqrt{r_1^2 - x^2 - y^2}$. Thus, a fourth BS is needed.

3. If it is negative, then there is no solution. This means that the measurement of the distances r_i was inaccurate. If this is the case, the coordinates will be estimated by including an error model, see Section 6.2.2.

2.3.3 Multilateration

The trilateration technique requires distances between the BS and the MD. If it is performed by measuring e.g. the ToA, it requires synchronization between the BSs and the MD. The multilateration, also known as hyperbolic positioning, is an alternative which uses the measurements of the TDoA, see Section 2.2.3, therefore only the BSs have to be synchronized and the MD must have only a stable clock without synchronization.

Like the trilateration, the multilateration demands three BSs for positioning in a 2D space. From two BSs, one value of TDoA is obtained. The MD is then located onto a hyperbola. Considering a third BS, a second value of TDoA is obtained and then the MD is located onto another hyperbola. The intersection of these two hyperbolas leads to one precise position, see Figure 2.9. In a 3D space, it is a matter of hyperboloids instead of hyperbolas and a fourth BS is needed to find the 3D position.

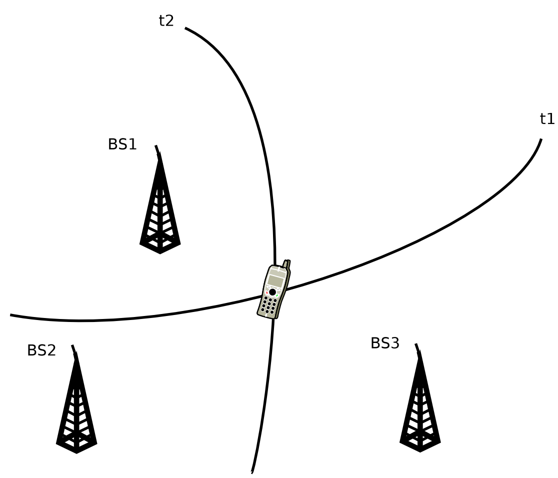


Figure 2.9: *By measuring the TDoA, hyperbolas can be formed to find the position of the user.*

The same simplification of the coordinate system as in the trilateration solving will be done.

- Let t_i be the ToA of the signal from the BS i.
- t_d the time when the signals were sent from all the BSs (it has to be the same this implies the necessity of synchronization between the BSs)
- $t_{i,j}$ the TDoA between a signal coming from the BSs i and j.

- d_i the distance between the BS i and the user.
- and c the the speed of light.

Known values are:

$$\begin{aligned} t_{i,j} &= t_i - t_j \\ &= (t_d + \frac{d_i}{c}) - (t_d + \frac{d_j}{c}) \\ &= \frac{d_i - d_j}{c} \end{aligned} \tag{2.18}$$

Therefore :

$$\begin{aligned} t_{2,1} &= \tfrac{1}{c}(\sqrt{(x-x_2)^2 + y^2 + z^2} - \sqrt{x^2+y^2+z^2}) \\ t_{3,1} &= \tfrac{1}{c}(\sqrt{(x-x_3)^2 + (y-y_3)^2 + z^2} - \sqrt{x^2+y^2+z^2}) \\ t_{4,1} &= \tfrac{1}{c}(\sqrt{(x-x_4)^2 + (y-y_4)^2 + (z-z_4)^2} - \sqrt{x^2+y^2+z^2}) \end{aligned} \tag{2.19}$$

Where (x, y, z) are the coordinates of the MD and (x_i, y_i, z_i) the coordinates of the BS i. Each of these equations represents the equation of an hyperboloid. The intersection of three hyperboloid gives the position of the user.

2.3.4 Error Model

When the values of the magnitudes, such as distance, AoA etc., include errors, using these theoretical solutions directly is not appropriate. In the trilateration technique for example, in most cases, the intersection of the three circles in a 2D space, will not lead to a unique point. Since these values are obtained by measurements, they will inevitably include errors. For example, the distance could be given by measuring the signal strength.

One of these techniques will be chosen further, and then a solution considering an error model and based on the ideal case solution will be described in Chapter 6.

2.4 Comparison

In this section the different measurement types for localization based on radio signals are discussed and compared in order to choose the most suitable method for the given scenario.

Received Signal Strength
As described previously, trilateration can be used to find the position using RSS in two dimensions. To perform trilateration at least three BSs must be used and the RSS measurements must be obtained shortly but not necessarily instantly after each other to have an accurate

position. The quality of the signal strength can be degraded due to obstacles, walls etc. that causes imprecise measurements. According to [Fig08] RSS is most suitable for positioning in a short range scenario.

Time of Arrival
Trilateration can also be used to find the position with use of ToA. Three BSs are needed to find the position in two dimensions and they all need to be synchronized with the MD in order to measure ToA correctly. To have an accurate position the timing on each device needs to be very precise and in a high resolution, because a small change in time can introduce large errors in calculation of the position. ToA is most suitable in a long range scenario [Fig08].

Time Difference of Arrival
In hyperbolic positioning the position is found by using TDoA. This method as well requires three BSs to find the position of the MD in two dimensions. The measurements are performed on the BSs and therefore they need to be synchronized. The MD does not need to be synchronized. For TDoA the timing is also very important and need to very precisely as it is the case for ToA. TDoA also shares the same properties as ToA and is therefore also most suitable in a large range scenario [Fig08].

Angle of Arrival
Two AoA measurements are used in the triangulation to find the position in two dimensions and three are needed in the other methods. This is due to the fact that the AoA of the signal from the MD to the BS and not the distance is calculated.

There is no need for synchronization but the measurements must, like in RSS triangulation, be obtained sequentially to have a precise position. The calculation of AoA is dependent on a LOS between the MD and the BSs, therefore obstacles and walls have a major impact on the measurements. To perform measurements special multi array elements antennas are needed to calculate the directions and these antennas are not widely available. As pointed out in [Fig08] AoA is also most suitable in large range scenarios. The comparison is summarized in Table 2.1.

	RSS	**ToA**	**TDoA**	**AoA**
Synchronization required	No	Yes BS and MD	Yes all BS	No
Precise timer needed	No	Yes	Yes	Yes
BS for 2D positioning	3	3	3	2
Signal reflection influence	High	Medium	Medium	Very high
Specific antenna	No	No	No	Yes
Best for short/long range	Short	Long	Long	Long

Table 2.1: *The different methods of finding the position is compared, which shows the advantages and disadvantages of each method.*

2.4.1 Conclusion of the Comparison

As described in this section there are both advantages and disadvantages of the different measuring methods. ToA requires synchronization between all devices and very precise timers in order to provide an accurate position, while TDoA has the benefit of only requiring synchronization between the BSs. This gives the flexibility of having an simple device acting as a MD while the demands for complexity is higher for ToA. The AoA needs a specific antenna, but has the advantage of requiring only 2 BSs for positioning in two dimensions. ToA, TDoA and AoA are most suitable for long range scenarios while RSS is best for short range. For more complex systems the methods can also be combined providing a more accurate position. The best method of positioning depends on the available equipment, technology and chosen scenario.

The given scenario is indoor positioning with Bluetooth which gives some limitations. In an indoor environment there will be obstacles that can have influence on signal path, therefore AoA is not the best solution since it demands on LOS and also a specific multi array antenna that is not available on the Bluetooth equipment provided by CSR. Precise timing and synchronization will be hard or impossible to implement with the equipment and since ToA and TDoA will need these properties they are not suitable for the project. ToA, TDoA and AoA are also better for large range scenarios while the scenario for this project can be considered to be a relatively short range scenario. Based on this it is decided to use RSS and trilateration to find the position for the indoor scenario using bluetooth equipment provided by CSR.

Chapter 3

Positioning System Issues

In Chapter 2 various methods for positioning systems have been described, but there are other issues which should be considered when developing a positioning system and do not fit in the layered reference model. These subjects are related to e.g. synchronization and computation issues. In this chapter these issues are described and some research results for Bluetooth positioning are mentioned. These are done by João Figueiras [Fig07] to address the problems of using the Bluetooth inquiry phase for positioning. It will also be decided whether to place the computation of the position in the network or on the MD.

3.1 Location Information in Bluetooth

In this project a localization system is build on top of an existing wireless technology, namely Bluetooth. Compared to designing such a system based on basic RF hardware, it is necessary to consider timing aspects in the existing protocols. Fluctuating latency of measurements can introduce large errors in the resulting position if the target moves. When using RSS measurements for positioning it is also important that the TSS is always known so the difference can be calculated. Bluetooth and other short range wireless technologies implements power control mechanisms mainly to save battery power, but also to decrease interference for other devices in range.

To solve the latency problem, the location information can be obtained when a connection between the target and the BS is established. This way the positioning system can get fast results from target through the established connection. The measurements however, are most likely influenced by the power control mechanisms of Bluetooth which makes them useless for determining a position. The solution to this problem could be to request the TSS from the target, alternatively the location information can be obtained at times where the power control mechanisms are not running. In Bluetooth, power control is not running in the inquiry phase, where a device is searching for other devices to connect to.

In the *inquiry substate*, the master who wants to connect to a slave, sends out Identity packets at different frequencies. In order for the slave to be connectable it must be in the *inquiry scan substate* listening at frequency f_i at the same time as the master transmits on frequency f_i. The slave can then send a Frequency Hop Synchronization (FHS) packet as response to the master. The response time is a random process as the the frequencies for transmitting Identity packets at the master and scanning for them at the slave, are chosen at random.

João Figueiras [Fig07] has simulated the inquiry procedure for N masters running inquiry of one slave. The result of the simulation for four different Ns can be seen in Figure 3.1. As expected the latency of inquiry increases as the number of masters increases. The interesting results are for $N \geq 3$ as 3 or more masters (BSs) are needed for trilateration in this project. It is seen from the CDF in Figure 3.1 that it takes approximately 4-5 seconds before all inquiries are complete with 90 % confidence. It can be concluded that introducing more BSs in order to enhance accuracy, also increases the latency in getting the measurements.

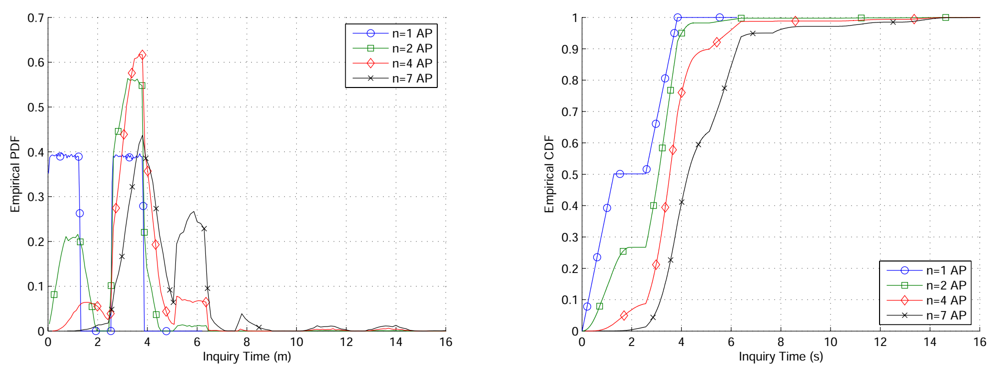


Figure 3.1: *Inquiry time for {1,2,4,7} masters (AP) inquiring one slave. Left: PDF. Right: CDF [Fig07]*

From this it is chosen to build a prototype with three BSs to keep it as simple as possible and avoid increased latency.

3.2 Localization Processing

According to the decision made en Section 2.4, two different approaches could be considered when thinking about indoor positioning. Either the user needs to have his own position using a MD, or there is a surveillance base which wants to track some MDs. These two scenarios lead to two different types of performing the positioning, the network based technique and the device based one. It is a matter of where the localization computation is done, on the MD or in a controller of the network. But it is also possible to use each technique for both scenarios by sending the position from the network to the MD and vice versa.

3.2.1 Device Based Technique

In this case, the calculation and the measurements are done in the user's MD. As the positioning is performed by trilateration using the signal strength, the MD will need at least three measurements, one from each of three BSs in a 2D space. Then the computing will be done to determine the current position of the MD, see Figure 3.2.

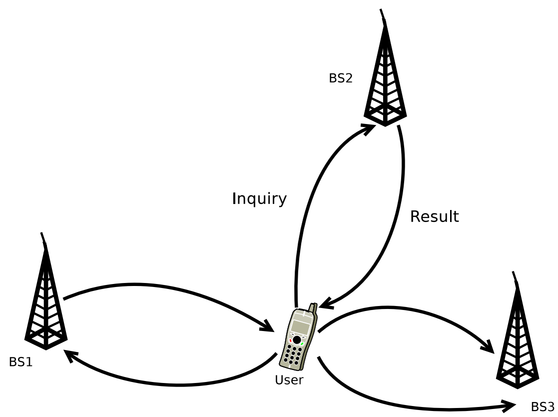


Figure 3.2: *Device based localization technique.*

In order to get the measurement, the MD needs to know the BSs in its range and their positions. This can be done either:

- By performing an inquiry and using the Service Discovery (SD) each time a position is calculated, see Appendix A.

- Using a list of all the BSs with their MAC addresses, then the MD will need to select those which are in its range from this list at a certain moment.

This technique requires the installation of software on the MD. Thus the software must handle the different operating systems of the MDs. Typically, this would only be possible using a smart phone such as one based on Symbian or Windows Mobile. Otherwise the installation must be done by the manufacturers.

3.2.2 Network Based Technique

In this case, the calculation and the measurements are done in a controller which is part of the network. Given the MAC address of the user's MD, the controller gets the signal strength measurement from all the BSs at approximately the same time. Thus, each BS is connected to the controller. However, if there are a great number of BSs then it is possible to use the network

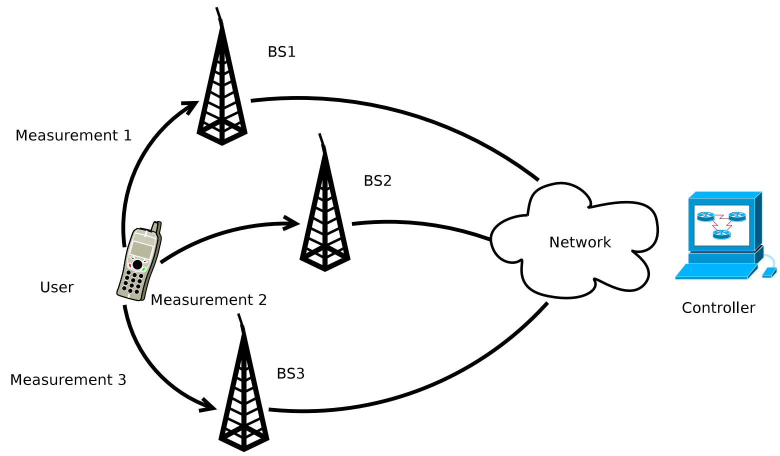


Figure 3.3: *Network based localization technique.*

if it already exists or build one if not. E.g. in an office environment a network of WLAN APs is likely to exist and these can be utilized as BSs in a positioning system, see Figure 3.3.

The controller will contain the list of all the BSs with their positions. Using these positions and the measurements, the controller will localize the user. An PC can be used as a controller. It will run software which will be in charge of doing the computation and getting the measurements.

3.2.3 Comparison

From the description of these two approaches, the main difference which appears is that in the mobile based technique, the computation will be done in the MD instead of a computer in the case of the network based technique. Thus, the device based technique will introduce constraints on the MD. However, if these constraints could be overcome, then the mobile based technique will provide a more distributed computing system. Still, it will not be an important advantage as long as a computer is able to handle the computing for an significant number of MDs.

Another difference is that in the mobile based technique, the MD will receive the measurements only from the BSs in its range. Whereas in the network based technique, the controller will receive measurements from all the BSs of the network, and then selected the most suitable ones. This selection could be a problem for the computation if the system is composed of a great number of BSs.

Also, in the device based technique, the MD is doing not only the computation, but the three measurements as well. Whereas, in the network based technique, the measurements are done by the BSs, so they could be done at the same time and then the measurements will be obtained faster.

These differences are summarized in the table below.

Device based	**Network based**
Requires software running on the MD	Any Bluetooth MD can be tracked
Distributed computing (scalable, low cost) when there are several users	Centralized computing, the calculations of the positions are done in one controller
Only response from BSs in range	Multiple inquiries results (could be resolved by a prediction model)
Multiple measurements by one MD	One measurement by each BS at the same time

Based on this it is chosen to perform network based tracking because any MD such as a normal mobile phone can be tracked without installing any software on that device. Also the prototype in this project will for simplicity only track one device and scalability will not be an issue.

Chapter 4

Problem Statement

In Chapter 1 to 3 issues regarding positioning have been investigated. The Location Stack proposed by Jeffery Hightower [Hig02] has been introduced and methods to measure distance or direction between a Mobile Device (MD) and a Base Station (BS) have been discussed as well as how this information can be used to find the position. For indoor positioning signal propagation has been described when obstacles or walls that alter the signal propagation is introduced to the environment.

Based on this analysis it has been chosen to perform indoor network based tracking of a MD by measuring Received Signal Strength (RSS) from different BSs and performing trilateration on the measurements. The aim of the project is then to design and implement a system that makes this possible via the Bluetooth equipment provided by CSR. A model that considers the movement of the user will also be implemented to improve the accuracy of the tracking system.

The main objectives in this project are to:

- Design a network based localization system in two dimensions
- Track the position of a MD based on RSS and trilateration
- Enhance the precision accuracy of the tracking using an enhanced techniques such as a mobility model and a filter
- Implement a prototype using the Bluetooth equipment from CSR
- Test and verify the implementation
- Discuss and compare the results of the simple and enhanced tracking techniques

In the following chapter the requirements and interfaces for the system will be described which will lead to the system design.

Chapter 5

Requirements Specification

In this chapter the requirements for the system and prototype will be specified. The specification will be divided into two sections describing the general system requirements and prototype requirements for the system and the various subsystems. After this a description of the system interfaces will be given.

5.1 Functional Requirements

1. The system must be able to compute the location of a MD in a two dimensional indoor environment.
2. Bluetooth must be used as wireless technology.
3. BSs must be used to measure RSS to compute the distance between MD and BS.
4. Trilateration must be used to compute the position of a MD.
5. The computation of the position must be done centralized on a control unit.
6. The position of the MD must be shown on a map.
7. The accuracy of the position must be in the order of 1 meter.
8. The computation of the position must take the movement pattern of the MD into account as well as the history of previous positions and noise on the measurements.

5.2 System Prototype Requirements

1. The system must consist of:

- Three BSs supporting Bluetooth class 2.
- One controller (PC) connected to the BSs by USB cables.

2. The system must be able to track one MD supporting Bluetooth class 2.
3. The positioning must be done inside a 240x240cm square with BSs in three of the corners forming a triangle.
4. The accuracy of the position in the system prototype must be in the order of 30 cm.

5.2.1 Base Station

1. The BS must be able to do an inquiry for all Bluetooth devices in range.
2. The BS must be able to measure RSS of a specific MD specified by the Bluetooth MAC address.

5.2.2 Controller

1. The controller must request and collect measurements from the BSs.
2. The controller must be able to compute the position of the MD based on RSS.
3. The position of the MD must be shown on a map in a GUI.
4. The map must be updated each time a new position is computed.
5. Each position must be stored as coordinates (x,y).
6. The precision of the map must be 5cm.

5.3 Interfaces

In this section the interfaces in the system will be described. In the system there is a graphical user interface, an interface between the controller and BSs and an interface between the BSs and the MD.

5.3.1 GUI

When the user is starting the application he is presented to a screen containing a map of the system. It is possible to scan for MDs in range of the system, after this scan is completed a MD can be selected from a list to be tracked. The user will always have the opportunity to stop or restart tracking. In Figure 5.1 an example of the GUI can be seen.

