Technical report, IDE0424, June 2004

Integration of positioning capability into an existing active RFID-system

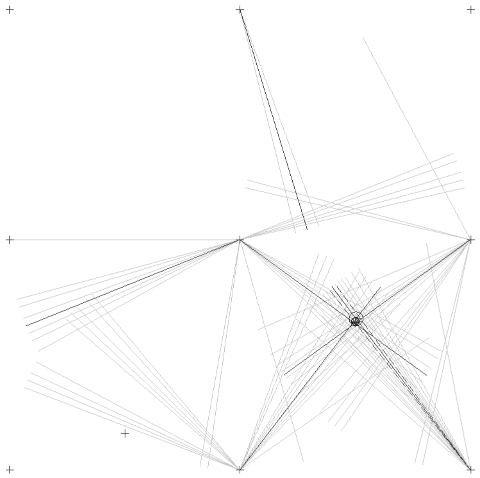
Master's Thesis in Computer System Engineering

submitted by

Andreas Franz

Wolfgang John

Philipp Nagele



School of Information Science, Computer and Electrical Engineering
Halmstad University

Acknowledgement

We would like to express our deepest gratitude to our families, which made our studies possible. Furthermore we want to thank our supervisor Pelle Wiberg for his support and helpful ideas throughout the whole project. Special thank goes out to Urban Bilstrup for inspiration during the topic research.

Jörgen Carlsson and especially the ISC supported us in many concerns and relieved us of many administrative difficulties, which arise while studying in a foreign country. Finally we want to thank all the exchange students in Halmstad during this year for all their patience and support.

Description of cover page picture: Example for the operational sequence of the developed positioning algorithm.

Preface

Team Members:	Andreas Franz Wolfgang John Philipp Nagele
Institution:	Högskolan Halmstad
Program of Study:	Master's programme for computer system engineering
Titel of thesis:	Integration of positioning capability into an existing active RFID-system
Supervisor:	Per-Arne Wiberg, Tech. Lic. CE

Keywords

1. keyword:	positioning system
2. keyword:	active RFID
3. keyword:	AoA-based

Abstract

The potential of active RFID systems is still under-utilised. Therefore this project sets out to design and simulate positioning functionality which fits into an existing RFID system. The system is dimensioned for indoor use in a warehouse-like non line of sight environment. Due to the lack of computation power within the RFID tags a centralized system which computes geometrical rawdata is proposed. An algorithm is implemented in Matlab for obtaining position information out of AoA-data. Relevant parameters for the communication protocol are optimised with the help of simulations. Furthermore, the position algorithm is evaluated for its abilities within different setups. Stringent hardware restrictions limit the system to quasi-static operation. The system ensures scalability regarding beacon placement. The tests reveal an accuracy of 20.6 cm in average assuming optimal conditions.

Contents

List of Figures

List of Tables

1 Introduction

With the fast changing needs of the retail industry, also the need for keeping track of what is produced, stored, handled, etc. is increasing. This need for information leads to several innovations in the automatic identification industry (AIM), which address many of these needs. The AIM is offering a couple of solutions to cope with the tremendous information needs. One of the best known applied solutions is the barcode technology. A development of this approach is based on radio frequency identification, generally known as RFID, which offers some enhancement to former automatic methods. This kind of identification is contact-less and does not require line of sight (like barcodes). For instance, successful operation is possible through sunlight, wetness, coldness, frost, dirt, grease and corrosive chemicals. Additionally, this technology allows to manipulate the stored data in the tags, which are fixed on the goods, which is not possible e.g. for barcode technology.

During the year 2003, big corporations in the U.S. (like Wal-Mart or the U.S. Department of Defense) started to incorporate RFID technology into their supply chains or announced the use of RFID for the next years. Also in Europe the Metro Group started a field trial called Future Store [1] for the use of RFID tags to optimise their supply chain management. The efforts of these companies are based on forecasts to be able to reduce considerably costs in their supply chain. "Reduced labor expenditures are an annually recurring benefit, estimated at 7,5% of warehouse labor" [2], which would result in savings of up to 6 billion Euro p.a. only for the German retail industry according to an actual study of A.T. Kearny [2].

Although at the moment the business attention lies on simple and cheap RFID tags, also more complex active RFID chips will profit from the deployment. Up to now, active RFID applications include automated tasks like temperature logging, inventory control or access control. Even though RFID technology nowadays is widely used in warehouse and retail environments, the need for positioning of items has not been addressed in the past. Nevertheless, this information can be crucial for a number of applications. With GPS a world-wide established solution for positioning already exists. However, GPS is not applicable in indoor environments like warehouse and retail facilities.

This project therefore sets out to design and simulate a communication protocol which fits into an existing RFID system. Based on this protocol an algorithm for calculating position coordinates within a certain area should be developed. A detailed simulation should verify the concept and should provide an insight into various system parameters. Based upon the simulation a reviewed system concept should be gathered and should give the basic ideas for the future implementation of a prototype of the positioning system.

The theoretical fundamentals for this thesis are presented in the background chapter, which deals with the RFID technology and positioning. Similar concepts and already deployed systems are described in the related work section. The proposed model of the positioning system and positioning algorithm is then explained in chapter 4 and builds together with the simulation setup and the simulation results in chapter 5 the main part of this thesis. Finally the lessons learned by the simulation are illustrated and discussed.

2 Background

This chapter provides readers with the most important theoretical fundamentals about RFID technology and positioning. A more detailed explanation about specific technologies applied within this project gives the basic background for further reading of the paper.

2.1 RFID

"RFID (radio frequency identification) is a technology that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency (RF) portion of the electromagnetic spectrum to uniquely identify an object, animal, or person." [3] The applied frequencies can lie in different spectra, including low frequencies (LF) centred around 125 kHz, high frequencies (HF) at 13.56 MHz and ultra high frequencies (UHF) ranging from 400 to 1000 MHz and finally microwave, in the range of 2.45 and 5.8 GHz. Every frequency range has its pros and cons concerning transmission range, bandwidth and costs.

Independent of the used frequency, every RFID system consists of two major components, the reader and the tag, as illustrated in Figure 2.1. These components work together to ensure wireless identification of objects. The reader can either be mounted on a fixed position or be accomplished as a handheld device, whereas the tags are normally fixed on goods, containers, etc. which should be identified and tracked. One of the basic functions of a reader is to provide a radio field, either as low radio frequency magnetic field or as high frequency electromagnetic wave (in the case of UHF or microwave). The low frequency magnetic fields radiate from the reader via the transmitting antenna, which is usually realised in form of a coil for low frequencies. The magnetic field of the reader acts then as a power source for the RFID tag, provided by inductive coupling. In a high frequency system, the reader transmits an electromagnetic wave, which propagates outwards in a spherical wave front. RFID tags which are situated within this field are exposed to this propagating wave and are in this way able to collect some of the energy as it passes by.

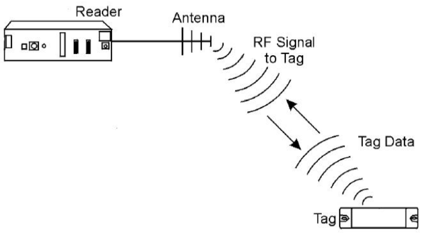


Figure 2.1: Basic RFID architecture (for microwave RFID systems) [4]

This procedure is referred to as propagation coupling [4]. To make the described coupling possible, the RFID tag has to contain an antenna (or coil). Additionally, a tag consists of a small integrated circuit (IC). The required amount of energy for operating is gathered via the antenna or coil when exposed to a magnetic field or electromagnetic wave, as described above. This energy is converted to electrical power and builds the power source for the IC.

While being powered, the IC is able to transmit the stored data of its memory in form of electromagnetic waves to the reader. When the signal from the tag is received by the reader's antenna or coil, it can be detected and processed by the reader. After this processing the signal can be fed into a back-end system and utilized for any kind of application, like identifying, tracking or comparing. Tags which are powered by the energy emitted by a reader are referred to as passive RFID tags.

2.1.1 Active RFID Tags

In contrast to passive tags, active tags are powered by an internal power source (battery) and are often read/write devices. Cells are usually used as batteries which offer a very good power performance compared to their size and weight. The typical temperature range for proper operation is from -50°C up to +70°C, which is no limiting factor for the most common applications.

The advantage of unlimited lifetime for passive tags turns into a disadvantage for active tags, because a power supply in form of a battery means that a tag has finite lifetime. In good systems with a suitable battery, powering well programmed low power ICs, the lifetime can be 10 years and more, always depending on temperature and the number of read/write cycles. Active tags are bigger in size and more expensive because of their integrated power cells. In return they allow greater ranges, are less prone to electromagnetic noise and offer higher frequency modes, which leads to higher transmission rates.

2.1.2 Microwave RFID Systems (2.45 GHz)

The 2.45 GHz frequency lies in the globally recognized ISM band, which was reserved internationally by the ITU for non-commercial use of RF electromagnetic fields for industrial, scientific and medical purposes. The ISM band is among other things used for WLAN (IEEE 802.11b,g) and Bluetooth applications.

The basic operating principle of microwave RFID systems is based on electromagnetic radio signals. Because of the nature of electromagnetic waves, microwave signals are attenuated and reflected by materials containing water or human tissue, and entirely reflected by metallic objects.

Microwave systems allow high data rates due to the use of high frequencies and the possible allocation of broad spectra. For active RFID systems data rates are not influenced by the operating range of the system, whereas passive systems provide lower data rates due to low power availability for the chip. Microwave systems can ensure data rates up to 1 Mbit/s. Typically, rates between 10 and 50 kbit/s are in use.

The operation range for passive tags is about 0.5 meters, which is less than for UHF systems. This is due to the fact that for shorter wavelengths antennas can collect less energy and attenuation and free space loss are higher. For active tags, ranges beyond 30 meters are possible. Compared to low frequency systems using inductive coupling, microwave systems have longer operation ranges, higher data rates and offer more flexibility in the tag design (higher frequencies require smaller antennas). On the other hand, inductive systems offer higher robustness for operating in non line of sight environments.

Overall, microwave RFID systems provide solutions at comparable cost and complexity levels, greater range and higher data rates than inductively coupled counter parts. They find use in active or passive form in electronic toll collection, supply chain management or access control (especially for trucks and vehicles).

2.2 Positioning

A deeper look into literature revealed that any existing positioning or localisation approach basically divides the process of obtaining a position into two phases: location sensing and calculation on the resulting values by positioning algorithms. The following parts give an overview of the methods used for both phases, following [5] and [6].

2.2.1 Location Sensing

Generally, there are three basic methods for obtaining distance measures. A distance can either be gained directly by measuring or indirectly by carrying out additional calculations. The following paragraphs summarise the fundamentals of the three main methods.

Received Signal Strength Indicator (RSSI) This method measures the power of a signal on the receiver side. Because the transmitting power is known, the effective propagation loss can be detected. When comparing the propagation loss to theoretical models for the particular environment, an estimated distance between sender and receiver can be achieved. RSSI is mainly used for RF signals. Even though a positioning system for RFID should be developed, RSSI is not a suitable solution for the project due to the fact that the used RFID tags do not support signal strength measurements. Furthermore RSSI requires substantial configuration and is therefore not applicable for fast changing environments.