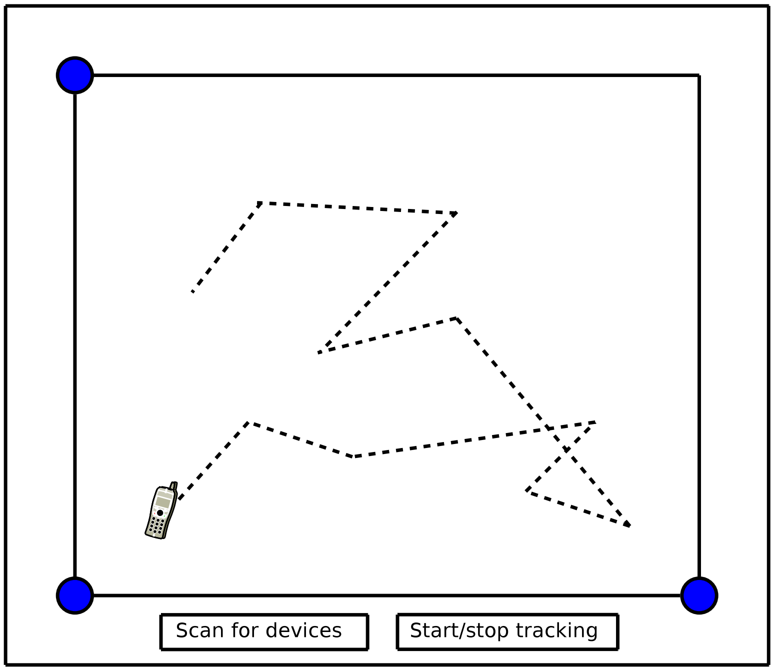


Figure 5.1: *The GUI of the application where a MD is tracked and the tracking history is shown.*

5.3.2 Controller - Base Station

Figure 5.2 shows how the controller and BS are communicating. The communication flow between the controller and the BS will be as follows:

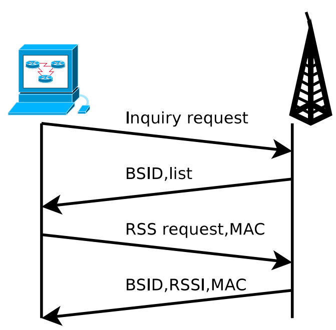


Figure 5.2: *The communication flow between the controller and a BS*

The controller will send an inquiry request to the BS. After the BS has performed the inquiry it will return its ID and a list of MDs discovered during the inquiry. When the controller wants to track the position of a certain MD a request for RSSI measurement and the MAC address of the MD is sent to the BS. The BS will return its ID, the RSSI measurement and the MAC address of the MD.

5.3.3 Base Station - Mobile Device

Between the BS and the MD the communication flow will be as follows:

The BS will perform an inquiry to discover all MDs in range. If it is in inquiry scan mode, the MD will respond to the inquiry. The flow of communication can be seen in Figure 5.3.

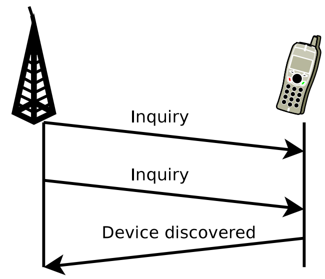


Figure 5.3: *The communication flow between the BS and a MD*

Chapter 6

System Design

In this chapter the design of the prototype will be outlined. The design is organized into two subsystems: Base Station and Controller. In Figure 6.1 a deployment diagram of the system is shown, where the system is split into two subsystems; the GUI, controller and BS. The user and the MD are actors to the system where the user will ask the system to track the MD. In the prototype there will be one controller serving the user with a graphical user interface and calculating the position as well as three BSs measuring RSS from the MD.

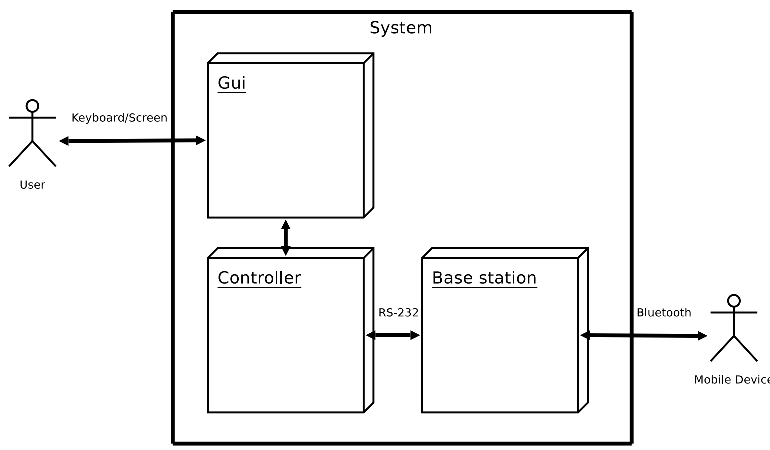


Figure 6.1: *Deployment diagram showing the actors and subsystems of the system*

Like the mechanisms of positioning described in Chapter 2, the design is also organized according to the Location Stack. Figure 6.2 shows the Location Stack reference model and a system stack of the prototype.

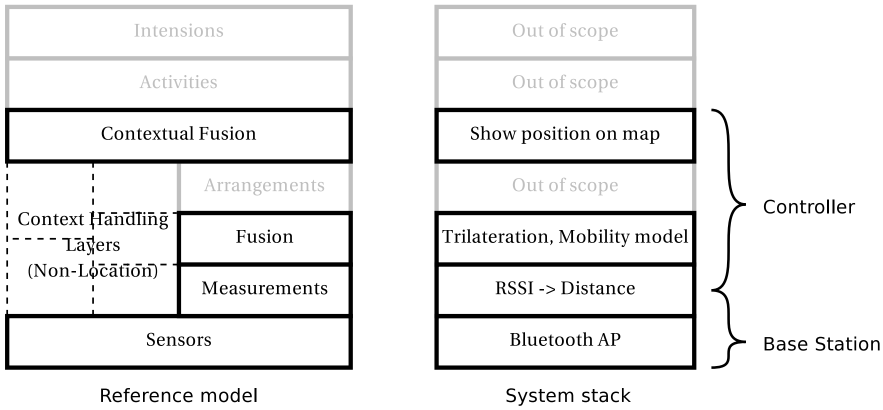


Figure 6.2: *The Location Stack (Reference model) compared to the system stack of tasks for the deigned prototype. The gray layers are not considered in this project.*

The BS is a Bluetooth AP responsible for obtaining RSSI measurements from the MD. Thus the BS covers the Sensors Layer and the Measurements Layer. The controller is a PC and handles the positioning calculations from converting RSSI to distance, to showing the position to the user on a map. Thus the controller covers the Measurements, Fusion and Contextual Fusion Layers. The layers Arrangements, Activities and Intentions are out of scope of this project. These layers deal with special usage of the positioning and will not be considered further. In the two following sections the subsystems will be described.

6.1 Base Station

In this section the activity diagram of the BS is described. The activity diagram which is shown in Figure 6.3 handles the requests sent from the controller, when the BS is in the idle state. Each received request is handled by the `Request from the controller`-function by forwarding the incoming requests to the corresponding subfunction. Those subfunctions are: `Do inquiry`-function which is activated by `Request for devices` and `Get RSSI`-function which is accessed by `Request for RSSI`. Each of these functions are described in the following:

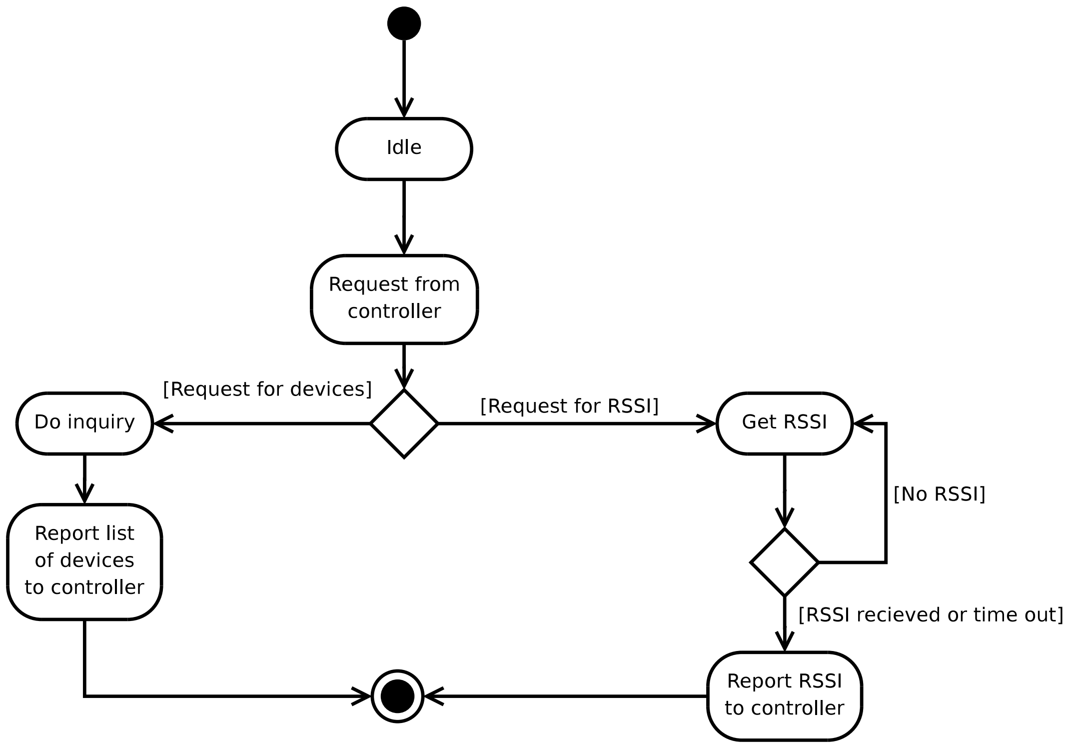


Figure 6.3: *Activity diagram of the BS*

Do Inquiry:
The objective of this function is to collect information about the available Bluetooth devices in range of the BS and report them back to the controller.
The `Do inquiry`-function handles this by calling the SD-module, which is located on Blue Core Host Software (BCHS), see Appendix A. The SD module contains an internal list with information regarding remote devices based on an inquiry. This information could be for example the MAC address or the name of a particular device. Upon a request from the `Do inquiry`-function, the SD will return the address of the available Bluetooth devices. A sequence diagram can be seen in Figure 6.4, which shows the interaction between the `Do inquiry` and SD.

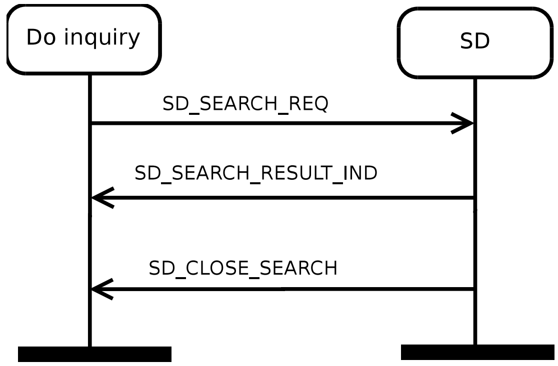


Figure 6.4: *Device search sequence. The SD returns an indication each time it finds a device. When the search timer runs out or the SD has returned the address of a specific device, a close search indication will be sent to the Do inquiry-function.*

Request for RSSI:
The RSSI value of a MD is retrieved from the result of an inquiry initiated by the controller. The request for RSSI will contain the MAC address of the specific MD, if the result of the inquiry contains a RSSI measurement for the MD this value is returned. This subfunction uses the Sensor Layer of the location stack described in Chapter 2.

6.2 Controller

In this section the design of the controller will be described. It serves a graphical user interface where the user is able to track a MD. The controller is then able to issue requests to the BSs and collect measurements that are utilized to compute the position.

The activity diagram of the controller can be seen in Figure 6.5.

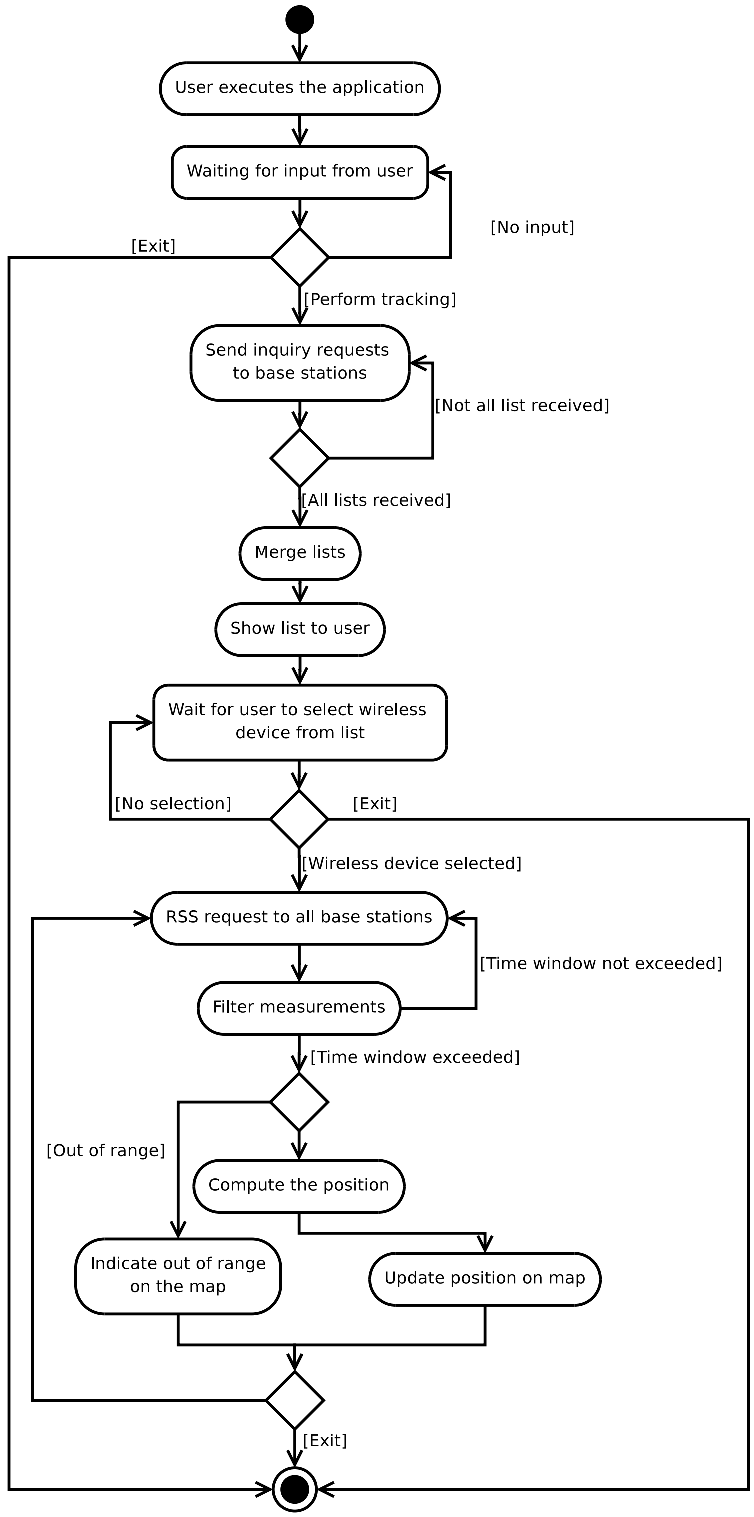


Figure 6.5: *Activity diagram of the controller*

User Executes the Application :

When the user executes the application, a graphical user interface with an empty map is presented. Here it is possible to select `Search for MDs` or `Exit`.

Waiting for Request from User :

The application is waiting for the user either to search for MDs or to exit the application. Depending on the choice the application will `Perform tracking` or `Exit`.

Send Inquiry Requests to BS :

An inquiry request is sent to each BS and the controller is waiting for the result of each inquiry. The results will be a list from each BS with discovered MDs.

Merge Lists :

A list is created from the received results. If a MAC address of a MD is listed more than once, the duplication will be deleted.

Show List to User :

The list of MDs will be presented to the user. It will contain the MAC address, name and type of each one.

Wait for User to Select the MD from List :

The application is waiting for the user either to select a MD from the list or to exit the application.

RSS Request to All BSs :

When a MD is chosen, a request for RSSI with the corresponding MAC address is sent to all BSs simultaneously. The application will wait for the BSs to perform inquiries and return the RSSI measurements. If the RSSI value indicate that the MD is out of range this will be indicated on the map. The time for performing all inquiries and getting the RSSI is defined as the Inquiry Period, where the starting point is when the request to all BSs is sent, and the end point is when the last inquiry result is received. The Inquiry Period is a random variable distributed as shown in Figure 3.1.

Filter Measurements :

It is impossible to get measurements from all the BSs in the same time and as long as the Inquiry Period is a random variable, it will be different each time. This filter is a protocol which will assign meaningful values of RSSI measurements from all the BSs at specific time instants. This will be described further in Section 6.2.1.

Compute the Position :

Using the values from the measurements, the coordinates of the MD on the map are computed by an algorithm described in Section 6.2.2.

Update Position on Map :

The new position will be shown on the map by a star and a line will be drawn from the previous position. The previous positions will remain on the map.

Indicate Out of Range on the Map :

The dot representing the MD will disappear from the map, but the previous positions indicated by the line will remain.

The Sections 6.2.1 and 6.2.2 will describe in detail the measurements filtering and how to compute the position properly from noisy measurements.

6.2.1 Measurements Filtering

The activity `RSS request to all base stations` from the activity diagram of Figure 6.5 is designed as a set of three measurements from the BSs, one for each base station. Due to the BCHS controlling the AP being out of scope of this project, it has been chosen to look upon the AP device as a black box providing sequentially RSSI values of a specific MD. The downside of this approach is that it may take a longer time for the MD to respond to the inquiry of one BS if it must respond to continuous inquiries from the other BSs. An additional overhead is also introduced as the measurements occur in bursts and some measurements may contain the same information. Alternatively the extra measurements can be used to average out some of the noise in the measurement.

The RSSI measurements from the BSs must be filtered in order to extract only new measurements which are simultaneously obtained from all three BSs. I.e a distance from the MD to all BSs must be available in order to calculate the position at any given time. This is achieved by means of a filtering algorithm which is executed in the Fusion Layer prior to the computation of the position. The algorithm consists of two steps.

Step 1 - Measurement grouping

1. Get measurements
2. Define time window
3. Compute mean RSSI within the window

The mean of the RSSI measurements m within time window of length w is computed i.e. $\forall m \in [t_n; t_n + w]$. Where w must be chosen dependent of the speed of the tracked MD and the rate of measurements. The details of step 1 are shown in the activity diagram of Figure 6.6.

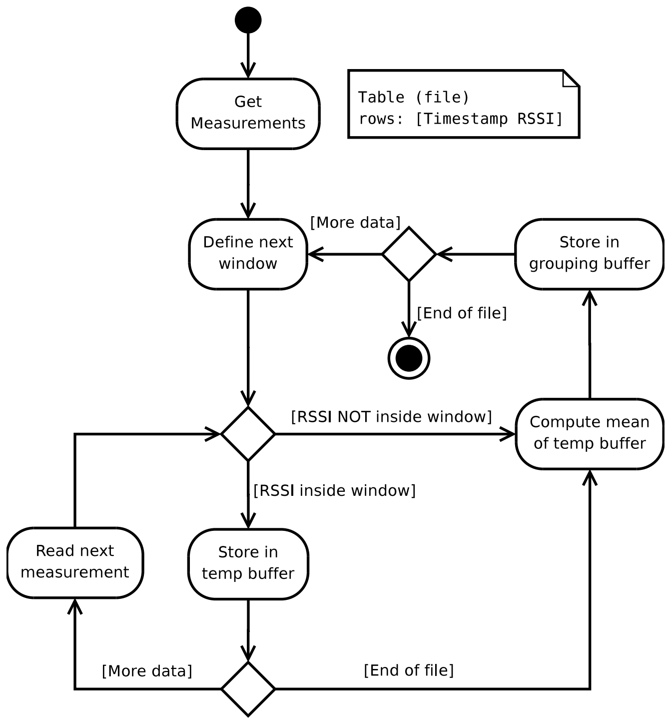


Figure 6.6: *Filter step 1: Grouping of measurements*

Step 2 - Comparison of BS results This step will discard measurements in time instances which does not contain measurements from all BSs. The following is executed.

1. Get grouped measurements
2. Compare grouped RSSI values for all BSs
3. Discard groups with missing RSSI value for any BS

The details of step 2 are shown in the activity diagram of Figure 6.7.

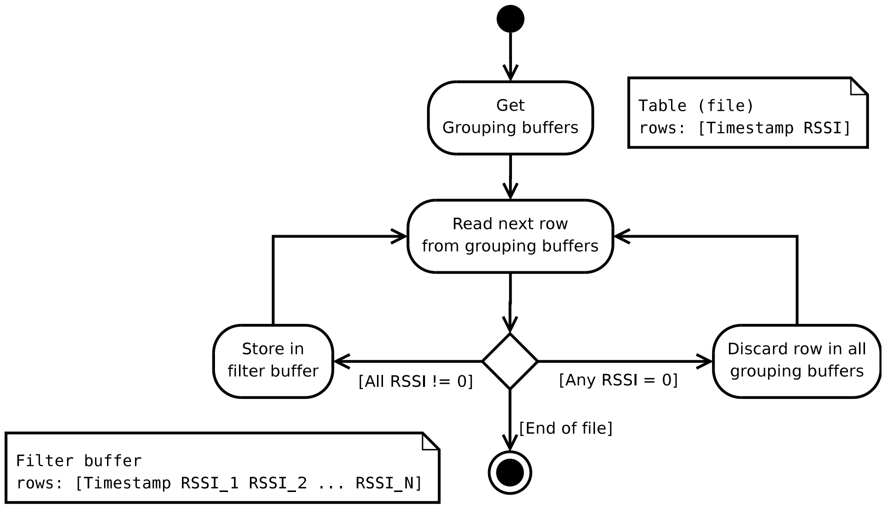


Figure 6.7: *Filter step 2: Comparison of BS results*

After running this algorithm only the time windows containing measurements from all three BSs are preserved. This way a position within each window can be computed. In worst case scenario only few windows with measurements are available and the position will be unknown outside these windows.

6.2.2 Computation of the Position

The computation of the position described in the activity diagram for the controller will be outlined in this subsection. The flow of the computation can be seen in Figure 6.8, where the `Convert to RSS` and `Convert to distance` functions are categorized under the measurement layer of the Location stack, the `Compute the position` part is the equivalent of the Fusion layer.

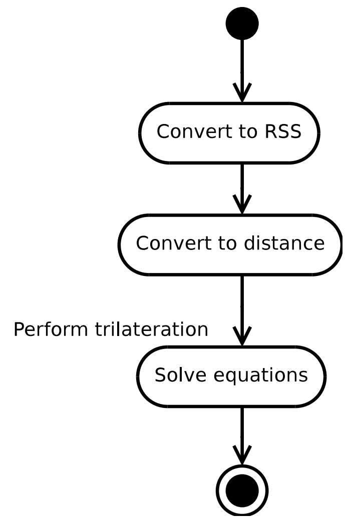


Figure 6.8: *The flow of the computation of position*

Convert to RSS

This section describes the first part of the measurement layer of the location stack where the received RSSI dB value from the BS, is mapped to RSS dBm.
This mapping of RSSI to RSS is based on the Golden Receiver Power Range which is the desired power level of the received signal, defined by the Bluetooth specification [BS07], see Figure 6.9. This is a property that is used by the Bluetooth Link Layer in order to optimize the power consumption of a transmitting device, described in Appendix A.

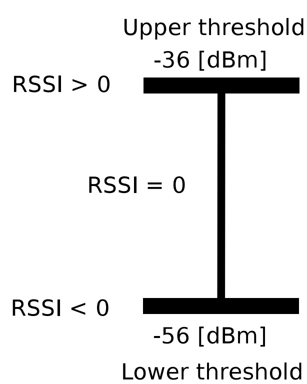


Figure 6.9: *The Golden Receiver Power Range.*

The lower and the upper threshold of the Golden Receiver Power Range are defined to be -56 dBm and -36 dBm with an accuracy of ± 6 dBm. A positive RSSI value, returned by the BS indicates how many dB the received signal power is above the upper limit and any negative value indicates how many dB the received signal power is below the lower limit of the Golden

Receiver Power Range. If the RSSI value is zero then it is assumed that the received signal power is inside the Golden Receiver Power Range. The returned RSSI value is then added to the threshold regarding the sign of the returned RSSI, e.g. if the returned value of the RSSI is -42 dB then the RSS dBm of this signal would be RSS = -56 dBm -42 dB = -98±6 dBm, Equation 6.1 shows the relation between the RSS and the RSSI.

$$RSSdBm = thresholddBm \pm RSSIdB \tag{6.1}$$

Where *threshold* = -56 dBm for negative RSSI and *threshold* = -36 dBm for positive RSSI.

Convert to Distance

This section describes the last part of the measurement layer of the location stack.
Given the received signal strength, the distances between the BSs and the MD is needed to find its position. The distance can be achieved by either using the deterministic approach, described in Section 2.2.4, or the probabilistic approach.
In the following these approaches are detailed.

Deterministic:

The deterministic conversion is obtained according to Section 2.2.4, where the relation between the received power and the distance is described as follows:

$$P_r dBm = P_0 dBm - 10\beta \log\left(\frac{d}{d_0}\right) \tag{6.2}$$

Where :

- P_0 is the transmitted power from the antenna
- d_0 is a reference distance for the antenna far field: at this distance the received power equals the transmitted one $P_r = P_0$
- β is the path loss exponent

The parameters can be obtained by an empirical measurement, see Appendix C. Given these parameters the distance can be computed using this equation:

$$d = d_0 \cdot 10^{\frac{P_0 - P_r}{10\beta}} \tag{6.3}$$

Probabilistic:

Apart from the deterministic model the distance between the BS and a MD can be achieved based on an estimate e.g. by looking at how frequent a specific RSS value is appearing for a given distance. This can be designed as a table consisting of the probabilities $c_{i,j}$ for an RSS

RSS/Distance	5 cm	10 cm	15 cm	...
-33 dBm	$c_{1,1}$	$c_{1,2}$	$c_{1,3}$	...
-34 dBm	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	...
-35 dBm	$c_{3,1}$	$c_{3,2}$	$c_{3,3}$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...

Table 6.1: *Contingency table*

value O_i for a given distance d_j ($c_{i,j} = P(O_i|d_j)$), an example of this is shown in Table 6.1, see Section 7.3 for more details.

Having such a table, the distance can be obtained from the RSS measurement by choosing the distance which maximize the probability for this given measurement.

Solve Equations

In this section the Fusion layer of the location stack is described. Given the distance from three different and non collinear BSs, the exact position (in a 2D space) of the MD can be calculated by trilateration in theory, see Section 2.3. As explained in Section 2.3, this computation has to handle the error introduced in the distance measurement. Considering the three circles having a BS as a center and the corresponding distance as a radius, in this case the circles do not cross each other in exactly one point contrary to the theoretical case.

- Let $\hat{d}_i$ be the measured distance between the MD and the BS ($i \in \{1, 2, 3\}$).
- $(0, 0)$, $(x_2, 0)$ *and* (x_3, y_3) the coordinates of BS1, BS2 and BS3 in this order, considering the same coordinate system described in 2.3.
- and $(\hat{x}, \hat{y})$ the estimated coordinates of the MD.

In the following, two different algorithms will be described to solve this problem. These algorithms are based on the trilateration algorithm in the ideal case, see Section 2.3.

Algorithm 1 :

Let $\hat{C}_i$ be the circle having the BS i as a center and $\hat{d}_i$ as a radius.
Considering the intersection of $\hat{C}_1$ and $\hat{C}_2$, three cases are possible :

- The intersection is exactly one point, see Figure 6.10:

$$\begin{aligned} \hat{x} &= \hat{d}_1 \\ \hat{y} &= 0 \end{aligned} \tag{6.4}$$

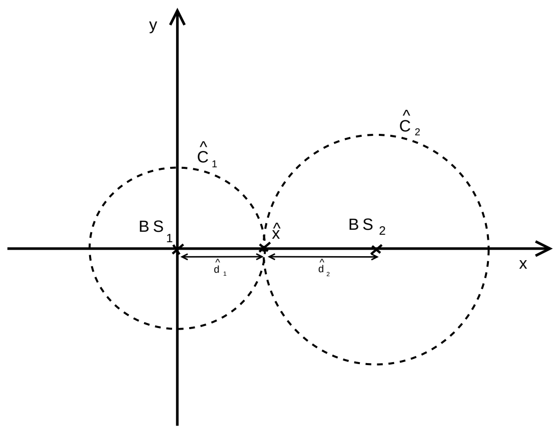


Figure 6.10: *Case 1: Illustrating the intersection of $\hat{C}_1$ with $\hat{C}_2$ at $\hat{x_1}$.*

This corresponds to the position of the MD.

- The intersection is two points, illustrated in Figure 6.11:

$$\hat{x} = \frac{x_2^2 + \hat{d}_1^2 - \hat{d}_2^2}{2x_2}$$
$$\hat{y} = \pm\sqrt{\hat{d}_1^2 - \hat{x}^2} \tag{6.5}$$

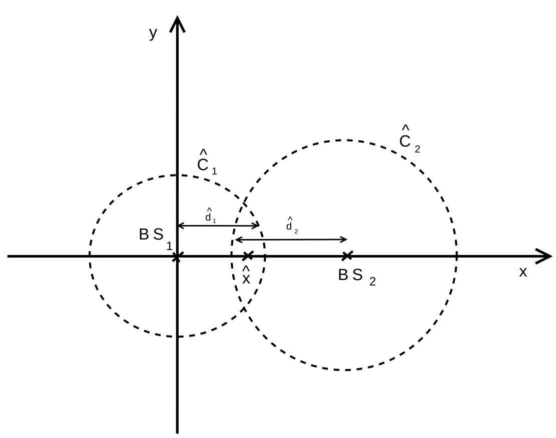


Figure 6.11: *Case 2: Illustrating the intersection of $\hat{C}_1$ with $\hat{C}_2$.*

Then the nearest point to the circle $\hat{C}_3$ will be chosen as position of the MD.

- The circles $\hat{C}_1$ and $\hat{C}_2$ do not cross each other, then three cases are possible, see Figure 6.12.
  - case 1 : if $x_2 - \hat{d}_2 \geq 0$ and $\hat{d}_1 \leq x_2$ then : $\hat{x} = \frac{1}{2}(x_2 - \hat{d}_2 + \hat{d}_1)$ and $\hat{y} = 0$.

- case 2 : if $x_2 - \hat{d}_2 \le 0$ and $\hat{d}_1 \le x_2$ then : $\hat{x} = \dfrac{1}{2}(x_2 - \hat{d}_2 - \hat{d}_1)$ and $\hat{y} = 0$.
- case 3 : if $x_2 - \hat{d}_2 \ge 0$ and $\hat{d}_1 \ge x_2$ then : $\hat{x} = \dfrac{1}{2}(x_2 - \hat{d}_2 + \hat{d}_1)$ and $\hat{y} = 0$.

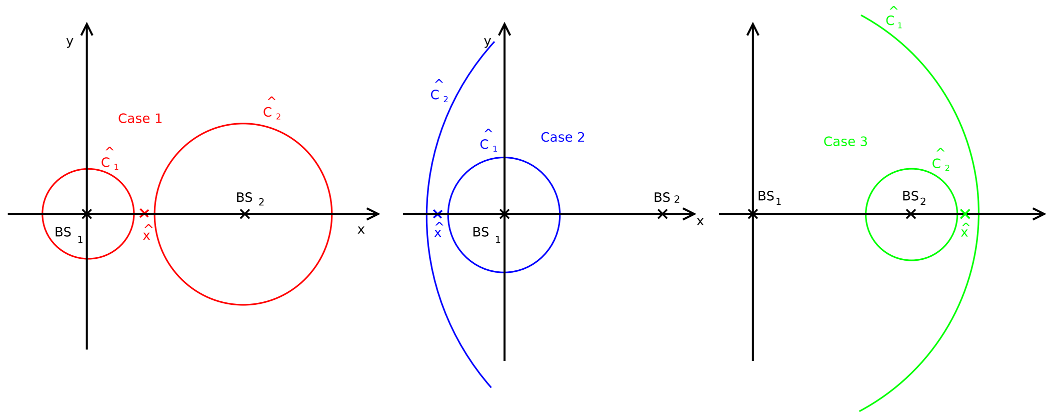


Figure 6.12: *Different cases of the intersection of* $\hat{C}_1$ *with* $\hat{C}_2$.

Algorithm 2 :

Assume that the measured distances are in the form: $\hat{d}_i = d_i + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, \sigma_i)$ (σ_i is a function of d_i) for $i \in \{1, 2, 3\}$.

Let c_{σ_i} be a constant > 0 and $P = \Pr\left(d \in [d_i - c_{\sigma_i}; d_i + c_{\sigma_i}]\right)$.
Then:

$$\begin{aligned} P &= \Pr\left(d \in [d_i - c_{\sigma_i}; d_i + c_{\sigma_i}]\right) \\ &= \Pr\left(-c_{\sigma_i} \le \varepsilon_i \le c_{\sigma_i}\right) \\ &= \int_{-c_{\sigma_i}}^{c_{\sigma_i}} \mathrm{f}_{\varepsilon_i}(x)dx \end{aligned} \tag{6.6}$$

Where $\mathrm{f}_{\varepsilon_i}(x)$ is the probability density function of ε_i which is :

$$\mathrm{f}_{\varepsilon_i}(x) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\dfrac{x^2}{2\sigma_i^2}} \tag{6.7}$$

Given this, the calculation of P leads to:

$$\begin{aligned} P &= \frac{2}{\sqrt{\pi}} \int_0^{\dfrac{c_{\sigma_i}}{\sigma_i\sqrt{2}}} \mathrm{f}_{\varepsilon_i}(x)dx \\ &= \operatorname{erf}\left(\frac{c_{\sigma_i}}{\sigma_i\sqrt{2}}\right) \end{aligned} \tag{6.8}$$

Where erf is the Gauss error function.
This means that if the needed probability of having $d_i \in [\hat{d}_i - c_{\sigma_i}; \hat{d}_i + c_{\sigma_i}]$ is α, then : $c_{\sigma_i} = \sigma_i\sqrt{2}\,\mathrm{erf}^{-1}(\alpha)$.

So with a high value of α (for example $\alpha = 0.9$), it is possible to consider that $d_i \in [\hat{d}_i - c_{\sigma_i}; \hat{d}_i + c_{\sigma_i}]$.

For $i \in \{1; 2; 3\}$ let :

- $\hat{C}_i$ be the circle which has BS_i as a center and $\hat{d}_i - c_{\sigma_i}$ as a radius.
- and $\hat{C}'_i$ be the circle which has BS_i as a center and $\hat{d}_i + c_{\sigma_i}$ as a radius.

Thus the MD is localized in the area between the circles $\hat{C}_i$ and $\hat{C}'_i$ for each $i \in \{1; 2; 3\}$, see Figure 6.13.

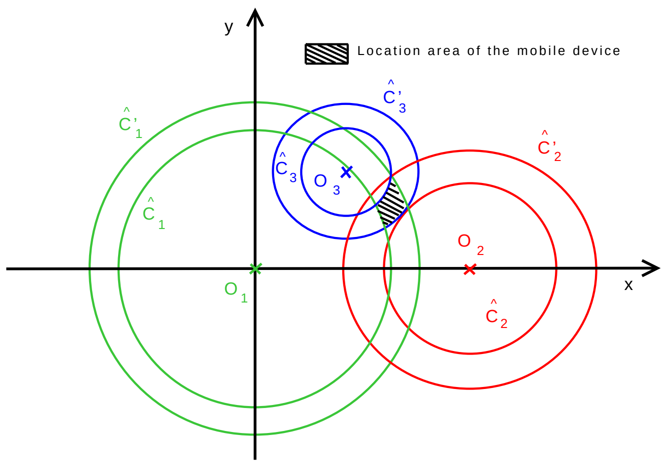


Figure 6.13: *Intersection of the 3 rings areas gives the possible location of the MD.*

The intersection of these 3 areas, which exists because $\forall i \in \{1; 2; 3\}$: $d_i \in [\hat{d}_i - c_{\sigma_i}; \hat{d}_i + c_{\sigma_i}]$, determine a possible location for the MD. Thus, the centroid of this area can be considered as the most suitable position of the MD.

Important Remark: As it is demonstrated above, the higher the value of α is, the more probable it is to consider that: $d_i \in [\hat{d}_i - c_{\sigma_i}; \hat{d}_i + c_{\sigma_i}]$. But when α approaches 1, this interval becomes larger and the accuracy of the trilateration decreases.

Summary and Enhancement

The solution described above is a first approach for dealing with the errors appearing in the measurements. But, it is a method which does not consider the mobility history of the MD. There are methodes which take into consideration the previous states of the system (in this case the state will be the kinetics of the MD) such as the Particle Filter or the Kalman Filter. A possible solution where the Less Drunk Model is used with the Particle Filter to improve the positioning, will be analysed and designed in the following chapter. This new approach will be different from the one designed above only in the `Measurements Filter` and `Compute Position` functions.

Chapter 7

Enhanced Positioning using the Particle Filter and the Less Drunk Model

In the System Design in Chapter 6 the approach for computing the positioning is dependent on time synchronized measurements. It is done in a geometric way only considering the current measurements to compute the current position. This chapter describes some of the techniques which can be used to improve the accuracy of a positioning system. This includes a mobility model, which can be used by a filter to model the movement of the target subject and to limit the fluctuation in the resulting position. This filter is presented to estimate the position considering the previous ones. It has been chosen that this filter should be the Particle Filter in this project.

7.1 Mobility Models

When tracking a MD, it is important to know in which motion pattern it is moving in order to make a mobility model, which together with a filter can improve the accuracy of the positioning. MDs can be moved in many ways depending on whether they are in a train, vehicle, on a bike or just pedestrians. Trains travel in long straight lines and does not change directions often, vehicles follow straight predefined roads which makes it predictable in future positions. Pedestrians on the other hand will move in a more random way that makes them less predictable. They can be either in a stop state or moving with a constant speed for a short time. The motion pattern of the movement follows straight lines.

There exists different models for the patterns of mobility in different situations. Some of them are:

- Constant velocity model
- Less Drunk model
- Random waypoint
- Brownian motion

In this project the Less Drunk Model has been chosen because it is considered as a suitable model for the project scenario, described in Section 1.2. Where the MD has a pedestrian movement pattern. The Less Drunk Model was developed in order to describe such a behavior as detailed in the following section. [Sch]

7.1.1 Less Drunk Model

The scenario of people walking around in e.g. an airport or a shopping mall can be modeled by the Less Drunk model. This model is suitable for pedestrians who are walking around and changing speed and direction frequently.