Time based methods These methods include *time of arrival* (ToA) and *time difference of arrival* (TDoA) measurements. Based on the known propagation speed of the carrier wave, measured time can be translated into distance. Although this methods can be used for almost all types of signals (including RF, acoustic, infrared) it is not applicable for the project because time based methods imply an accurate time synchronisation between sender and receiver which is not provided by the existing RFID system.

Angle of Arrival (AoA) AoA methods, in comparison to RSSI or time based methods, do not measure the distance between sender and receiver itself, but calculate the position based on geometric relationships given by measured angles to known positions. This approach does neither require time synchronization nor sophisticated signal strength measurements on the tags, and is therefore the most suitable method for the project.

2.2.2 Calculation of Position

Positioning algorithms represent the second phase of any localization approach. Different calculation methods for retrieving the position can be divided into three main groups.

Hyperbolic trilateration This approach is based on distance measurements and simply retrieves the position by calculating the intersection of at least three circles with radius of the measured distance.

Maximum Likelihood (ML) ML estimation localises a tag by minimising the differences between the measured distance and the distance from the estimated position to known readers. This approach has the advantage to be scalable for any number of known positions (readers) and can incorporate weights for handling reliabilities of values. Because no direct distance information between reader and tag is available according to the design considerations, the above described methods were not taken into account. Nevertheless the ideas of scalability and weighting of the ML approach were considered for the positioning algorithm.

Triangulation Triangulation is the obvious method for calculating intersection points with AoA data and therefore gives the base for the chosen positioning algorithm.

This method computes positions based on trigonometry laws. Figure 2.2 illustrates the basic concept of triangulation. At least two straight lines, g and h, are required to calculate an intersection point respectively a position. The straight lines are defined by:

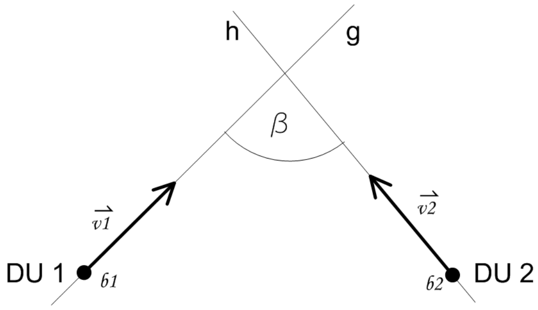


Figure 2.2: Basic principle of triangulation

$$g : \begin{pmatrix} b1_x \\ b1_y \end{pmatrix} + \lambda \cdot \begin{pmatrix} v1_x \\ v1_y \end{pmatrix} \tag{2.1}$$

$$h : \begin{pmatrix} b2_x \\ b2_y \end{pmatrix} + \mu \cdot \begin{pmatrix} v2_x \\ v2_y \end{pmatrix} \tag{2.2}$$

$\vec{b1}$ represents the position vector and $\vec{v1}$ the directional vector of the first signal. The calculation of the intersection angle β between the two vectors is:

$$\beta = \arccos\left(\frac{\vec{v1} \cdot \vec{v2}}{|\vec{v1}| \cdot |\vec{v2}|}\right) \tag{2.3}$$

μ and λ define the dilation of the vectors, which can either be positive or negative. They can be analysed to make an assertion if the intersection point is calculated with a negative direction vector. The formulas for the calculation are:

$$\mu = \frac{(b1_x - b2_x) \cdot v1_y + (b2_y - b1_y) \cdot v1_x}{v1_y \cdot v2_x - v1_x \cdot v2_y},\ \lambda = \frac{b2_x - b1_y + \mu \cdot v2_y}{v1_y} \tag{2.4}$$

If $v1_y$ is 0, the following formula is used to avoid a zero-division:

$$\mu = \frac{b1_y - b2_y}{v2_y},\ \lambda = \frac{b2_x - b1_x + \mu \cdot v2_x}{v1_x} \tag{2.5}$$

Finally, the intersection coordinates are calculated:

$$x = b1_x + \lambda \cdot v1_x \tag{2.6}$$

$$y = b1_y + \lambda \cdot v1_y \tag{2.7}$$

3 Related Work

The following chapter gives an overview of existing papers describing positioning systems for mobile computing applications and wireless sensor networks, partly following [5], [7].

3.1 Positioning in Mobile Computing Applications

Since the 1970s, RF signals were used for positioning systems. The first application field was automatic vehicle location (AVL) for police and military transportation. In this early system fixed base stations use ToA and TDoA methods to estimate distances. The actual position is then calculated using Taylor Series Expansion to linearise the optimization problem. [8] [9]

In the 1990s, location for RF based systems became a big topic due to the requirement of the U.S. Federal Communications Commission (FCC) that all wireless service providers should be able to supply Emergency 911 (E911) services with positioning information. For this purpose the cellular base stations use TDoA for distance estimations and positions are then calculated using least square methods. The system described in [10] only provides an accuracy of 125 meter in 67 percent of time. Due to these values combined with the fact that ToA methods are used within a cellular system covering very large areas, this approach was not followed in more detail.

An indoor tracking system which is more comparable to present needs is the RADAR system [11]. This RF based locating system uses a RSSI method from three fixed stations on a floor. This information is sent to a central server, where the signal strength values are interpreted as distance according to pre-generated signal strength maps. The accuracy of this system is about 3 meters on a floor of 20x40 m, with the drawback of complex initial effort for generating the maps within the building. As mentioned before, signal strength measurements are not applicable and therefore the RADAR system mainly provided basic architectural ideas like the use of fixed base stations and a centralised system.