The characteristics of the model are, that the pedestrian can:

- Have the same speed and direction for a time interval
- Change velocity after a time interval
- Stay in the same point in a time interval

This is illustrated in Figure 7.1.

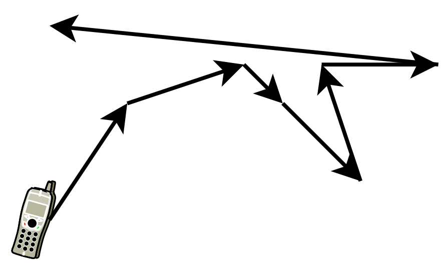


Figure 7.1: *The MD is moving in a pattern as a pedestrian. The MD is moving with the same velocity in a time interval and can choose to stop for a new time interval or change velocity.*

In the model the time is defined discrete as t_n and the speed within a time interval t_n, t_{n+1} is constant. After each time interval it is chosen to either stay in the same place, use the same velocity or choose new velocity. The probability of changing velocity is dependent on a parameter λ , if this value is large this probability is also large, for small λ this probability is small, see Figure 7.2.

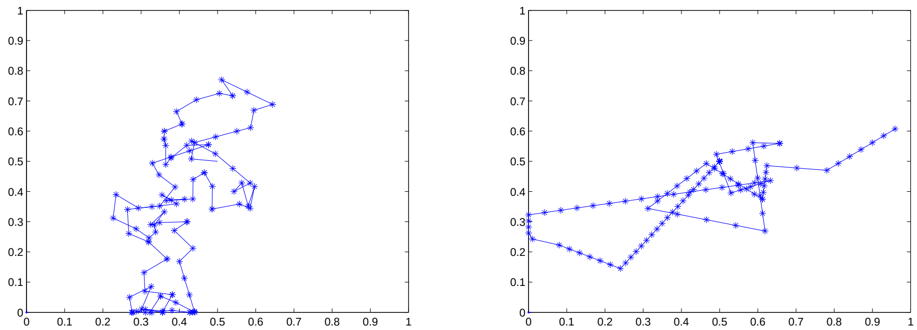


Figure 7.2: *The Less Drunk model for different values of λ. In the left figure large values are chosen and in the right small values.*

A pedestrian can either stay at a specific position for a while or have a speed between limits U and L. This can be seen in Figure 7.3, where the gray area Q represents the different possible velocity vectors (in 2D). The choice, of a velocity equals to zero or not, is made based on a probability Z which is also a parameter of the model. [Sch]

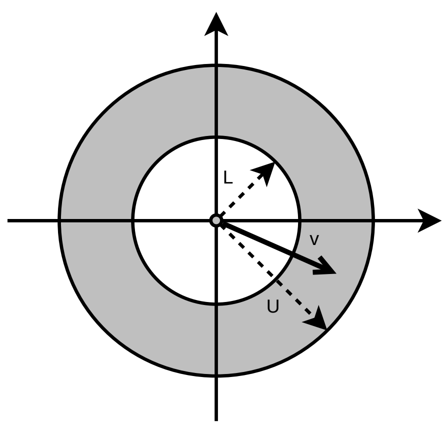


Figure 7.3: *The Figure shows the area Q (in grey). U and L are the upper and the lower limit and v is an example of a velocity that can be selected.*

7.2 Particle Filter

This section describes the Particle Filter in details and is based on [BR04].

Particle Filters is a Nonlinear filtering which is an estimation technique used in different fields such as Tracking Applications, Speech Recognition, Image Processing, Econometrics etc. The goal is to estimate the state of a dynamic system using a list of noisy measurements. In order to perform this estimation, at least two models are required:

- **The dynamic model** which describes the evolution of the system in time.
- **The measurement model** which describes the relation between the noisy measurement and the corresponding state.

A state is modeled as a "state vector". Each component of this vector represents information about the system and it has to be sufficient to describe the system state. In a positioning system the state vector could be the kinematic characteristics (position, velocity and acceleration) of the tracked MD. The principle of Nonlinear filtering is based on a recursive method using the previous state and the received measurement. This is performed in two main steps:

- Prediction: The following state is predicted given the previous state and using the dynamic model.
- Update: The prediction is updated using the latest measurement according to the measurement model.

These steps are a perspective of the Bayesien framework solution to this problem.

7.2.1 Problem Formulation

Let $X_n \in \mathbb{R}^N$ be a random variable representing the state vector, where: N is the dimension of the state vector and $n \in \mathbb{N}$ is a time index assigned to the time instant t_n.

- Dynamic model: The state vector evolution during the time interval $T_n = t_n - t_{n-1}$, is described according to such a model:

  $$X_n = f(X_{n-1}) + \varepsilon_n \tag{7.1}$$

  where f is the mobility model and $\varepsilon_n \sim \mathcal{N}(0, \sigma_n)$ is the noise related to this modeling.
- Measurement model: The observed measurement O_n is obtained from the state X_n using the following model:

  $$O_n = h(X_n) + \omega_n \tag{7.2}$$

  where h is function which can be determined using a theoretical model and empirical calibration for example and $\omega_n \sim \mathcal{N}(0, \delta_n)$ is the measurement noise.

The problem is to find an estimation of X_n using the sequence of all the previous measurements $O_i, i \in \{1, ..., n\}$ for each time index n. The Bayesian solution consists of, first of all, predicting X_n by (7.1), then updating it using the measurement O_n which is obtained by (7.2).

For each $n \geq 1$ and $N \geq n$, the observation O_n is independent of all the other observations and states from 1 to N except the state X_n :

$$P(O_n|X_N, ..., X_n, ...X_1, O_N, ...O_{n+1}, O_{n-1}, ..., O_1) = P(O_n|X_n) \tag{7.3}$$

And for each $n \geq 0$, the state X_{n+1} is independent of all the previous states except the state X_n:

$$P(X_{n+1}|X_n, ..., X_0) = P(X_{n+1}|X_n) \tag{7.4}$$

Thus, the problem can be seen as a Hidden Markov model, see Figure 7.4.

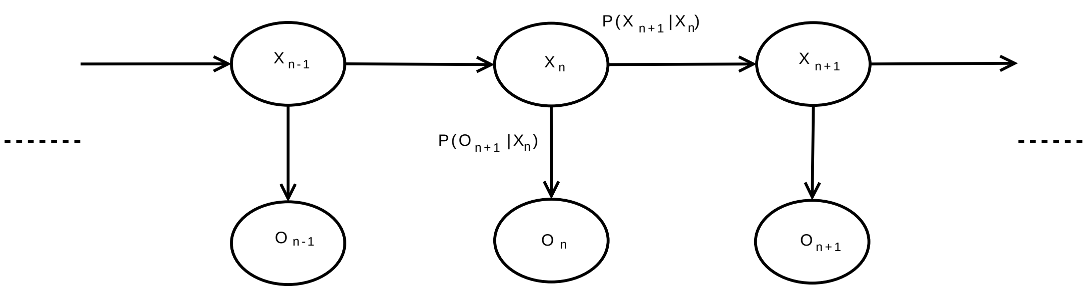


Figure 7.4: *The estimation problem as a Hidden Markov model, where X_n is the hidden state and the random variable O_n is the observation at time t_n.*

7.2.2 The Conceptual Solution

The Particle Filter is a possible approach to solve this problem which uses at each time t_n a set of particles $\left\{X_n^{(m)}/m \in [\![1, M]\!]\right\}$ (M different possible states) related to a set of probabilities $P_{n,m} = P(X_n^{(m)}, O_n, ..., O_1)$.

Starting by an initial density $P(X_0^{(m)})$ of the state vector, the initial set of particles is generated. Usually, at this initial time instant, there is no available information about the state. The uniform distribution is then the most appropriate for the initial distribution.

Assume that such a set $\left\{X_n^{(m)}/m \in [\![1, M]\!]\right\}$ with the corresponding set of probabilities $P_{n,m}$ is available for the time index n. Using the mobility model, see (7.2), the new set $\left\{X_{n+1}^{(m)}/m \in [\![1, M]\!]\right\}$ is obtained.

Considering all possible states X_n the probabilities $P_{n+1,m} = P(X_{n+1}^{(m)}, O_{n+1}, ..., O_1)$ for $m \in$

$[\![1, M]\!]$ can be calculated as:

$$\begin{aligned} P(X_{n+1}^{(m)}, O_{n+1}, ..., O_1) &= \sum_{i=1}^{M} P(X_{n+1}^{(m)}, X_n^{(i)}, O_{n+1}, ...O_1) \\ &= \sum_{i=1}^{M} P(O_{n+1}|X_{n+1}^{(m)}, X_n^{(i)}, O_n, ...O_1) \cdot P(X_{n+1}^{(m)}, X_n^{(i)}, O_n, ...O_1) \end{aligned} \tag{7.5}$$

From the equality 7.3:

$$P(O_{n+1}|X_{n+1}^{(m)}, X_n^{(i)}, O_n, ..., O_1) = P(O_{n+1}|X_{n+1}^{(m)}) \tag{7.6}$$

And:

$$P(X_{n+1}^{(m)}, X_n^{(i)}, O_n, ...O_1) = P(X_{n+1}^{(m)}|X_n^{(i)}, O_n, ..., O_1) \cdot P(X_n^{(i)}, O_n, ..., O_1) \tag{7.7}$$

From Proof 1 in Appendix C :

$$P(X_{n+1}^{(m)}|X_n^{(i)}, O_n, ..., O_1) = P(X_{n+1}^{(m)}|X_n^{(i)}) \tag{7.8}$$

Which leads to :

$$P(X_{n+1}^{(m)}, X_n^{(i)}, O_n, ...O_1) = P(X_{n+1}^{(m)}|X_n^{(i)}) \cdot P(X_n^{(i)}, O_n, ..., O_1) \tag{7.9}$$

Thus:

$$P(X_{n+1}^{(m)}, O_{n+1}, ..., O_1) = \sum_{i=1}^{M} P(O_{n+1}|X_{n+1}^{(m)}) \cdot P(X_{n+1}^{(m)}|X_n^{(i)}) \cdot P(X_n^{(i)}, O_n, ..., O_1) \tag{7.10}$$

$P(X_{n+1}^{(m)}|X_n^{(i)})$ can be obtained from the mobility model and $P(O_{n+1}|X_{n+1}^{(m)})$ from the measurement model. Then, the probabilities $P_{n,m} = P(X_n^{(m)}, O_n, ..., O_1)$ are updated using (7.10). Thereby, for each time instant t_n, a set of particles $\left\{X_n^{(m)}/m \in [\![1, M]\!]\right\}$ with the corresponding set of probabilities $P_{n,m}$ is available recursively. These particles give an idea about the distribution of the state X_n. This random variable can be estimated either by the minimum mean-square error estimate :

$$\hat{X_n} = E\{X_n|O_n, ..., O_1\} = \sum_{m=1}^{M} X_n^{(m)} \cdot P_{n,m} \tag{7.11}$$

or the maximum a posteriori estimate:

$$\hat{X_n} = \arg \max_{m \in \{1..M\}} P_{n,m} \tag{7.12}$$

7.3 Design of the Enhanced Positioning Method

In this section, the design of the functions `Measurement Filter` and `Compute Position` adapted to the enhanced approach will be described. The other functionalities of the system remain as designed in Chapter 6.

7.3.1 Measurements Filter

As it was discussed in Section 6.2, the measurements received from the BSs are not synchronized. In the simple approach, where the tracking is performed using trilateration without considering the dynamics of the system, three measurements are needed for the same time stamp.

But in this enhanced approach, more measurements can be taken into consideration. This is done by introducing a value for an *empty* measurement and adapt the contingency table to it as it is detailed below.

7.3.2 Compute Position

In this subsection, a design of the Particle Filter, described in Section 7.2, for position computation will be detailed. The Less Drunk Model, described in Section 7.1.1, is chosen as a mobility model.

For a tracking problem, the hidden state vector is composed of the space velocity coordinates of the mobile device. As long as the pedestrian behavior of the MD is considered having the same velocity during each time interval, the acceleration is zero. Therefore, the state vector for a time index n, in a known coordinate system, is:

$$X_n = [x_n, y_n, \dot{x}_n, \dot{y}_n] \tag{7.13}$$

Regarding the observed state vector, it is composed of the measurements of the signal strength from the different BSs. as described in Section 7.3.1, the controller is designed so as to provide a list of measurements from all the BSs in the same time instant, with an *empty* measurement when the corresponding BS does not respond. Then, considering the scenario of three base stations, the observation vector for a time index n is:

$$O_n = [O_{n,1}, O_{n,2}, O_{n,3}] \tag{7.14}$$

Where $O_{n,i},\ i = 1...3$ is the measured signal strength from the BS i at the time t_n, the value $O_{n,i} = 0$ is attributed for the *empty* measurement.

The initial set of particles $\left\{X_0^{(m)}/m \in [\![1, M]\!]\right\}$ is a randomly generated with a uniform distribution. Thus the corresponding probabilities are : $P_{0,m} = \dfrac{1}{M},\ \forall m \in [\![1, M]\!]$.

As seen in the conceptual solution, the state at the time t_n is estimated using a set of probabilities $P_{n,m}$ related to the particles $X_n^{(m)}$ $(m = 1...M)$ and the observation O_n. Starting by the initial state, these two sets are obtained recursively.

Assume that at the time t_n, such a set $\left\{X_n^{(m)}/m \in [\![1, M]\!]\right\}$ with the corresponding set of probabilities $P_{n,m}$ is available. The next sets are then obtained as explained below:

Presuming that from t_n to t_{n+1} the MD has a constant velocity, each particle will also have a constant velocity. Thus, the positions (for $m \in [\![1, M]\!]$ at the time t_{n+1} can be calculated using:

$$x_{n+1}^{(m)} = T_{n+1} \cdot \dot{x}_n^{(m)} + x_n^{(m)} \tag{7.15}$$

$$y_{n+1}^{(m)} = T_{n+1} \cdot \dot{y}_n^{(m)} + y_n^{(m)} \tag{7.16}$$

Where $T_{n+1} = t_{n+1} - t_n$

But, according to the Less Drunk Model, from the time interval $T_n = t_n - t_{n-1}$ to $T_{n+1} = t_{n+1} - t_n$ the MD can change its velocity with a probability λ. So from these new positions, two sets of particles are obtained. The first one corresponds to the states of the new positions with the old velocities, and the second one to the new positions with new velocities, see (7.17) and (7.18) below. The new velocities $(\dot{x}_{n+1}, \dot{y}_{n+1})$ are randomly selected from the area Q detailed in section 7.1.1, see Figure 7.3, using a parameter Z as probability of having a velocity equals to zero.

Thus, at this step two sets are available :

$$\left\{X'^{(m)}_{n+1} = [x_{n+1}, y_{n+1}, \dot{x}_n, \dot{y}_n] \, /m \in [\![1, M]\!]\right\} \tag{7.17}$$

$$\left\{X''^{(m)}_{n+1} = [x_{n+1}, y_{n+1}, \dot{x}_{n+1}, \dot{y}_{n+1}] \, /m \in [\![1, M]\!]\right\} \tag{7.18}$$

Remark As long as the new velocities are randomly selected according to the Less Drunk Model, the set with new velocities above (7.18) can have a size larger than M by choosing other possible velocities. But as a simplification, it has been chosen in this project to consider only M particles with new velocities at each step.

The probabilities $P_{n,m}$ are also updated using the (7.10), which gives for $m \in [\![1, M]\!]$:

$$P'_{n+1,m} = \sum_{i=1}^{M} P(O_{n+1}|X'^{(m)}_{n+1}) \cdot P(X'^{(m)}_{n+1}|X_n^{(i)}) \cdot P_{n,i} \tag{7.19}$$

$$P''_{n+1,m} = \sum_{i=1}^{M} P(O_{n+1}|X''^{(m)}_{n+1}) \cdot P(X''^{(m)}_{n+1}|X_n^{(i)}) \cdot P_{n,i} \tag{7.20}$$

And knowing that for each $i \in [\![1, M]\!]$, the particle $X'^{(i)}_{n+1}$ is obtained from the new position of $X_n^{(i)}$ at time t_{n+1} keeping the same velocity and $X''^{(i)}_{n+1}$ corresponds to the same new position but the velocity changing at time t_{n+1}. The probability of changing velocity is λ, then:

$$P({X'}_{n+1}^{(m)}|X_n^{(i)}) = \begin{cases} 1-\lambda \text{ if } m=i \\ 0 \text{ else} \end{cases} \tag{7.21}$$

$$P({X''}_{n+1}^{(m)}|X_n^{(i)}) = \begin{cases} \lambda \text{ if } m=i \\ 0 \text{ else} \end{cases} \tag{7.22}$$

Using 7.21 and 7.22 the (7.19) and (7.20) become:

$$P'_{n+1,m} = P(O_{n+1}|{X'}_{n+1}^{(m)}) \cdot P({X'}_{n+1}^{(m)}|X_n^{(m)}) \cdot P_{n,m} \tag{7.23}$$

$$P''_{n+1,m} = P(O_{n+1}|{X''}_{n+1}^{(m)}) \cdot P({X''}_{n+1}^{(m)}|X_n^{(m)}) \cdot P_{n,m} \tag{7.24}$$

The observation O_{n+1}, which is composed of the signal strength measurements, is independent of the velocity. And for each m, the states ${X'}_{n+1}^{(m)}$ and ${X''}_{n+1}^{(m)}$ correspond to the same position, thus:

$$P(O_{n+1}|{X'}_{n+1}^{(m)}) = P(O_{n+1}|{X''}_{n+1}^{(m)}) \tag{7.25}$$

The measurements $O_{n,i},\ i = 1...3$ are independent, thus:

$$P(O_{n+1}|{X'}_{n+1}^{(m)}) = P(O_{n+1,1}|{X''}_{n+1}^{(m)}) \cdot P(O_{n+1,2}|{X''}_{n+1}^{(m)}) \cdot P(O_{n+1,1}|{X''}_{n+1}^{(m)}) \tag{7.26}$$

From the (7.29), the observation $O_{n+1,i}$ depends only on the distance $d_{n+1,i} = \sqrt{(x_{-}a_i)^2 + (y_n - b_i)^2}$ for different states X_{n+1}, where (a_i, b_i) are the coordinates of the BS i. Thus, the dependency between $O_{n+1,i}$ and ${X'}_{n+1}^{(m)}$ is simplified into the dependency between $O_{n+1,i}$ and $d_{n+1,i}$.

Then, the remaining unknowns are values such as $P(O_i|d_j)$, where the distance and observation spaces are discretized into values : $\{O_i/i \in [\![0, n-1]\!]\}$ and $\{d_j/j \in [\![0, m-1]\!]\}$, for each BS. And this value can be obtained using two methods:

Statistical Contingency Table :

The contingency table is a matrix C (size $n * m$) defined as

$$C_{i,j} = P(O_i|d_j) \tag{7.27}$$

The values $P(O_i|d_j)$ are obtained statistically by getting several measurements O_i for the same distance d_j. Then:

$$P(O_i|d_j) = \frac{N_{i,j}}{N_j} \tag{7.28}$$

Where $N_{i,j}$ is the number of occurrences of the measurement O_i for the distance d_j, and N_j is the number of measurements for the distance d_j.

Path loss model :

The path lost model, see Section 2.2.4 states that the RSS is related to the distance between the MD and the BS i according to the following equation:

$$O_i = P_0 - 10\beta \log\left(\frac{d_j}{d_0}\right) + \omega \tag{7.29}$$

Where :

- P_0 is the transmitted power from the antenna
- d_0 is a reference distance for the antenna far field (at this distance the received power equals the transmitted)
- β is the path loss exponent
- $\omega \sim \mathcal{N}(0, \delta)$ is the measurement noise.