Another indoor localization system is called Cricket [12]. As great difference to other systems it additionally uses ultrasound instead of solely RF signals for localization purposes. Cricket also uses fixed base stations (called beacons) inside the building to provide listening nodes with necessary information. Ultrasonic time of flight data is used in combination with a RF control signal to gain a position of the listener nodes within 1.5 m^2.

The BAT system [13] is a further development of ultrasound based approaches. Here the nodes transmit ultrasound signals which are received by ceiling-mounted readers. With the help of so-called multilateration accuracy of up to a few centimetres can be achieved.

For the purpose of this project, ultrasound is no alternative because the project relies on the existing hardware provided by Free2Move, where ultrasound components are not included.

Active Badge [14] is an indoor positioning system for persons based on infrared technology. It consists of a cellular proximity system, where persons within the system have to wear an infrared transmitter. Fixed sensors receive the infrared signals and send them to a central server. Because it is a proximity system, the accuracy is depending on the current room size. Like mentioned before, this project focuses on RF based systems and therefore the Active Badge approach is not suitable for future considerations.

3.2 Positioning in Wireless Sensor Networks

Apart from the above mentioned systems based on fixed base stations, there exist also a number of papers dealing with positioning within so called wireless sensor networks in ad-hoc fashion. Wireless sensor networks assume connectivity via RF between all nodes or sensors, which is the main difference to the above described approaches. The proceeding paragraphs describe the papers classified as most relevant, partly following [15].

One approach which is based on connectivity between nodes is the GPS-less system [16]. Here, a grid of nodes with known positions is set up as reference which transmit periodic beacon signals. Unknown nodes set their own position to the centroid of the positions of all connected reference points. The accuracy is therefore about one third of the distance between the reference points. This implies a high number of known positions in order to achieve a good accuracy.

The convex position estimation paper [17] describes an approach, which also uses connectivity between nodes to model a set of geometric constraints which are used to estimate the positions of unknown nodes. The accuracy depends mainly on the number of known positions at the initial phase (anchor nodes). Simulations showed that the estimated positions are very close to the actual positions. [17] performs this estimation calculation on a centralized system.

The Ad-hoc positioning system (APS) [18] on the other hand describes a completely ad-hoc and decentralized system. APS is based on a distance vector (DV) hop approach in combination with GPS. After an adjustable number of anchor nodes within the sensor network retrieves their absolute coordinates via the Global Positioning System GPS [19] they send their position out to all sensors. Each unknown node stores the minimum number of hops to at least three anchors and calculates the distance between these anchors. With this data it converts the hop counts into distances by dividing the distance through the number of hops. Triangulation on three distances to anchor nodes finally gives the position for the own node. The accuracy depends strongly on the number and topology of the anchor nodes.

Another approach based on a dense distribution of anchor nodes is described in [5]. The dynamic fine-grained localization paper proposes a connection for every node to at least three anchor nodes. Via triangulation the position of the node is computed and the sensor is considered as anchor node for the next iteration of the algorithm. After a number of iterations all nodes have position coordinates which's quality is depending on the number of initial anchor nodes.

Very similar to the just described models another approach is discussed in [15] which estimates distances using Hop-TERRAIN, a distance vector method. An iterative calculation is also proposed, but in contrast to [5] a confidence value is introduced as weighting for the quality of each anchor node. This value is recalculated for every iteration based on distance to initial anchor nodes and is increased for every refinement iteration. The results of the simulations showed a comparatively good accuracy, still strongly depending on the number of initial nodes with known position.

More recent papers describe approaches which need a smaller number of anchor nodes. Two of these approaches are very similar and give a good idea of how to set up coordinate systems for nodes. In [20] and [21] a single node starts building a relative map by measuring the distance to the neighbouring nodes. One of these nodes is then considered as second node and spans together with the initializing node one axis of the map. Via triangulation of the distance to a third node, a relative coordinate system can be set up. Adjacent relative maps can then be combined to one map with absolute positions by aligning the relative maps. The main error for this type of positioning is given by the calculations when combining different maps.

All the above mentioned localisation systems for wireless sensor networks give a good overview about positioning algorithms, but are not applicable for this project because no connectivity between tags is available in the existing RFID system. Nevertheless, one approach gives a very good discussion about positioning using AoA within wireless sensor networks. [22] proposes a system which requires at least three fixed beacon nodes. These beacon nodes send out beacons in a highly directed beam rotating with a constant angular speed. The received beacons are stored and time stamped by every node. The calculation finally is carried out on a central system based on triangulation of the angular values retrieved by the beacon time-delays. This approach may seem promising, but by taking a closer look it can be observed that in [22] all nodes have to be time synchronized, which is not applicable on the existing RFID framework.

4 Model of Positioning System for active RFID

The following pages give an overview about the design considerations for establishing an active RFID positioning system. The ideas are based on the existing system of Free2Move. The proposed system is dimensioned for indoor use in a warehouse-like non line of sight environment and should work with the existing Free2Move hardware. As field of operation an area of approximately 50x50 meters and a transmitting range of the directional units (DU) and the readers of about 25 meters are assumed. Due to the lack of computation power within the tags a centralized system which computes Angle of Arrival (AoA) raw-data is proposed. The goal of the project is to achieve the ability to determine the actual position of a tag with the help of a reader and a backend system. Other assumptions include quasi-static operation, a relative coordinate system and low cost components.

4.1 System Components

In the following subchapters the necessary components of the proposed system are described. The components of the system contain the hardware of Free2Move, additional directional units and a backend system. The existing system of Free2Move is an active RFID system consisting of one or more reader units and simple transponders (tags). The tags are equipped with limited memory capacity and computing power. Data exchange is done in the 2.45 GHz ISM-band via a self developed communication protocol.