The parameters P_0, β and δ can be found experimentally.

Thereby, given the distance d_j, O_i has a normal distribution with the parameters:

- Mean :$M = P_0 - 10\beta \log\left(\frac{d_j}{d_0}\right)$
- Standard deviation: δ.

Which gives the value of: $P(O_i|d_j)$.

The *empty* measurement in the contingency table :

As stated in Section 7.3.1, the Particle Filter does not require measurements from all the BSs at the same time. To deal with this, the *empty* measurement was introduced. Therefore, the contingency table has to be adapted to this new measurement as detailed below.

The number of measurements $\{N(t)/t \geq 0\}$ received by the controller from a specific BS up to time t is a homogeneous Poisson process with associated rate parameter λ.

Let $C_{i,j} = P(O_i|d_j)$ be the contingency table for the normal measurements of RSSI, where d_j is the distance from the BS discretized into values $\{d_j/j \in [\![0, m-1]\!]\}$ and O_i is the RSSI measurement discretized into values $\{O_i/i \in [\![0, n-1]\!]\}$. Considering both cases for a time duration T, when a measurement is received and when not, the new contingency table is:

$$\begin{aligned} C' &= \begin{pmatrix} \mu T \cdot C \\ 0 \cdots 0 \end{pmatrix} + \begin{pmatrix} 0 \\ (1-\mu T) \cdots (1-\mu T) \end{pmatrix} \\ &= \begin{pmatrix} \mu T \cdot C \\ (1-\mu T) \cdots (1-\mu T) \end{pmatrix} \end{aligned} \tag{7.30}$$

Where: $\mu = P(N(t+T) - N(t) \geq 1) = 1 - e^{-\lambda T}$, see Proof 2 in Appendix C.

Then, when the probabilities are updated using the equations 7.19 and 7.20, the new set of states $\left\{ X_{n+1}^{(m)} / m \in [\![1, M]\!] \right\}$ is selected from elements of both sets : $\left\{ X'^{(m)}_{n+1} / m \in [\![1, M]\!] \right\}$ and $\left\{ X'^{(m)}_{n+1} / m \in [\![1, M]\!] \right\}$ by choosing the ones with the highest probabilities. And given this new set, the new state X_{n+} can be estimated using the mean-square error estimate, see (7.11), or the maximum a posteriori estimate, see (7.12).

Chapter 8

Implementation and Validation

This chapter describes the implementation of the simple positioning system prototype designed in Chapter 6 and the enhanced prototype which is designed in Section 7.3. The prototype implemented is however a more simple version then the prototype designed. Therefore parts of the implementation that deviates from the design will also be clarified in this chapter.

8.1 System Overview

For the implementation of this prototype, CSR has provided three Casira endpoint development kits which should serve as BSs in the positioning system. They can be connected to a PC through various interfaces e.g. USB and RS-232 serial port. In order to implement a prototype according to a network based technique, three Casira devices can be used as BSs. Each one of them will be directly connected by USB to the controller. The controller is a PC running an application which has a list of all the BSs and their positions. The setup can be seen in Figure 8.1. The Casira devices will get continuous RSSI measurements and the application will compute the position using these measurements.

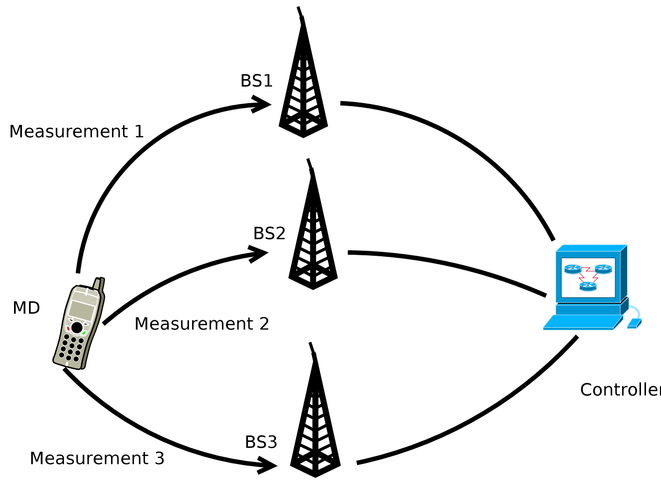


Figure 8.1: *System setup of the implementation of the network based technique.*

In the following sections the deviations from the design will be described. A general comment to the implementation is that it is done in an analytical way by recording raw measurements and doing the localization processing afterwards, rather than performing real time tracking.

Both the simple method using direct computation by trilateration and the enhanced one using the Particle Filter and the Less Drunk model, were implemented, tested and then used for the measurement data, see Chapter 9. The implementation was done so it respects the test scenario, see Section 9.1. The data processing for measurements filtering, trilateration, etc. was all implemented in MATLAB which is suited for handling large amounts of vector based data.

8.2 RSSI Measurement using Casira Devices

The implementation of the Casira device regarding the inquiry of a MD and the RSSI measurement is achieved by executing the Serial Port Application software provided by CSR. The Serial Port Application is a simple example application that utilizes the property of the inquiring mechanism, which enables retrieving the RSSI value from the detected MDs.

In this project an initiation procedure has been added to the existing Serial Port Application and the implementation of the Casira application and the Bluetooth protocol will be considered as a black box. This black box will return the MAC-addresses and the RSSI-values of the MDs which is described in Section 6.1

This is carried out in two steps.

1. The black box, see Figure 8.2, where the MAC-addresses and RSSI-values are seen as an

output of the black box. This output is the response to an inquiry.

2. How the RSSI value is retrieved from this output of the black box, see Figure 8.3.

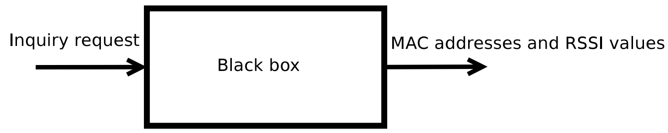


Figure 8.2: *Black box approach, where the MAC-addresses and the RSSI-values is returned as a response to a inquiry request.*

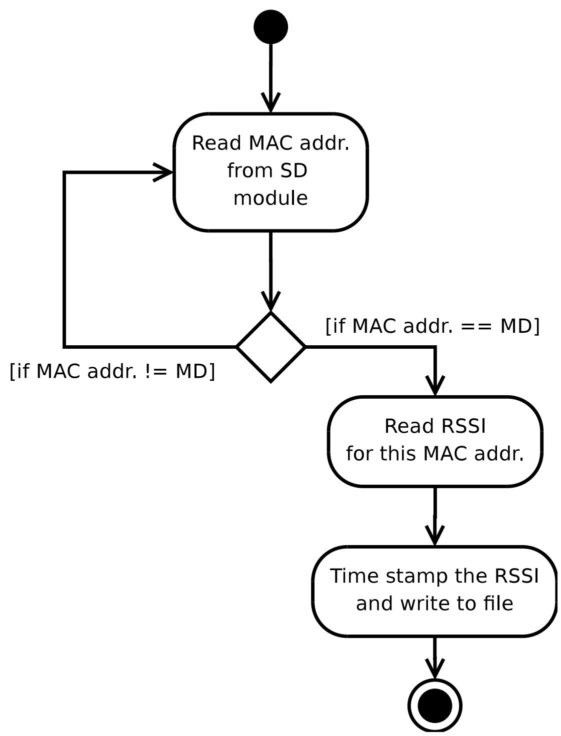


Figure 8.3: *Activity diagram showing the implemented flow of the Casira.*

Each time the black box returns a MAC address of a detected device, the address of this device is compared with the MAC address of the MD which is tracked. If these two MAC addresses are the same then the RSSI value for this MD is stored in a file with a unique time stamp. The procedure which can be seen in Figure 8.2 and 8.3 is repeated until the application exits.

8.3 Simple Positioning

The implementation of the simple approach, which was designed in Chapter 6, will be detailed in this section by describing the important functions used for the data processing.

8.3.1 Measurements Filter for Trilateration

The measurements filter for trilateration is implemented as described in the design, Section 6.2.1. It loads the files containing measurements from all BSs and discards the time windows not containing all BS measurements. The time window size can be adjusted in order to maximize the output of the current measurements. As expected, from the results of João Figueiras research presented in Section 3.1, a window size of approximately 4-6 seconds will contain inquiry results from all BSs in most cases. Larger window size will also contain all BS results, but the time between results will be longer and the rate of new positions correspondingly lower. Thus a window size of 5 is considered to be the best trade off.

8.3.2 Convert to Distance

The implementation of the `Convert to distance` function was done by a calibration experiment. The purpose of this calibration is to compare the different distances with the RSSI-values to determine a relation between RSSI and distance. In this experiment a Nokia N70 was used as MD and a CSR Casira was used as BS. The experiment was performed with the BS and the MD placed on the floor. The MD was placed at every 5 cm from the BS in a straight line, and 65 measurements were obtained from each distance. The setup can be seen in Figure 8.4.

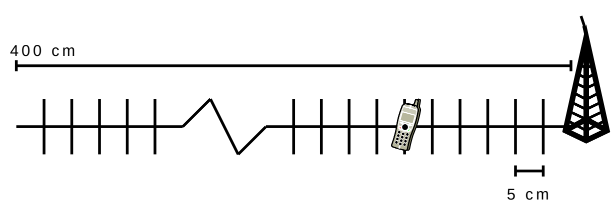


Figure 8.4: *The test setup for the calibration test.*

At each distance, the mean of the 65 measurements is computed as an attempt to average out noise, but multipath phenomenon (from the floor and distant walls) may still be present. The graph in Figure 8.5 shows the relation between distance and RSSI. The black line is the mean of the obtained measurements and it is clear that multipath phenomenon are not removed by computing the mean. The red line is a fitted function using the path loss formula of (6.2) with

$P_0 = -33$ (RSSI), $\beta = 2.5$ and $d_0 = 5cm$ resulting in:

$$RSSIdB = -33 - 10 \cdot 2.5 \cdot \log\left(\frac{d}{5cm}\right) \tag{8.1}$$

This relation is then used to find the distance from the measured RSSI.

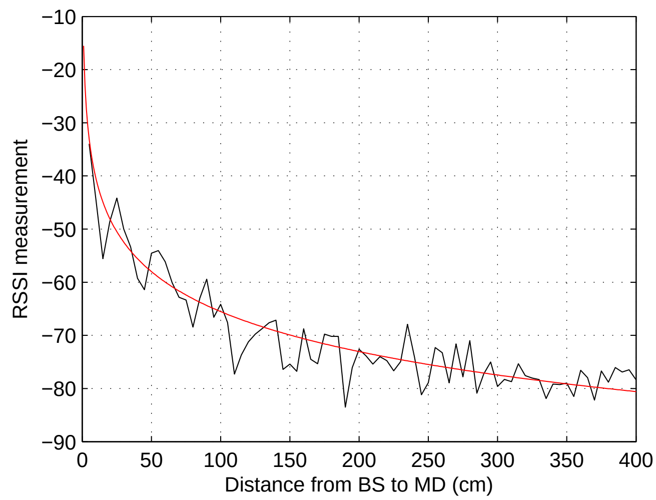


Figure 8.5: *The relation between distance and RSSI. The black line is the measurements and the red line is a fitted logarithmic function.*

From the plot in Figure 8.5 it can be seen that after aproximately 3 m, it is difficult to determine a good estimate of the correct distance based on the RSSI values, because the slope of the curve is too flat and the error is too large.

8.3.3 Trilateration

Two different algorithms for performing trilateration were described in the Section 2.3. The first algorithm `algo1` was implemented exactly as presented in this section, but the second algorithm was implemented in two different approaches.

The first approach for the second algorithm `algo2.1`, see description of Algorithm 2 in Section 6.2.2, uses a geometrical solution, where the intersection area between the four circles: $\hat{C}_1$, $\hat{C}'_1$, $\hat{C}_2$ and $\hat{C}'_2$ is considered roughly as two quadrilaterals, they will be crossing each others when the tracked point is near to the x-axis. The circles $\hat{C}_3$ and $\hat{C}'_3$ are used only to determine which quadrilateral is going to be chosen. Finally the estimate position will be the centroid of this area.

In the second method `algo2.2`, the rings are discretized into circles with a specific step, which is a parameter depending on the measurement noise. The intersection of the 3 rings is found discretized as well, according to the step chosen.

These algorithms are summarized in Table 8.1. Where σ is the standard deviation of the noise in the measurements, α is the probability introduced in the description of Algorithm 2, and *step* is the resolution of the discretization used in `algo2.2`.

Algorithm	algo1	algo2.1	algo2.2
Inputs	Distance measurements from the 3 BSs	Distance measurements from the 3 BSs	Distance measurements from the 3 BSs
Parameters	The BSs coordinates	The BSs coordinates, σ and α	The BSs coordinates, σ, α and *step*
Outputs	The estimated position of the MD	The estimated position of the MD	The estimated position of the MD
Comments	Implemented exactly as detailed in the design	Geometrical approximation of the solution	Solution using a discrete space

Table 8.1: *The different algorithms used in implementing the trilateration*

The implementation of these algorithms was done in Java. Then a test program was implemented in order to compare them. The test program is given a correct position, then a Gaussian noise is added to the distance measurement. Finally, the algorithms : `algo1`, `algo1.2` and `algo2.2` estimate the position using only the noisy measurements and the parameters summarized in Table 8.1. As simplification, the noise was considered independent of the distance.

In Table 8.2, the estimation error for several correct positions and different values of σ, the standard deviation of the noise, are presented. This error is obtained by calculating the distance difference between the estimation and the correct position.

This was done with adjusted values of α (between 0.7 and 0.9) and $step = \frac{\sigma}{100}$. The distance unit used is cm, the BSs coordinates are (0,0), (0,200) and (200,0) in respect to the test scenario, see Section 9.1

.

correct coordinates	σ	**algo1.1 error**	**algo2.1 error**	**algo2.2 error**
(30,60)	1	0.6131	0.6068)	0.6425
(30,60)	10	15.0450	15.4640	6.6597
(180,20)	1	0.6275	0.6013	0.2064
(180,20)	10	19.9960	30.5740	1.3642
(100,100)	1	1.6505	1.6386	0.5485
(100,100)	10	19.6710	20.1630	10.3850
(170,159)	1	1.8753	1.8763	0.2260
(170,159)	10	8.9816	8.3863	4.0275

Table 8.2: *Test-comparison between the 3 different implementations of trilateration*

These results show that `algo2.2` gives better approximation to the position. Therefore, it has been decided to implement it in MATLAB and use it later for data processing.

8.3.4 Validation of Trilateration

The validation of the trilateration is conducted by feeding the algorithm, described in Section 6.2.2, with simulated data which define a trajectory of a MD with respect to the test scenario, see Figure 9.1. This trajectory is achieved by executing a MATLAB script, which outputs (x,y) coordinate. Based on this simulated coordinate the distance from the BS to the MD is calculated.

The validation is based on the following two cases:

- Case 1: The simulated coordinate represent the correct trajectory of a MD.
- Case 2: The simulated coordinate is contaminated by Gaussian distributed noise with standard deviation σ

The trilateration algorithm is then executed, and as the output, an estimation of the trajectory of this simulated path of the MD is shown in Figure 8.6 and 8.7.

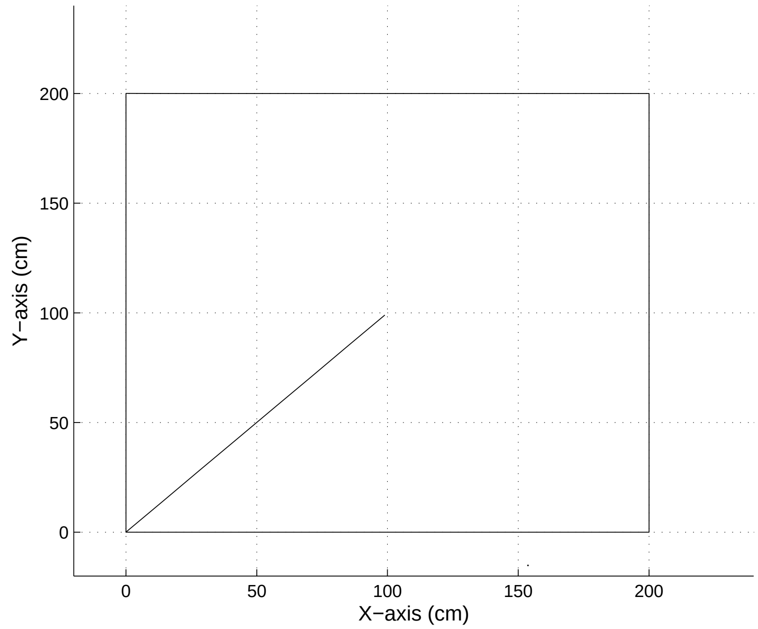


Figure 8.6: *This figure represents the results calculated by MATLAB using the trilateration approach. In this case the trilateration algorithm uses data with 0 noise, this means that the BSs are expected to return the correct distance to the moving MD.*

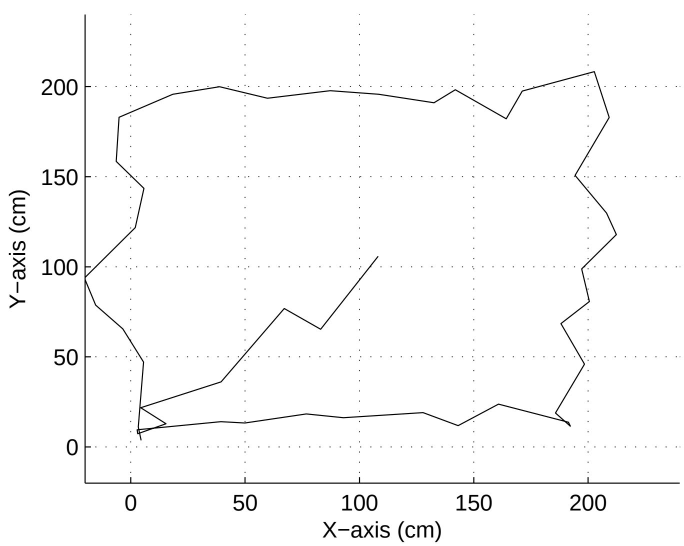


Figure 8.7: *Trilateration approach, where noise is added to the simulated coordinate sets. The added noise is Gaussian distributed with* $\sigma = 10$.

Figure 8.6 and 8.7 shows that the trilateration method is capable of tracking the trajectory of a MD from the simulated data. In the first test case the distance returned from the BSs are considered being exactly. The second test case was when the noise is added to the simulated distance data, as it can be seen from Figure 8.7 in second test case the trilateration method was not able to give an exact mapping of the simulated trajectory of the MD, but is still able to represent the main trend of the real path.

8.4 Enhanced Positioning

The implementation for the enhanced method which is designed in Section 7.3 will be detailed in this section by describing the important functions.

8.4.1 Measurements Filter for Particle Filter

The measurements filter used in the Particle Filter is almost the same as the one used for trilateration. The measurements are grouped in windows and the mean is computed. The difference from the trilateration filter, is that a missing measurement from a BS will not result in the others being discarded. In comparison to the trilateration, the particle filter does not need measurements from all BS in every time window to compute the estimated position. Thus an empty window from one BS will result in an empty measurement, which is tolerated by the Particle Filter.

8.4.2 Contingency Table

As stated in Section 7.3, the Particle Filter requires a contingency table. Two different methods of obtaining the contingency table were described in Section 7.3.2.

The distance space was discretized into values: $\{-20, -15, -10...., 395, 400\}$ and the RSSI measurements into values $\{-33, -34, ... -89, -90\}$

The empirical method is based on statistical estimation of the conditional probability $C_{i,j} = P(O_i|d_j)$ using the measurements obtained from the calibration experiment, see Section 8.3.2. This method was implemented, but its results were not appropriate when the test was performed later. That is because the 65 experiment measurements for each distance are not sufficient for doing a statistical estimation.

Thus, the method which will be used for the implementation is the one based on the path loss model, see Section 7.3.2. Where the probabilities $C_{i,j} = P(O_i|d_j)$ are obtained from a gaussian distribution with as mean $M = P_0 - 10\beta \log\left(\frac{d_j}{d_0}\right)$ and a standard deviation δ. The Figure 8.8 shows the contingency table implemented in MATLAB with $\delta = 2.7$. Using the measurements from the calibration experiment detailed in Section 8.3.2, the parameters were obtained:

- $P_0 = -60\,dBm$
- $\beta = 1.3$
- $d_0 = 1\,cm$

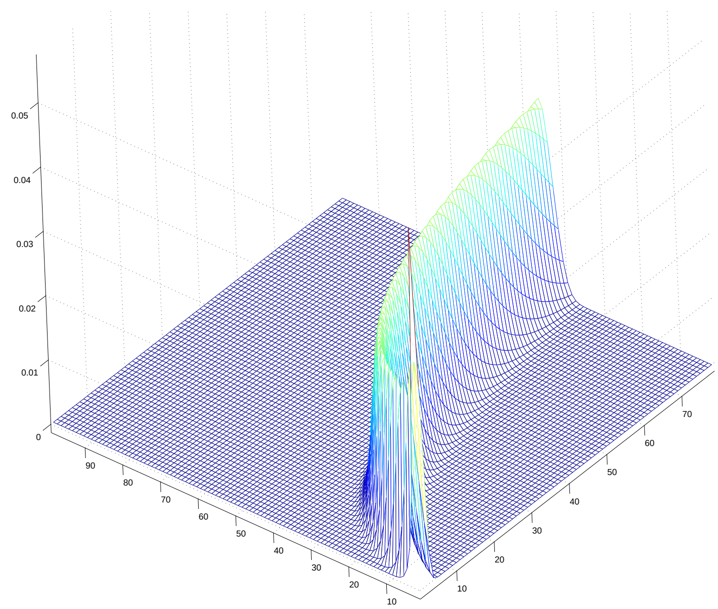


Figure 8.8: *The contingency table implemented using the path loss model. The x and y axis represent the matrix indexes and the z axis represents the value of the corresponding probability.*

Then, the procedure detailed in Section 7.3.2 to deal with the *empty* measurement was applied to this contingency table, using $\mu = \frac{1}{5}T$, because the BSs were giving in average 1 measurement in 5 seconds when the test was performed.

8.4.3 Particle Filter using the Less Drunk Model

The Particle Filter which is designed in Section 7.3.2 has been implemented as a function in MATLAB. The inputs are two matrices: O and C. O is the matrix containing the filtered measurements with corresponding time stamps. And C is the contingency table obtained from Section 8.4.2.

For each i:

- O(i,1) is the time stamp.
- O(i,2) is the measurement from the BS 1.
- O(i,3) is the measurement from the BS 2.

- O(i,4) is the measurement from the BS 3.

And for each $(i, j) : C_{i,j} = P(O_i|d_j)$.

The implementation of this procedure was done exactly as described in Section 7.3.2. The only difference is that in the design, the probabilities utilized for the update are the joint probabilities: $P_{n,m} = P(X_n^{(m)}, O_n, ..., O_1)$. But, as long as MATLAB has a finite number of possible decimals and these probabilities approche zero for large value of N, this is not an appropriate way to calculate the probabilities. Thus, the conditional probabilities, see (8.2), have been used for the implementation.

$$P(X_n^{(m)}|O_n, ..., O_1) = \frac{P(X_n^{(m)}, O_n, ..., O_1)}{\sum_{i=1}^{M} P(X_n^{(i)}, O_n, ...O_1)} \tag{8.2}$$

The used parameters were adapted to the test scenarios, see Section 9.1. E.g, the parameters used for the one of the tests are :

- $L = 1.18$ cm/s
- $U = 4$ cm/s
- $Z = 0.08$
- $\lambda = 0.6$
- $M = 1000$

The estimation is done using the maximum a posteriori estimate.

The output is then a matrix X with the different position estimations corresponding to the time stamps in the input O.

8.4.4 Validation of the Particle Filter

In this section the implemented Particle Filter is validated. This validation is carried out in the same way as the validation of the trilateration described in Section 8.3.4.

This validation is based on the following three cases:

- Case 1: The simulated coordinate represent the correct trajectory of a MD.
- Case 2: The simulated coordinate is contaminated by Gaussian distributed noise with standard deviation $\sigma = 10$
- Case 3: The simulated coordinate is contaminated by Gaussian distributed noise with standard deviation $\sigma = 15$

Thus, the observation matrix O is composed of distances and time stamps with a step of 1 second. Than the corresponding contingency table C is built as well. In the ideal case, the contingency table will be the identity matrix. But, as long as the Particle Filter is based on a discretization of the state space using a set of particles, the identity matrix is not suitable. No particle will fit the exact distance. Therefore, the used contingency table was a matrix where the values are gaussian distributed around the diagonal, see Figure 8.9. The standard deviation was adapted depending on the noise added to the distance measurement.

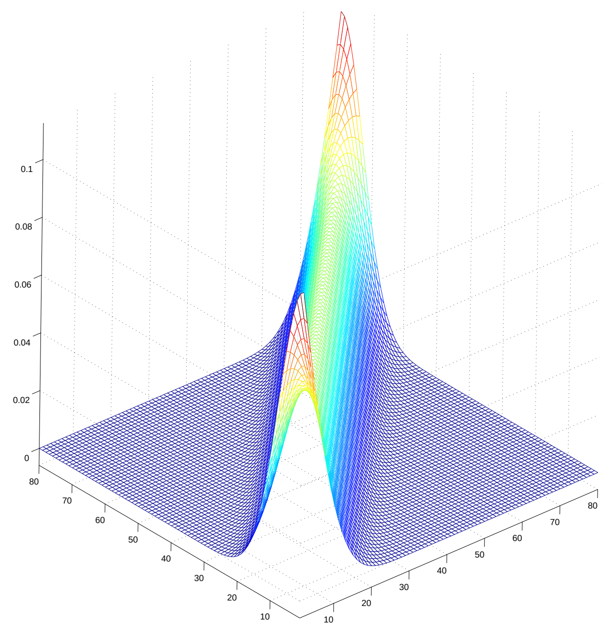


Figure 8.9: *The contingency table used for the validation. The x and y axis represent the matrix indexes and the z axis represents the value of the corresponding probability.*

The simulated data from these three cases are processed by the Particle Filter and an estimation of the trajectory of the MD, is shown on Figure 8.10, 8.11 and 8.12.

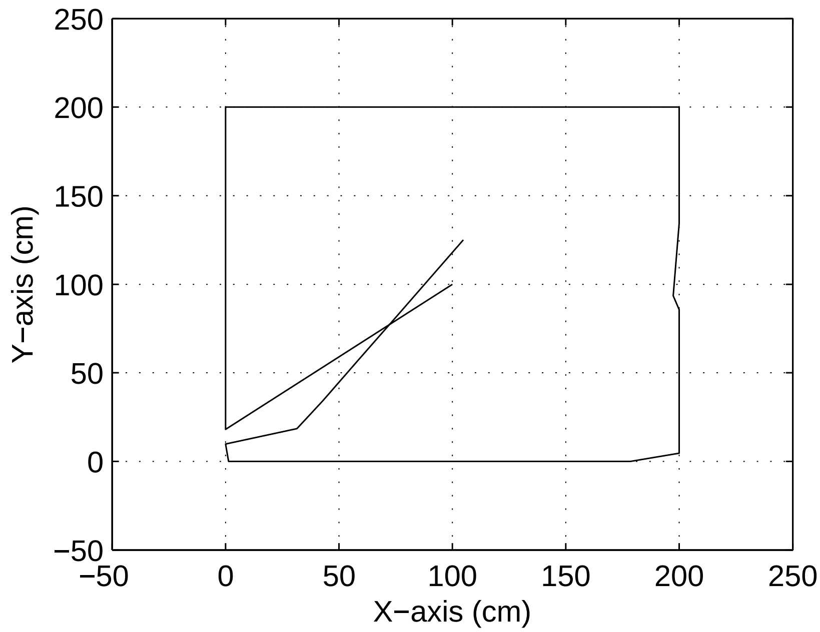


Figure 8.10: *This figure represents the results calculated by MATLAB using the Particle Filter. In this case the this algorithm uses data with 0 noise, this means that the BSs are expected to return the correct distance to the moving MD.*

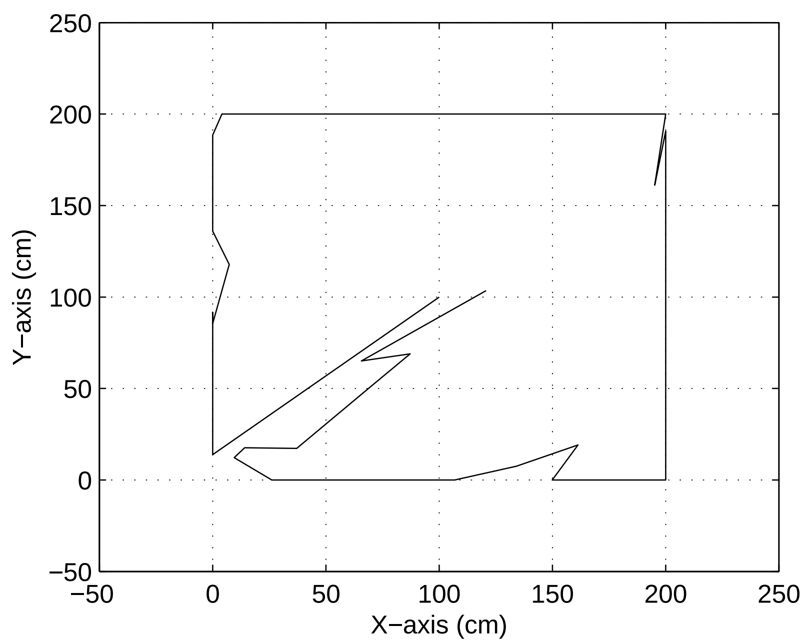


Figure 8.11: *resuts of the Particle Filter, where noise is added to the simulated coordinate sets. The added nois is Gaussian distributed with* $\sigma = 10$.

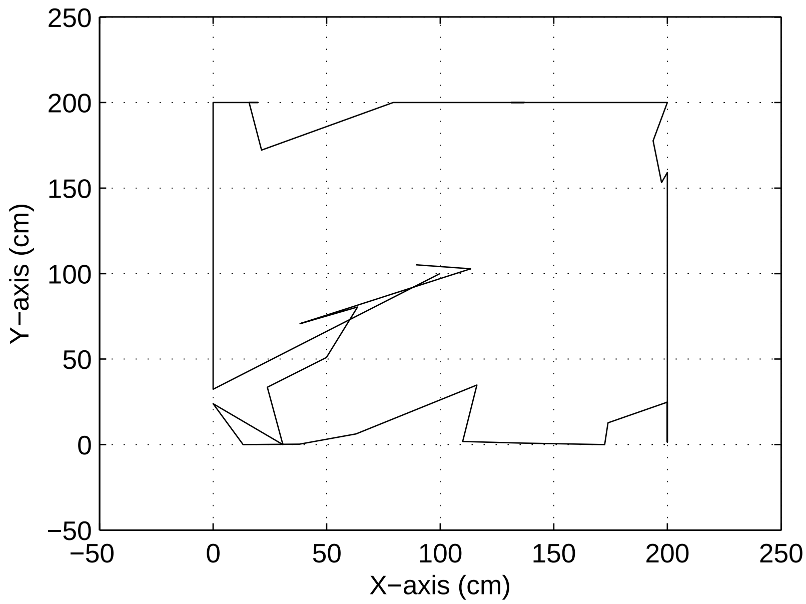


Figure 8.12: *resuts of the Particle Filter, where noise is added to the simulated coordinate sets. The added nois is Gaussian distributed with $\sigma = 15$.*

Figure 8.10, 8.11 and 8.12 shows that the Particle Filter is capable of tracking the trajectory of a MD from the simulated data. From these figures it can be seen that some of the estimation made by the Particle Filter differs from the original position and this tendency increases as the deviation σ increase. In all these three cases the Particle Filter manages to estimate the correct path properly from the simulated data.

Chapter 9

Test

The scenario chosen in Section 1.2 indicates that the MD is moving in straight lines mostly along walls. From this a test case has been set up and it will be described in the following section. After this the test specification is detailing how the tests should be performed. Finally the obtained measurements from the tests will be fed to both the simple and enhanced prototype, and the results are described in Chapter 8.

9.1 Test Scenarios

Several different tests cases were set up according to the chosen scenario. The following is a description of the goal and the execution of these tests. The goal of these tests was to obtain measurements which can be fed to the two different positioning prototypes. The results from the different tests can be seen in Section 9.3

A Nokia N70 was used as MD and three CSR Casira devices were used as BSs. The test setup with the path of the MD and the positions of the BSs can be seen in Figure 9.1. It has been chosen to perform the test in a 240x240 cm square room based on the result of the calibration test in Section 8.3.2.

The real path had to be recorded, i.e. when the mobile phone was at specific places. This was done by recording timestamps every 20 cm. By doing so it was possible to compare the real path with the obtained measurements. All three BSs were obtaining measurements at the same time during this test. The tests were performed both with and without random appearing obstacles in the room. The obstacles used in this test were two group members moving in the room.

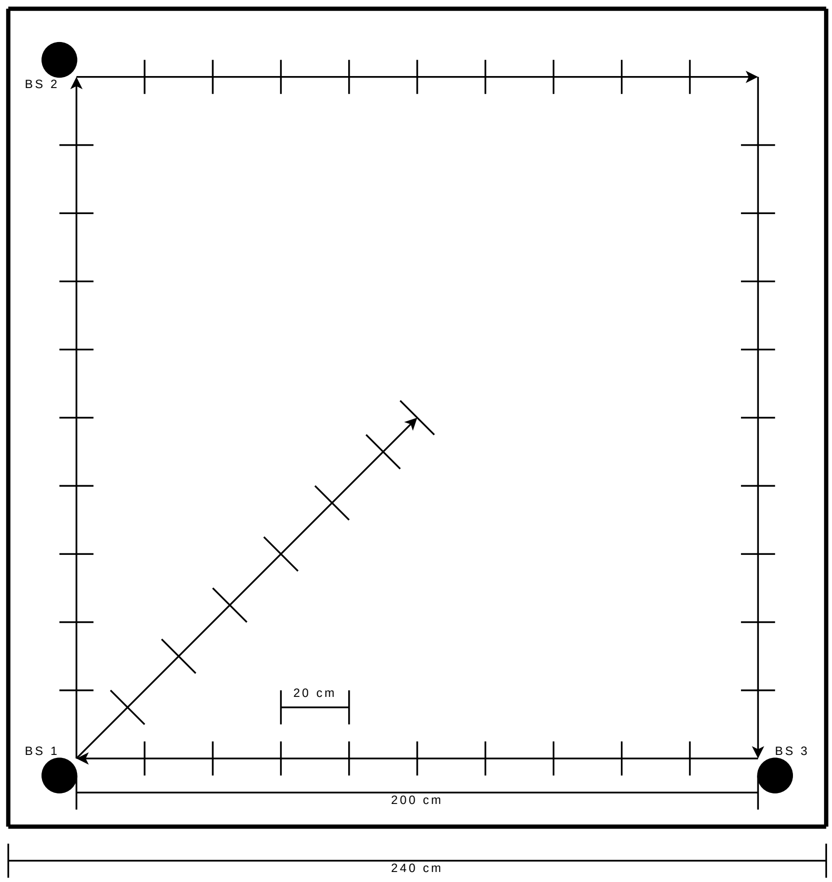


Figure 9.1: *The test setup. The position of the BSs and the path of the MD is shown. It is also indicated where the timestamps were recorded.*

9.2 Test Specification

To test the performance of the two different positioning prototypes, it was necessary to obtain different data during the testing. These are described in the following:

The correct path: It is important to know the correct path of the MD, because this is the reference for the tests. The correct path is obtained by the different positions of the MD with corresponding timestamps.

The measurements: The RSSI measurements are needed as input for the prototypes.

The prototypes: The results from the prototypes must be compared to each other and to the correct path.

The test should be performed according to the following requirements:

- A MD should be tracked by the use of three BSs.

- The MD should move in a predefined path in the room while the BSs record measurements.
- A test should be performed in a room with no obstacles.
- Another test should be performed with random appearing obstacles.
- These tests should be performed several times, where the MD moves along the same path. This way it is possible to compare the results from the different tests.

9.3 Test Results

In this section the results from the different tests will be described and discussed. The test uses the two different positioning prototypes detailed in Chapter 8 to process the data.

The test is based on four different scenarios:

- Track1: Measurements are collected from a fast moving MD.
- Track2: Measurements are collected from a semi fast moving MD.
- Track3: Measurements are collected from a slow moving MD.
- TrackOBS: Measurements are collected from a slow moving MD in a scenario with obstacles.

The RSSI measurement obtained from these 4 Tracks, will be compared to the correct distance from each BS and the result will be discussed. Then the simple prototype and the enhanced prototype will process these measurement and the result will be discussed. This discussion is based on inspection of the plotted tracks. This inspection will be supported by calculations of the mean error between the results form the prototypes and the real path. This is done in the final comparison.

9.3.1 RSSI Measurements

The two steps which are described in Section 6.2.1 are performed on the RSSI measurements from the test. After this the RSSI-values are converted to distances using the formula found in the calibration test. In Figures 9.2-9.4 the distances from BS1-BS3 are shown along with the correct distances.

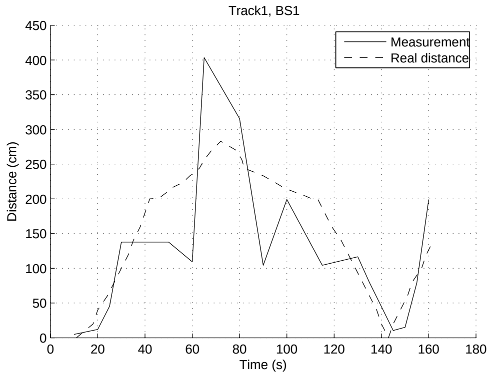


Figure 9.2: *Track1 BS 1*

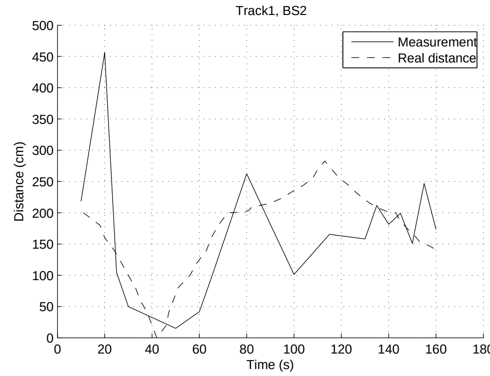


Figure 9.3: *Track1 BS 2*

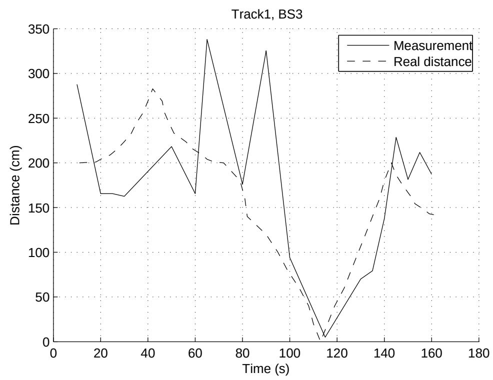


Figure 9.4: *Track1 BS 3*

From the figures it can be seen that even though the measured distances does not fit the correct path exactly, the same tendencies can still be seen except from a few jumps. In Figure 9.2 the peaks fit the correct path, but this is not the case in Figure 9.3 and 9.4. Therefore it must be concluded that the peaks does not always indicates the real ones.