4.1.1 Directional Unit (DU)

A directional unit (DU) is a simple device which provides angular data. It consists of a motor, which powers a rotatable platform. On this platform an antenna is mounted. The design of an antenna is a time consuming and expensive task which lies beyond the scope of the project. It is desirable, that the antenna possesses a narrow beam width (α_{beam}) and high directivity with minimum side/back lobes. The rotation takes place with a constant angular velocity (ω). The range in which the antenna is moving can be adjusted to the environmental conditions. The antenna constantly sends out its angular position (beacon) referred to a common reference (pole). Therefore, each DU must be calibrated in an initial setup phase. The following parameters should be configured during this phase:

- Start and end angle of the transmission range
- Pole (initial position for 0 degree)
- Angular velocity (ω)

The specification of the existent RFID-system and suggested hardware from Free2Move lead to the following guidelines for the design of the first prototype:

- Transmission range of approx. 25 meters
- minimal resolution of 1 degree
- minimal 15ms for each step (accords to 1 degree)
- minimal period between sending data packets is set to 1.72 ms
- This results into minimum 9 data packets per step ($9 \cdot 1.72ms = 15.48ms$). The maximum angular velocity is then given as $\omega = 64.6°/s$

4.1.2 Tag

An adapted tag from Free2Move is used as RFID-receiver. The tag operates in the ISM-band and is able to communicate with readers. There is 1 kbyte of memory available for positioning purposes, which limits the size of the data history.

The tag is able to tune into different frequencies within the ISM-band. At the moment there exist 80 different channels which are separated by 1 MHz. The minimum time for switching the communication channel is approx. 3 ms. In order to provide flexible operation it should be possible to configure the tag dynamically. The parameters explained in Table 4.1, and illustrated in Figure 4.1 are adjustable and must be set within a start-up phase. Additionally a DU-list (or update list) has to be transmitted every time a tag enters a new environment.

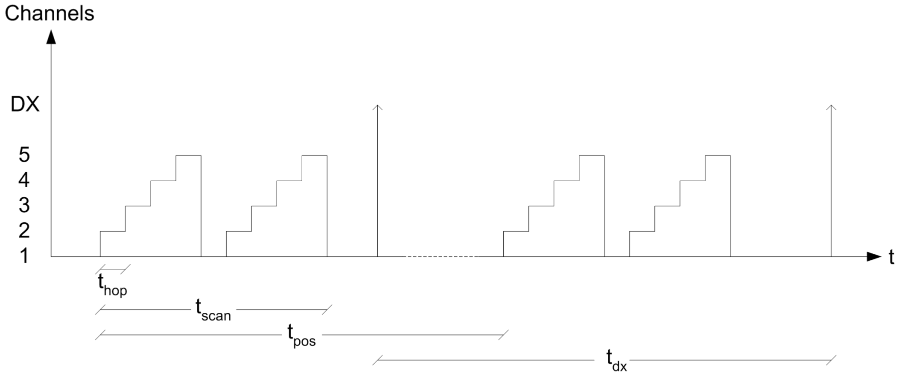


Figure 4.1: Illustration of configurable time intervals on a tag

The tag should store a tuple illustrated in Figure 4.2, which is referred to as position rawdata. The rawdata is extracted from the beacon signal of a DU. One tag is able

Hopping interval	t_{hop}	time for listening on one channel
Overall scanning interval	t_{scan}	Total time for listening on the channels
Position interval	t_{pos}	time between successive scans
Data exchange interval	t_{dx}	time between successive data exchanges with a reader

Table 4.1: Configurable time intervals on a tag

to store up to 1 kybte respectively 128 tuples overall. If the memory is full, data is overwritten in FIFO manner. Instead of the DU-ID (40 bit) only the frequency (8 bit) is taken as identifier to save the very limited memory capabilities. Furthermore a tuple consists of the AoA (8 bit) and a timestamp information (27 bit). The timestamp is a relative timer in order of seconds and should be sufficient for a three year use.

Frequency	AoA	Timestamp
8 Bit	8 Bit	27 Bit

Figure 4.2: Packet structure of a tuple

4.1.3 Reader

The reader from Free2Move will remain unchanged in the main parts. It is responsible for configuring the tags and receiving position raw data. Therefore its software has to be adapted to the communication protocol.

4.1.4 Backend system (PC)

The backend system is hosting a database and performs the positioning computation. Therefore it needs to be connected to all used reader either by Ethernet or Bluetooth.

4.2 Communication Protocol

The communication protocol should provide an efficient data exchange between DU, reader and tag. Efficient means a time and power saving process, which provides the backend system with necessary positioning information. The protocol is based on the considerations in the following subchapters.

4.2.1 Packet Design

For the communication between DU and the tag a minimum packet structure is proposed (see Figure 4.3), consisting of the DU-ID and the current transmission angle:

DU-ID	AoA
40 Bit	8 Bit

Figure 4.3: Packet structure on the air interface

In order to be able to tune into the right frequencies, the tags need to know the used frequencies of the relevant DUs. This information is provided by one or more reader by transmitting the following packet to the tags, which is also referred to as DU list, illustrated in Figure 4.4:

Begin	Frequency 1	...	Frequency n	End
4 Bit	8 Bit		8 Bit	4 Bit

Figure 4.4: Basic structure of a DU list

The 8 bit information for the frequencies consists of the 8 least significant bits of the binary representation of the frequency with offset 2.400 GHz. Within the ISM Band this allows 80 different channels, 1 MHz bandwidth each.

4.2.2 Data Flow

The following paragraphs describe the data flow between reader, tag and DUs. The communication bandwidth of the air interface between these components is 250 kbit, which results in an airtime of about 1ms for one packet, where one packet consists of 260 bit. The proposed communication flow is visualised in Figure 4.5.

During the installation of the RFID system, a start-up phase is necessary to adjust each tag which should be positioning-enabled. As a first step, tags are configured by the reader with a basic setting, including tag-ID and a certain data channel. After an acknowledgment by the tag, all parameters necessary for positioning are set on the tag over the arranged data channel. These parameters include among others position interval t_{pos}, data exchange interval t_{dx}, overall scanning interval t_{scan} and the DU-list, as described above in chapter 4.1.2. If the data is received error-free, the startup phase is finished with another acknowledgment by the tag.

After the start-up phase, which is usually carried out once as initial configuration, the tag changes into the *sleep mode*. When the set position interval (t_{pos}) is over, it ”wakes up” to listen on the different channels defined within the DU-list. This listening is done for the duration of t_{scan}, to change back into ”sleep mode” afterwards. At this point, all received tuples from different DUs are stored within the memory of the tag. Next, when the timer

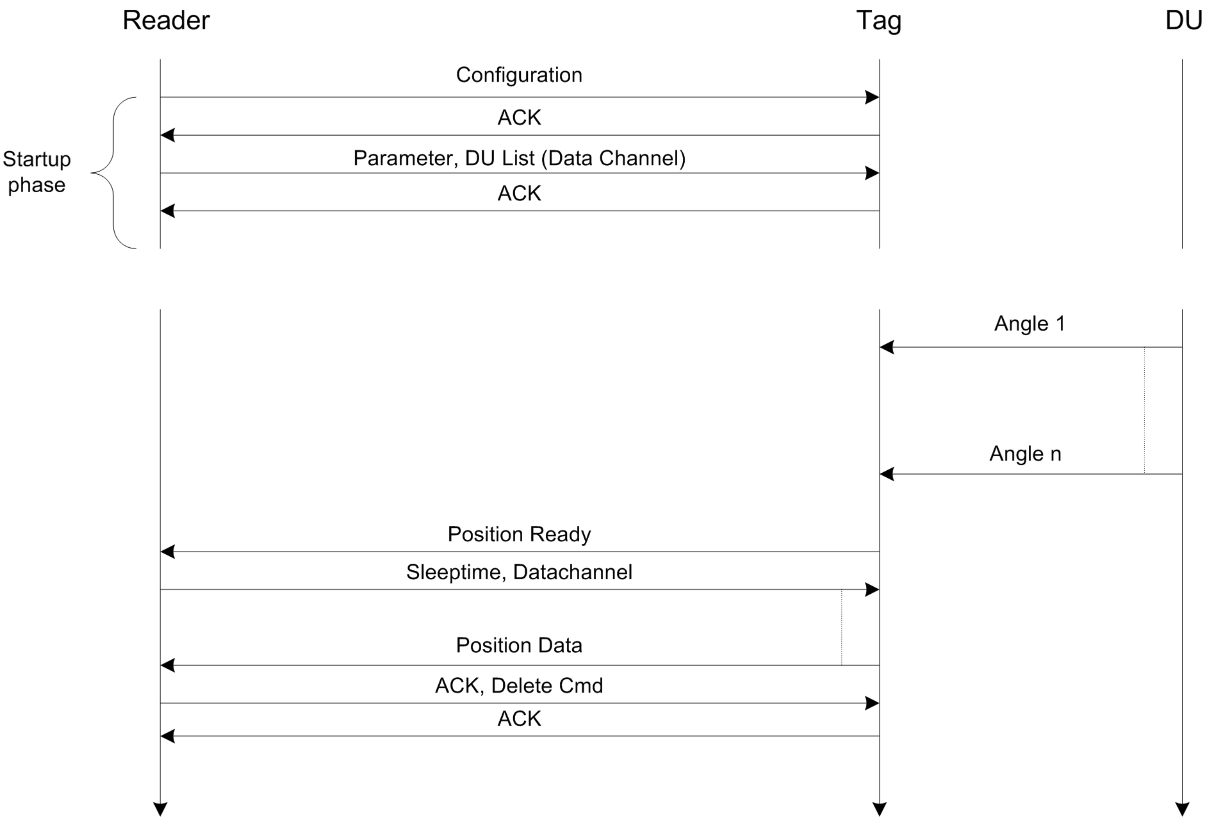


Figure 4.5: Flow digram of the communication with a reader

reaches the data exchange interval (t_{dx}), the tag gets active again to report to a reader, that position data is available. This communication is done on the default channel. If a reader is within range, it responses on the same channel by assigning a sleeptime and a data channel to the tag. When the sleeptime is over, the tag sends all its position data via the assigned data channel to the reader, where it can be stored for further computation. After acknowledgments by both parties, the tag changes into *sleep mode* again, to start the listening again after the set position interval (t_{pos}).

4.2.3 Multiple Access

For accessing different channels a pure FDMA approach with every DU transmitting on an own frequency was suggested. The drawback of this model can be clearly seen if the above shown process is considered. In order to receive every possible DU, the tag has to wait $n \cdot t_{max}$, whereby n is the total number of DUs in the area and t_{max} the maximum rotation time for one DU. For possible use cases this would result in a total listening time in the order of minutes. The second discussed approach is an enhancement to the first one. Therefor TDMA would be used to shorten the listening time by dividing the broadcast time for a certain angle through the number of available DUs. The problem of this approach lies in the need for tight synchronization of DUs, which is not possible because the DUs are not interconnected with each other.