The figures for Track2, Track3 and TrackOBS can be seen in Appendix B.

9.3.2 Trilateration

The trilateration method is applied to the RSSI measurements and the result can be seen in Figures 9.5-9.8 with the correct path as reference. In the figures, a circle is a result of the trilateration in one time window and the solid line represent the track between two calculated positions. The dashed line is the real path.

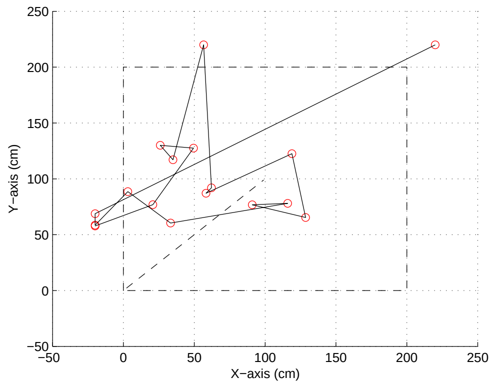


Figure 9.5: *Track1*

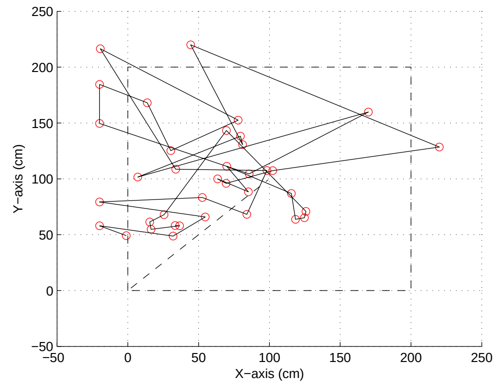


Figure 9.6: *Track2*

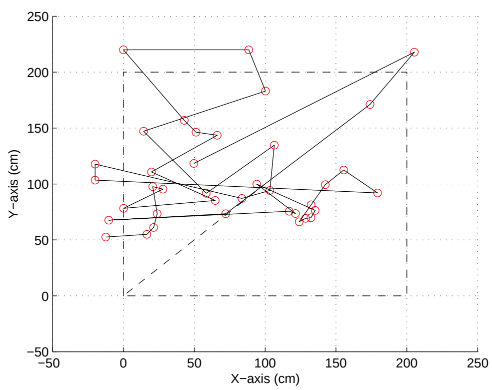


Figure 9.7: *Track3*

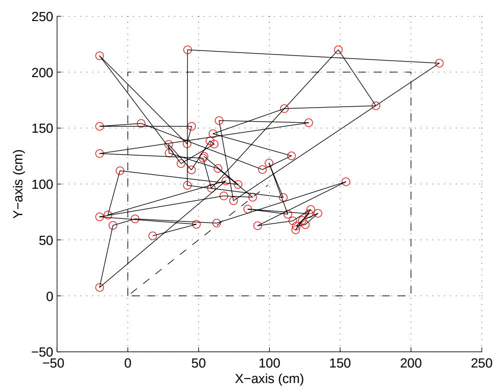


Figure 9.8: *TrackOBS*

As it can be seen in the figures, the estimated path does not fit the correct path so it must be concluded that the measurements contain too much noise for the path to be derived correctly. The calculation of the trilateration is only based on current distances and does not depend on previous positions or velocity of the tracked MD. This can be seen in all figures as the MD is making large jumps from one position to the next. This is because no mobility model for predicting the movement of the MD is used in the trilateration.

It can be concluded that trilateration with no mechanisms to consider history of movement is not sufficient in order to achieve a good estimate of the track, i.e. a track which represents a trend similar to the real path.

9.3.3 Particle Filter using the Less Drunk Model

The RSSI measurements are applied to the enhanced prototype and the result can be seen in Figures 9.9-9.12 with the correct path as reference. In the figures, a circle is a position estimate in one time window and the solid line represent the track between two calculated positions. The dashed line is the true path.

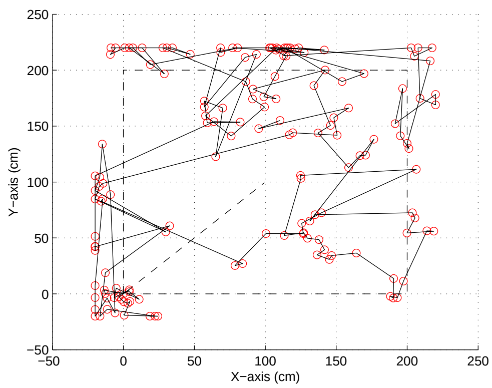


Figure 9.9: *Track1*

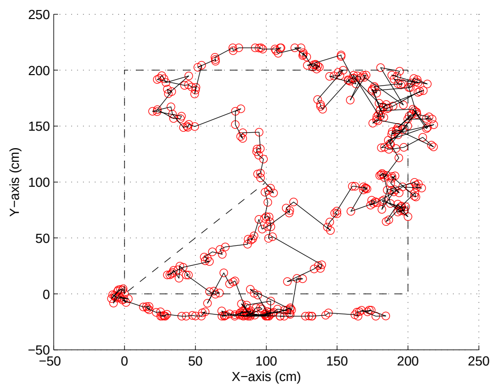


Figure 9.10: *Track2*

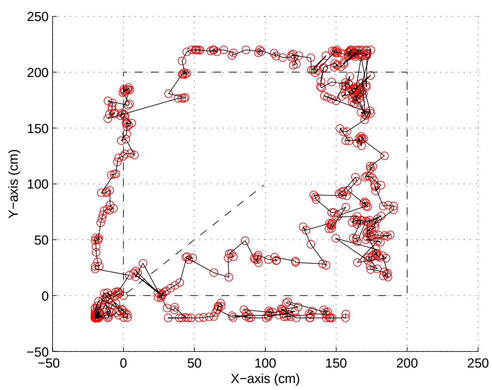


Figure 9.11: *Track3*

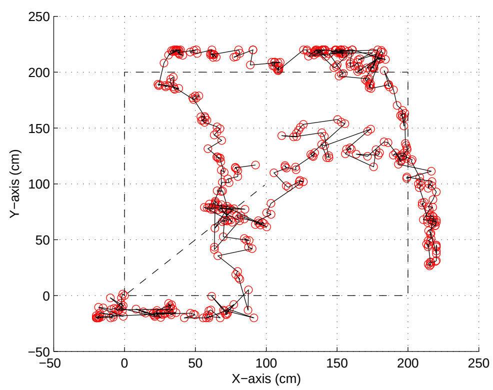


Figure 9.12: *TrackOBS*

As seen in Figures 9.10, 9.11 and 9.12 the Particle Filter is capable of estimating a path that is close to the real one, but performs less precise on Track 1, see Figure 9.9. This is due to the fast moving speed of the MD which causes few measurements to be obtained. From Track3 the measurements are collected from a slow moving MD, compared to Track1 and Track2, hence a better estimation, see Figure 9.11. Measurements from TrackOBS was achieved from a slow moving MD with obstacles such as moving persons, see Figure 9.12. The estimation of the path is less precise than the other tracks due to the random obstacles.

9.3.4 Comparison

The mean error for each track using the two prototypes can be seen in Table 9.1. The errors are obtained by comparing each estimated position to the true one, then the mean of these errors is calculated.

It appears from running the Particle Filter step by step that the estimation is delayed in time compared to the real path. This is due to the recursive nature of this algorithm. A possible way to discard this delay in the error calculation was implemented. The error is calculated by comparing the estimation to the nearest correct position instead of looking at the time stamp. This method is not exactly correct, as long as the error could be calculated for a different position instead of the estimated one when the error is too large. However, from Figures 9.10, 9.11 and 9.12 it can be seen that the estimated path has the trend of the real one. Thus this method could give a meaningful estimation error.

	Track1	**Track2**	**Track3**	**TrackOBS**
Trilateration	92.881	90.523	85.479	80.397
Particle Filter	66.091	65.765	53.627	61.606
Particle Filter considering the delay	12.690	11.256	10.373	11.925

Table 9.1: *Mean error (cm) for the different tracks using either the trilateration or Particle Filter*

The efficiency of the system can be reflected by a Success Rate as well. It is calculated by comparing the error of each estimation to the prototype requirement : 30 cm accuracy, see Section 5.2. This is presented in Table 9.2.

	Track1	**Track2**	**Track3**	**TrackOBS**
Trilateration	5.88	5.26	17.07	5.35
Particle Filter	21.21	26.21	30.39	28.57
Particle Filter considering the delay	100	100	100	100

Table 9.2: *Success Rate in (%) for the different tracks using either the trilateration or Particle Filter*

As it can be seen in the table the enhanced prototype performs better than the simple one in all test cases. For each prototype, the estimation is improving from Track1 to Track3. This is due to the movement speed of the MD. Because it affects the number of measurements obtained which influences the precision of the mean estimate. The number of measurements for each track can be seen in Table 9.3.

	Track1	**Track2**	**Track3**	**TrackOBS**
Trilateration	17	38	41	56
Particle Filter	165	431	559	504

Table 9.3: *Number of measurements used for the trilateration and Particle Filter*

Chapter 10

Conclusion

The main purpose of this project was to design and implement a system capable of performing indoor positioning with Bluetooth as well as minimizing the location errors when using a mobility model.

Through the analysis, different scenarios have been proposed to see how indoor positioning can be used in real life. The Location Stack [Hig02] was used as reference model in the positioning procedure, where different methods of finding the distance from or angle between a Base Station (BS) to a Mobile Device (MD) such as Angle of Arrival (AoA), Received Signal Strength (RSS), Time of Arrival (ToA) and Time Difference of Arrival (TDoA) were examined and described. Some of the problems of radio propagation were also investigated i.e. penetration, diffraction, reflection and multipath propagation to explain how these phenomenas can influence measurements made on wireless signals.

Different methods that utilize the measurements to find the position have been described. These are trilateration, triangulation and multilateration which are located in the Fusion Layer of the Location Stack.

The methods were compared, and based on the analysis and the provided Bluetooth equipment, RSS measurements was chosen to obtain distances and trilateration used to compute the position as a simple approach. In this phase, it was also decided to perform tracking of the MD.

The design has been carried out in two phases, first the simple method of positioning has been designed, then an enhanced method of positioning using the Less Drunk mobility model and the Particle Filter has been designed.

The implementation of the two approaches have been done in MATLAB and executed on a controller. Three Casira Bluetooth devices provided by Cambridge Silicon Radio (CSR) have been used as BSs and a Nokia N70 as MD to test the system.

The implemented algorithms have been validated using a simulated correct data set with added noise. The validation shows that the algorithms are working as intended for both the trilateration and the Particle Filter using the Less Drunk model.

Measurements have been obtained from a test carried out in a 240x240cm square area bounded by walls with three BSs in three of the corners. The MD was then moved around in this setup in a predefined path. Obstacles were added to one of the test to see how they influence the results.

The simple approach with trilateration showed that the measurements were highly influenced by noise and it was not possible to see a clear trend of how the MD had moved in the setup. The Particle Filter gave better results as the estimated path showed a trend corresponding to the correct path.

The accuracy of these prototypes was compared by calculating the error of the estimation according to the real path. From this comparison, it was proved that the results from the enhanced prototype are better than the results from the simple one. Also, this shows that the accuracy depends of the movement speed of the MD due to the number of obtained measurements. The method used to deal with the Particle Filter delay shows that the prototype requirement are fulfilled. However, as discussed in this comparison, this method is not exactly correct. Therefore, further work on this delay must be done to have a more appropriate error calculation.

It should be noted that RSS measurements in Bluetooth is not intended for accurate distance measurements, but for power consumption optimization on a connected Bluetooth link. The rate of measurements is influenced by the frequency hopping in the Bluetooth inquiry phase and this reduces the rate of new position estimates.

It can be concluded that Bluetooth with RSS measurements in a 240x240cm setup with three BSs can be used for location purpose. Both the simple and enhanced approach have proved to be good methods of estimating positions when a simulated correct data set with controlled noise is used. If there is too much noise in the measurements, the Particle Filter should be used to obtain the best estimate of the position.

Chapter 11

Future Perspectives

As stated in the conclusion, the RSS measurements based on Bluetooth inquiry is very noisy and consequently the position of the MD is hard to estimate. In order to get better measurements the equipment could be calibrated to the given scenario regarding the mapping from RSSI to distance or by using a directed antenna when the BS is in a corner, in order to avoid reflections from the walls.

Further investigation of mobility models is needed to find an optimal model for a given scenario. More BSs could also be deployed and then use the nearest ones. Also it is not investigated in this project whether Bluetooth is a good technology for positioning compared to other wireless technologies e.g. wireless LAN. It might be possible to use other wireless technologies and thereby achieve a better precision.

The Particle Filter using the Less Drunk model gives improved results, but needs to be tested in a scenario with more obstacles. The obstacles should also be fixed in order to test the performance of the system e.g. when the MD is behind walls.

It is not possible to completely avoid errors in the positioning using RSS measurements from Bluetooth. There will always be a possibility that the received measurements have too much noise, the measurements are delayed, one or more BS fail to obtain the signal strength etc. Therefore the scenario should be extended to ensure more correct positioning.

There are two simple ways to extend the scenario. One where more sensors (BSs) are added, and one where different kind of sensors are added.

In the first extension more BSs are added ensuring more measurements for each positioning point and thereby decreasing the risk of errors. However it will increase the inquiry period. This problem can be avoided if a new method for measuring RSS is used.

In the second extension different kinds of sensors should be added. The different kind of sensors could contain cameras, pressure sensors placed in the floor, laser ranger or ultrasonic ranger. If more of these kind of sensors where combined this would minimize the errors in the positioning as well. Depending on the sensors they could be weighted differently. E.g. the camera or the laser ranger might be perceived as more reliable than e.g. the RSS of the Bluetooth BS or the ultrasonic ranger. Also futher investigation could be done to compare RSS measurements to other techniques such as ToA, TDoA and AoA.

The implementation of the prototype can also be improved by developing a new complete program including a GUI as described in the design chapter so the MD can be tracked in real time.

Appendix A

Bluetooth

Bluetooth is a low powered technology which enables short-range wireless voice and data communication with other Bluetooth device. The Bluetooth Special Interest Group (SIG) has developed an open specification for Bluetooth wireless communication which will be used as literature throughout this chapter. This specification consists of a *Bluetooth Core System* which covers the four lowest layer of the associated protocol stack.In this chapter there will first be a description of the usage of this technology and its application according to the relevance of this project, followed by a description of the protocol stack defined by the *Bluetooth Core* specification.

Usage and Applications

The usage of Bluetooth can be categorised in three groups; voice/data AP, peripheral interconnect and Personal Area Network (PAN) which is described as follows:

- **Voice/data AP:**
  This category involves connecting a computing device such as a pocket-PC or a laptop (mobile computer) to a cell phone via Bluetooth. For example, a mobile computer with Bluetooth, where the computer uses a cell-phone via Bluetooth to access the internet, in this case the cell-phone acts as a personal AP.
- **Peripheral interconnect:**
  This category of use, involves connecting peripheral devices such as, mouse, keyboard, headset using Bluetooth e.g. to a PC or a mobile computer.
- **Personal Area Network:**
  The last usage is where Bluetooth is used in a PAN, where an ad-hoc network is established between to devices where data exchange may take place.

Bluetooth Core Specification

In this section the Bluetooth Core is described. The *Bluetooth Core* architecture describes the four lowest layers of the Bluetooth protocol stack, and these are; Radio layer, Baseband Layer, Link Manager Layer and the L2CAP Layer, see Figure A.1.

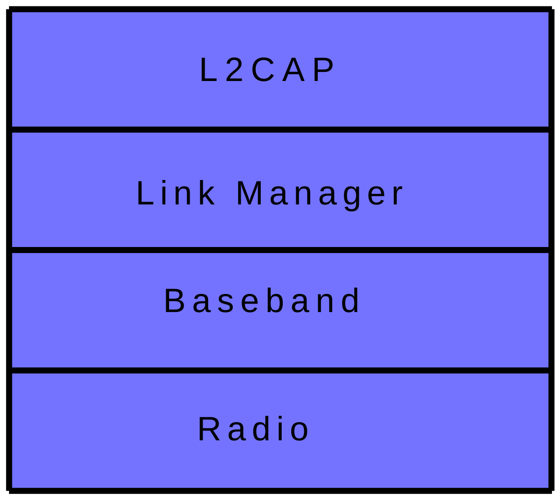


Figure A.1: *The four lowest layer of the Bluetooth Core architecture.*

The Bluetooth Core can be grouped into a two subsystems, the Bluetooth Controller which contains the three lowest layers and the Bluetooth Host which consist the higher layers. The interface of these two subsystems are maintained by the Host to Controller Interface (HCI).

In the following the relevant blocks of the Bluetooth core system regarding to this project will shortly be introduced.

Radio Layer
This subsection is based on [BS07].

The Bluetooth Radio Layer is the lowest specified layer of the Bluetooth Core specification, and it defines the requirement of the Bluetooth transceiver device that operates in the 2.4 - 2.48 GHz ISM band. There are 79 available RF-channels in total, and those are spread by ”frequency hoping” in 79 hops, displaced by 1 MHz. see Equation A.1.

$$f = 2402 + k, k = 0, 1, .., 78 \tag{A.1}$$

Where f is the frequency of a RF-channel, starting from 2402 MHz up to 2480 MHz. The

Power Class	Maximum Output Power	Minimum Output Power
1 (long range 100 meters)	100 mW (20 dBm)	1 mW (0 dBm)
2 (standard range 10 meters)	2.5 mW (4 dBm)	0.25 mW (0 dBm)
2 (short range 10 cm)	1 mW (0dBm)	N/A

Table A.1: *Power classes*

parameter k is the number of ”frequency hoping” which runs from 0 to 78. The RF-channel hoping procedure is handled by the Baseband Layer which lies on top of the Bluetooth Radio Layer.

Transmitter Characteristics

Each transmitting Bluetooth device is classified into three different power classes. These power classes defines the output power a device uses when transmitting data, for example a long range transmission requires more output power in compare to a short range transmission, see Table A.1.

As seen in Table A.1 each device can optionally vary its transmitting power within the defined output power boundary. When transmitting with power over 4dBm, in this case Power Class 1, then the transmitted power needs to be limited, this mean that a Power Class 1 device which transmit for example with pmax (20 dBm) shall be able to control its transmit power down to 4 dBm or less.
This is done by using the *Power Control* mechanism which optimizes the transmission by regulating the output power eg. to optimize the signal interference level or the power consumption level during a transmission. The *Power Control* is handled by the Link Manager Layer, where the RSSI level is measured, and reported back to the Radio Layer if for example the transmission power shall be increased or decreased according to a given situation. The changes in the power level are carried out in steps, where the maximum step size is 8 dB and the minimum step size is 2 db and these steps shall form a monotonic sequence. The Power Control is optional for those power classes which uses an output power level below 4 dBm.

Baseband Layer

The Baseband layer is the equivalent physical layer of the OSI model. This layer lies on top of the radio layer. The Baseband protocol contains Link Controller and the Baseband Resource Manger -blocks. The Link Control block, of the Baseband, is responsible for carrying out link level routine such as link connection and power control. The Link connection consist of two type of links, the Synchronous Connection-Oriented (SCO) link and the Asynchronous Connection-Less (ACL) link.