As a further strategy fast frequency hopping was proposed as enhancement to the original approach. The hopping through the channels faster than a DU transmits values for one angular unit results in a complete scan over all frequencies. With this approach it is possible to receive enough packets of every DU within only one t_{max}, as illustrated in Figure 4.6. The drawback of this strategy is the loss of information, while hopping through the different frequencies. Although hopping introduces a minor systematic error, the advantage of a short overall listening time prevails.

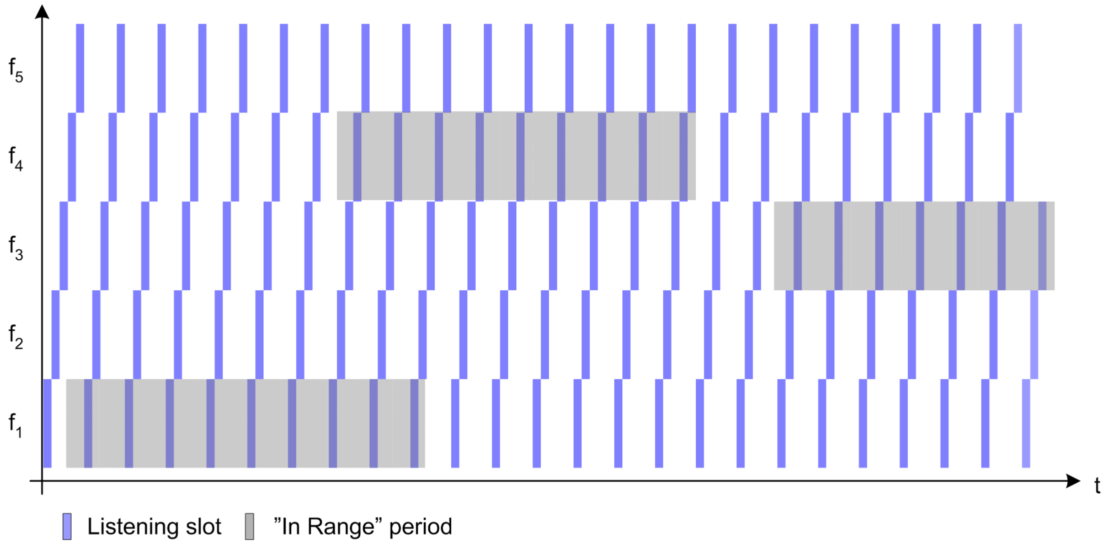


Figure 4.6: Hopping pattern

Two different strategies for reacting on the reception of a beacon were taken into account. In order to receive all possible packets on one channel respectively DU it would be necessary to remain on the channel when one beacon was received. After a timeout the hopping starts again. The drawback of this approach would be that the tag is occupied for the whole receiving period and can not receive signals from other DUs. The situation is exacerbated if a DU has a lower angular range and therefore can be "seen" more often by a tag. This also applies to multipath propagated signals. Hence, this approach was not considered any further.

The second strategy is to continue unchanged after a beacon is successfully received at one channel. The hopping is done until the scanning interval t_{scan} times out (adjustable parameter during start-up phase). Still there are different suggestions how the hopping pattern can look like:

- The most obvious hopping pattern follows the frequencies incrementally from 1 to n, whereas n means the number of frequencies in the DU list. Then it restarts at 1 again.

- Another possibility is to hop between channels in a randomized fashion. This would imply the need for a random generator or a pseudo random static list on the tag.

- Also a slight variation of the first pattern was considered, where the scanning process follows the channels incrementally for n times, and then restarts at variable start points.

4.2.4 Non-static Operation

The above described methods are based on a static field of operation. To provide mobility, a more rapid communication process has to be developed, which may be not possible within the existing framework.

4.3 Positioning Algorithm

The proposed positioning algorithm consists of several computational steps illustrated in Figure 4.7. All these processing steps are carried out on a backend system which is connected to the reader, either by Ethernet or by Bluetooth and is therefore able to receive the position raw data of a tag.

First, the received rawdata are ordered by DU-ID as primary sorting criteria, secondary by timestamp and third by the received angle. The result is a list of all available data per DU additionally sorted by time and angle. Due to multipath, more than one angular group per DU can be received. Therefore the sorted rawdata is grouped by taking into account the amount of signals per DU respectively per group. Out of the value group, a best value is estimated, which means that a reasonable angle is assigned to the group. It is possible that for one DU several dominant value groups exist, which all are used for further calculation, because at this point no assertion can be made which value group is the correct one.

During the triangulation, intersection points between all the groups are calculated based on methods of trigonometry. This is possible, because for every group a vector can be spanned from the DU in the angular direction. The resulting coordinates are then analysed and checked on their plausibility based on the following criteria:

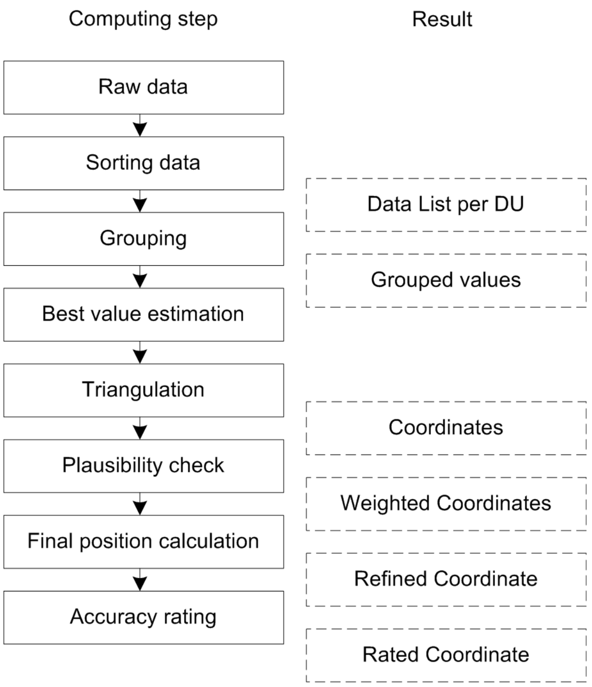


Figure 4.7: Processing chain of the positioning algorithm

Intersection area: The intersection points lie inside the intersection area of the two DUs from which the point is calculated. Otherwise one of the vectors derives definitely from a reflection and the intersection point can be discarded.