The SCO link is mainly used to transport voice data, and the ACL is used to transmit non -voice data packets. The Baseband Manger block is responsible for all access to the radio medium and have two main functionalities; creating access contract with other devices and a scheduler. An access contract between devices can be made eg. in order to deliver a certain level of QoS, which may be required for a user application to perform as expected. These access contracts will then be scheduled according to their priority by the scheduler.

Packet Type

There are 13 different packet types defined for the baseband layer. Higher layers use these packets to compose Protocol Data Unit (PDU). The most relevant packets type for this project will be stated in the following itemisation:

- ID
  Is a 68-bit packet that is used in routines such as paging, inquiry and response. This packet fills the first part of the packet stream which is the Access Code part, see Figure A.2.
- FHS
  Frequency Hopping Synchronisation, as it says, is the packet that is used to synchronize channel hopping for two or more connected devices. This packet contains the Bluetooth device address and the clock of the transmitting device.

Packet Format

The Packet consists of 3 parts; the access code (68/72-bits), the header (54-bits), and the payload (0-2745-bits), see Figure A.2.

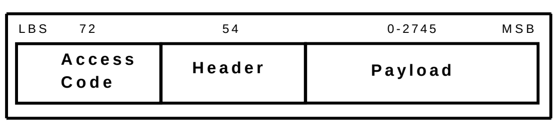


Figure A.2: *Packet format*

Access Code: There are three different type of access code: Channel Access Code (CAC), Device Access Code (DAC) and Inquiry Access Code (IAC). These access codes are used for eg. to time the data synchronisation, paging/response and device inquiry (packet are defined by the ID-packet type).
Header: Contains the ordering of the packet, information for packet acknowledgment, flow control, slave address and header error check.
Payload: This part can either consist of voice field or data field or even both. In case of the payload consist of a data field then there will also be a payload header.

Link Manager Layer

The Link Manager Layer controls and negotiates all the aspect that involves connection between two Bluetooth devices. This means that carrying out link setup, authentication and link configuration between two devices. This is carried out by a number of PDU's, in the following only those PDU's which are relevant for this project will be stated.

- Power Control
  As mentioned earlier, this PDU's is used for controlling the power level the transmission (tx) eg. if the RSSI value differs too much from the threshold of a device, then that device can send a request to the transmitting device by sending a Power Control PDU with the request for a decreases or increases of the TX power level.

Host Controller Interface

The Host Controller Interface (HCI), provides a service interface between the Bluetooth Controller and the Host. Figure A.3 shows an overview of the interaction between the controller and the Host.

HCI Firmware is embedded on a Bluetooth device and this firmware implements the HCI-commands which give access to the Bassband layer.

HCI Driver is located on the Host, and will notify the Host if it detects any event "something" has occurred eg. a packet of some kind. In this case the packet will be passed to the higher layer (Hosts) in order to determine the occurred event.

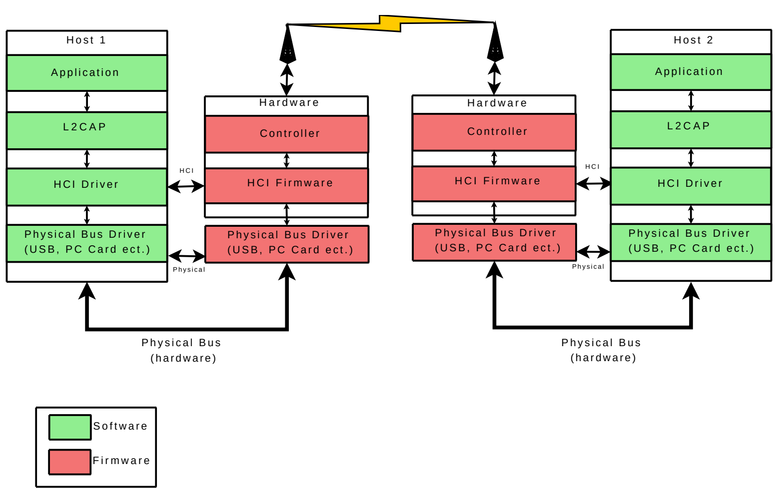


Figure A.3: *Interface between the Bluetooth Controller and the Host*

Logical Link Control and Adaptation Protocol

The Logical Link Control and Adaptation Protocol (*L2CAP*) is the equivalent Data Link Layer of the *OSI* model and placed over the Baseband Layer. This layer supports both a connection-oriented and connectionless data service to upper layers. This is carried out by using Channel Identifiers (*CIDs*). A *CID* represents the local names of the logical channel an end-point device may have. The *L2CAP* layer may have multiple *CIDs* representing multiple end-point devices. When considering the Connection-oriented representation then the *L2CAP* layer uses the *CIDs* to identify and to establish a connection to an endpoint device. In contrast the connection less representation the *CID* on the source side, represent one ore more remote devices and in this case the data flow is restricted to a single direction. On Figure A.4 the *L2CAP* state machine is depicted.

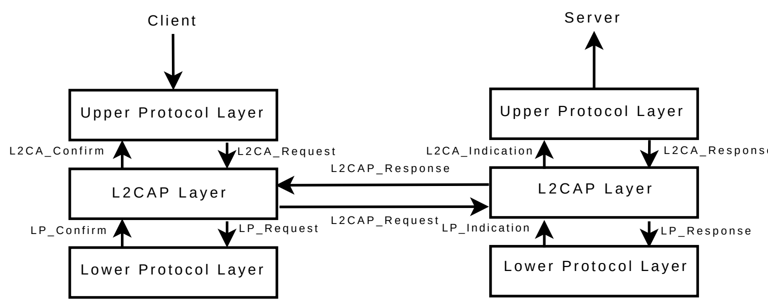


Figure A.4: *Interface between the Bluetooth Controller and the Host*

The Figure A.4 illustrates an example of a client/server connection, where a request message is initiated by the client. The upper layer, of the client device, uses the services of the lower layer in order to deliver the request message to the server. This is carried out by sending a L2CAP_Request. The lower layer, on the server device, sends an indication message to the upper layer about the request. The Upper layer response the indication to the lower layer and finally the client will receive a confirm message.

Appendix B

Test Results

In this appendix the test results from test 2, 3 and with obstacles (OBS) can be seen. The figures correspond the ones from test 1 in Section 9.3.1.

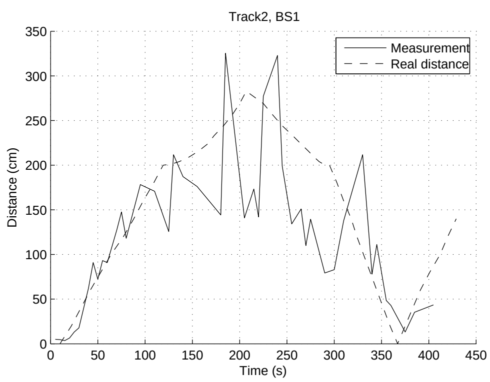


Figure B.1: *Track2 BS1*

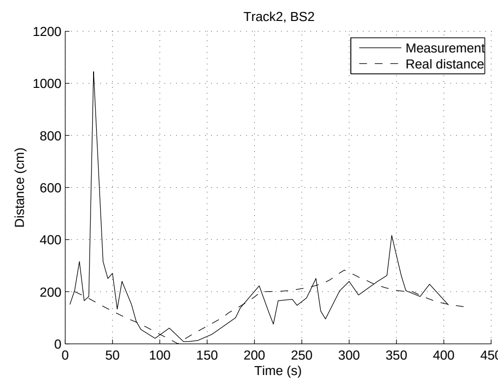


Figure B.2: *Track2 BS2*

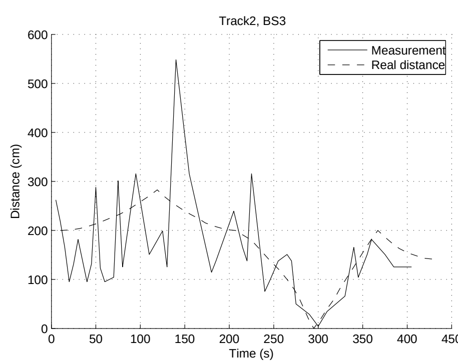


Figure B.3: *Track2 BS3*

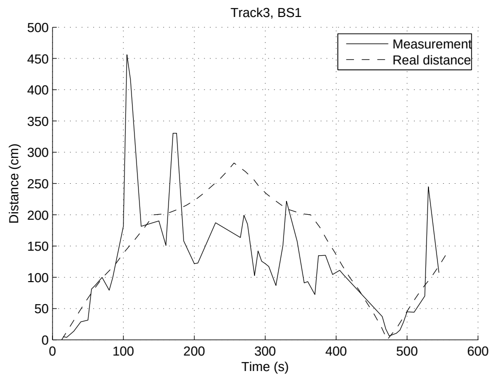


Figure B.4: *Track3 BS1*

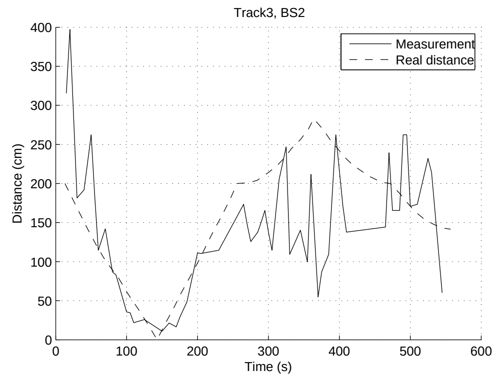


Figure B.5: *Track3 BS2*

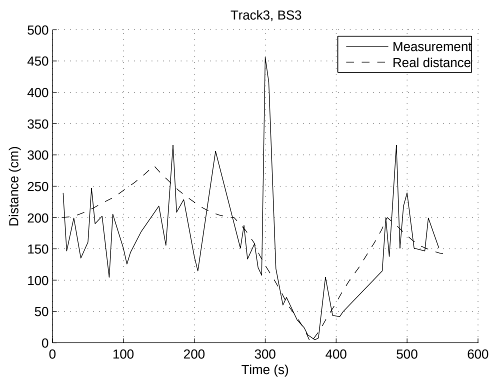


Figure B.6: *Track3 BS3*

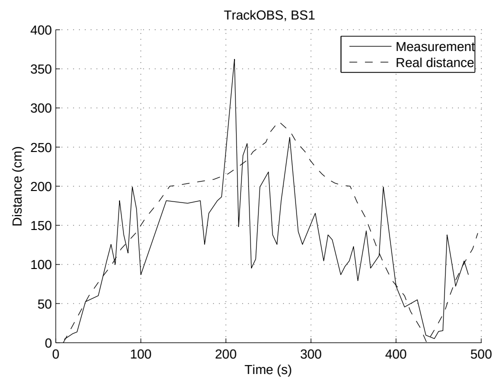


Figure B.7: *TrackOBS BS1*

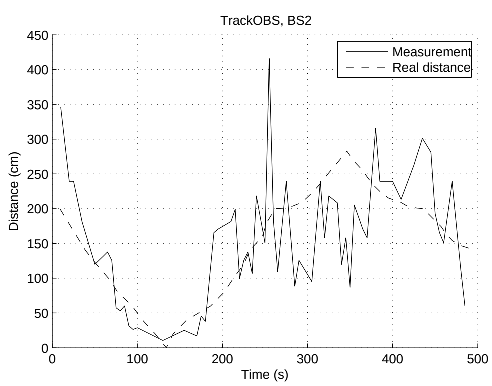


Figure B.8: *TrackOBS BS2*

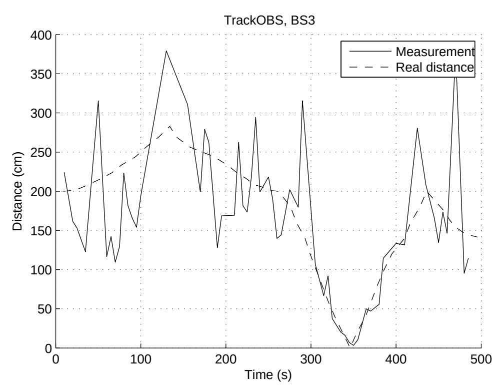


Figure B.9: *TrackOBS BS3*

Appendix C

Mathematical Proofs

Proof 1

In this section, the relation $P(X_{n+1}^{(m)}|X_n^{(i)}, O_n, ..., O_1) = P(X_{n+1}^{(m)}|X_n^{(i)})$ used in the conceptual solution of the Particle Filter will be proved. As a simplification the states $X_{n+1}^{(m)}$ and $X_n^{(i)}$ will be noted X_{n+1} and X_n.

$$P(X_{n+1}|X_n, O_n, ..., O_1) = \frac{P(X_n, O_n, ..., O_1)}{P(X_{n+1}, X_n, O_n, ..., O_1)}$$

And:

$$\begin{aligned} P(X_{n+1}, X_n, O_n, ..., O_1) &= \sum_{X_{n-1}} ... \sum_{X_1} P(X_{n+1}, ..., X_1, O_n, ..., O_1) \\ &= \sum_{X_{n-1}} ... \sum_{X_1} P(O_n|X_{n+1}, ..., X_1, O_{n-1}, ..., O_1) \cdot P(X_{n+1}, ..., X_1, O_{n-1}, ..., O_1) \end{aligned}$$

From the equation 7.3:

$$P(O_n|X_{n+1}, ..., X_1, O_{n-1}, ..., O_1) = P(O_n|X_n)$$

Thus:

$$P(X_{n+1}, X_n, O_n, ..., O_1) = \sum_{X_{n-1}} ... \sum_{X_1} P(O_n|X_n) \cdot P(X_{n+1}, ..., X_1, O_{n-1}, ..., O_1)$$

Repeating this same procedure for $O_{n-1}, ..., O_1$ leads to:

$$P(X_{n+1}, X_n, O_n, ..., O_1) = \sum_{X_{n-1}} ... \sum_{X_1} \left(\prod_{j=1}^{n} P(O_j|X_j) \right) \cdot P(X_{n+1}, ..., X_1)$$

And:

$$P(X_{n+1}, ..., X_1) = P(X_{n+1}|X_n, ..., X_1) \cdot P(X_n, ..., X_1)$$

Therefore:

$$P(X_{n+1}, X_n, O_n, ..., O_1) = \sum_{X_{n-1}} ... \sum_{X_1} \left(\prod_{j=1}^{n} P(O_j|X_j) \right) \cdot P(X_{n+1}|X_n, ..., X_1) \cdot P(X_n, ..., X_1)$$

From the equation 7.4:

$$P(X_{n+1}|X_n, ..., X_1) = P(X_{n+1}|X_n)$$

Thus

$$\begin{aligned} P(X_{n+1}, X_n, O_n, ..., O_1) &= \sum_{X_{n-1}} ... \sum_{X_1} \left(\prod_{j=1}^{n} P(O_j|X_j) \right) \cdot P(X_{n+1}|X_n) \cdot P(X_n, ..., X_1) \\ &= P(X_{n+1}|X_n) \cdot \sum_{X_{n-1}} ... \sum_{X_1} \left(\prod_{j=1}^{n} P(O_j|X_j) \right) \cdot P(X_n, ..., X_1) \end{aligned}$$

On the other hand, using the same reasoning leads to:

$$\begin{aligned} P(X_n, O_n, ..., O_1) &= \sum_{X_{n-1}} ... \sum_{X_1} P(X_n, ..., X_1, O_n, ..., O_1) \\ &= \sum_{X_{n-1}} ... \sum_{X_1} \left(\prod_{j=1}^{n} P(O_j|X_j) \right) \cdot P(X_n, ..., X_1) \end{aligned}$$

Thus, the equations C.1, C.1 and C.1 give:

$$P(X_{n+1}|X_n, O_n, ..., O_1) = P(X_{n+1}|X_n)$$

Which is what is needed to be proved.

Proof 2

In this section, the relation $\forall t \geq 0, \forall T \geq 0 : P(N(t+T) - N(t) \geq 1) = 1 - e^(- \lambda T)$, where $\{N(t)/t \geq 0\}$ is Poisson process with associated rate parameter λ, will be demonstrated:

$$P(N(t+T) - N(t) \geq 1) \quad = \quad P\left(\bigcup_{k=1}^{+\infty} (N(t+T) - N(t) = k)\right)$$

All the events $(N(t+T) - N(t) = k)$ for $k \geq 1$ are mutually exclusive, Thus:

$$P\left(\bigcup_{k=1}^{+\infty} (N(t+T) - N(t) = k)\right) = \sum_{k=1}^{+\infty} P(N(t+T) - N(t) = k)$$

Therefore:

$$\begin{aligned}
P(N(t+T) - N(t) \geq 1) &= \sum_{k=1}^{+\infty} \frac{e^{-\lambda T} (\lambda T)^k}{k!} \\
&= e^{-\lambda T} \cdot \sum_{k=1}^{+\infty} \frac{(\lambda T)^k}{k!} \\
&= e^{-\lambda T} \cdot \left(\sum_{k=0}^{+\infty} \frac{(\lambda T)^k}{k!} - 1 \right) \\
&= e^{-\lambda T} \cdot (e^{\lambda T} - 1) \\
&= 1 - e^{-\lambda T}
\end{aligned}$$

Which is what is needed to be proved.

Appendix D

Acronyms

AP Access Point

BS Base Station

BCHS Blue Core Host Software

CSR Cambridge Silicon Radio

GPS Global Positioning System

MD Mobile Device

RSS Received Signal Strength

RSSI Received Signal Strength Indicator

TSS Transmitted Signal Strength

LOS line of sight

NLOS non line of sight

SD Service Discovery

AoA Angle of Arrival

ToA Time of Arrival

TDoA Time Difference of Arrival

Bibliography

[AFP08] Physorg.com AFP. High-tech shopping carts to be deployed in singapore, Februrary 2008.

[BR04] Niel Gordon Branko Ristic, Sanjeev Arulampalam. *Beyond the Kalman Filter - Particle Filters for Tracking Applications*. Artech House Publishers, 2004. ISBN: 1-58053-631-x.

[BS07] Inc. Bluetooth SIG. *Core Specification v2.1 + EDR*. `www.bluetooth.com`, 26 July 2007.

[BS08] Inc. Bluetooth SIG. *Bluetooth Technologie Web site*. `http://www.bluetooth.com/Bluetooth/Technology`, 10 March 2008.

[Dye01] Stephen M. Dye. *Mobile Positioning*. Mobile Lifestreams, 2001. ISBN: 1929105398.

[Fig07] Joao Carlos Prazeres Figueiras. *Accuracy Enhancements for Positioning of Mobile Devices in Wireless Communication Networks*. PhD thesis, Aalborg University, 2007.

[Fig08] Joao Carlos Prazeres Figueiras. Wireless networks - positioning aspects, March 2008.

[Gol05] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005. ISBN: 978-0-521-83716.

[Hig02] Jeffery Hightower. The location stack: A layered model for location in ubiquitous computing. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002)*, 2002.

[Hig03] Jeffery Hightower. The location stack. In *Intel Research*, 2003.

[JA95] S. Yoshida JB Andersen, TS Rappaport. Propagation measurements and models for wireless communications channels. *Communications Magazine, IEEE 33*, (1):42–49, 1995.

[Med08] MediaCart. Mediacart website, March 2008.

[Par01] J. D. Parsons. *The Mobile Radio Propagation Channel*. John Wiley & Sons Ltd., 2001. ISBN: 1929105398.

[RO07] Lars Jessen Roost and Michael Østergaard. Simultaneous localization and mapping for wireless networks. Master's thesis, Aalborg University, 2007.

[Sch] Henrik Schiøler. Less drunk mobility model.