Negative direction vectors: In the present case it is not possible that the intersection of two vectors is calculated with a negative direction vector. If a negative direction vector is computed, one angular information of the performed calculation can not be correct.

Border check: Due to the known dimension of the operation area, intersection points which lie outside this area are discarded.

The remaining intersection points are then weighted based on the following known criteria:

Angular border: For DUs with certain rotation angles, the start and end area of the rotation of the DU are prone to errors, because not all angles for one group can be received. In this regions a lower weighting is applied.

Number of values per group: The amount of received angles is an indicator for the quality of an intersection point. It is assumable that tuples from multipath propagation are less constant in their occurrence. Hence correct groups can be recognised by their higher number of values per group.

Obtuse intersection angle: Intersection angles between two vectors are only limited suitable for precise triangulation. Similar to fundamental landscape surveying a sufficient basis is required between two measuring points for accurate results. This means that intersection angle between 40°-140° are more reliable than others. This fact is taken into account by applying a corresponding weighting.

Number of intersection points within a certain area: It is more reasonable that the real position is within the area of more intersection points. A weighting based on the amount of values is applied.

Out of the weighted coordinates, the final position is calculated which results in refined coordinates. Finally, a value is assigned to the calculated position which indicates an estimated accuracy and accordingly its reasonability.

5 Simulation of the proposed model

Based on the design guidelines and the above described model, a detailed simulation was conducted. The purpose of the simulation was on one hand to give a better insight of certain system processes and on the other hand to be able to make a profound statement about the performance and accuracy of the system. The physical transmission process was simulated in Matlab Simulink. The learnings were then built into a more flexible simulation model for Matlab Script, which evaluates the performance of the positioning system. Also the configuration of different system parameters was optimized throughout the simulation. Another focus laid on the testing and implementation of the positioning algorithm in Matlab. The simulation model can be adapted to different scenarios and give valuable information about the behaviour of the system.

5.1 Simulation of the Wireless Network via Simulink

In a first attempt the communication part was simulated with Matlab Simulink. In Simulink it is easy to create time-based simulations of communication systems. The libraries of Simulink offer a variety of different pre-designed blocks, also for more complex devices. The simulation should cover the listening process between the tag and DUs and should be as realistic as possible. Therefore the system was simulated on bit level, which resulted in a sampling frequency of 1 MHz. The general model (according to Section 4.2) was implemented with multiple DUs which transmit their data on one common channel but on different frequencies. The tag listens to this channel according to a hopping pattern. The setup of the basic model with three different DUs is shown in Figure 5.1. The model of a DU consists of a signal producing part (DU Signal) and according to the specifications a GFSK modulator. The DU beacon is a 260-bit stream which consists

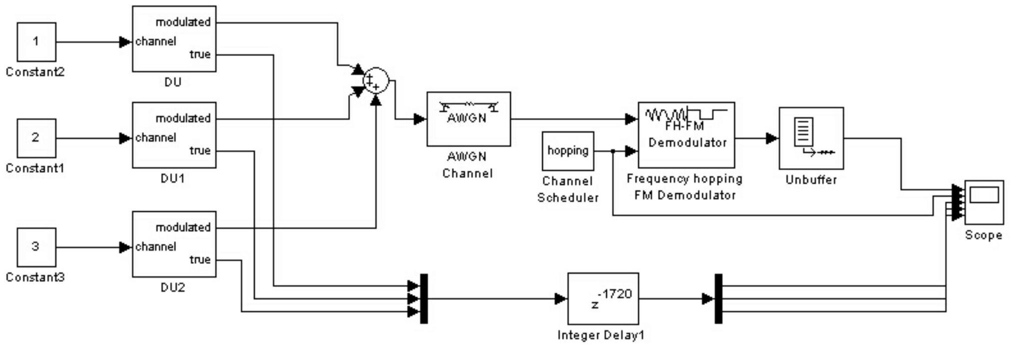


Figure 5.1: Simulink model with three DUs and one tag listening

of the DU-ID and the actual angle. Additionally, the signal is extended with 113 zeros which represent $440\mu s$ configuration time. The signal is only transmitted once during one rotation of a 90°-DU, which accords to approx. $1.35s$, for the time of sending 15° (α_{beam}). The signal is then modulated with a GFSK modulator and mixed from the baseband onto a carrier frequency. All carrier frequencies are separated by 1 MHz. The signals from each DU are then summed up and fed through an block, representing additive white gaussian noise (AWGN).

The tag is represented by a GFSK demodulator, which hops between different carrier frequencies. Before the signal is demodulated the signal itself is mixed from the carrier down to the baseband and then filtered with a low-pass filter. Figure 5.2 shows a short period of the simulated transmission. In this simulation three DUs are used, whereby DU 2 and DU 3 are transmitting ($t_{packet} = 1.72\,ms$). As it can be seen, the tag changes the frequency every 3 ms (t_{hop}) and is only able to receive packets on channel 2 and 3.

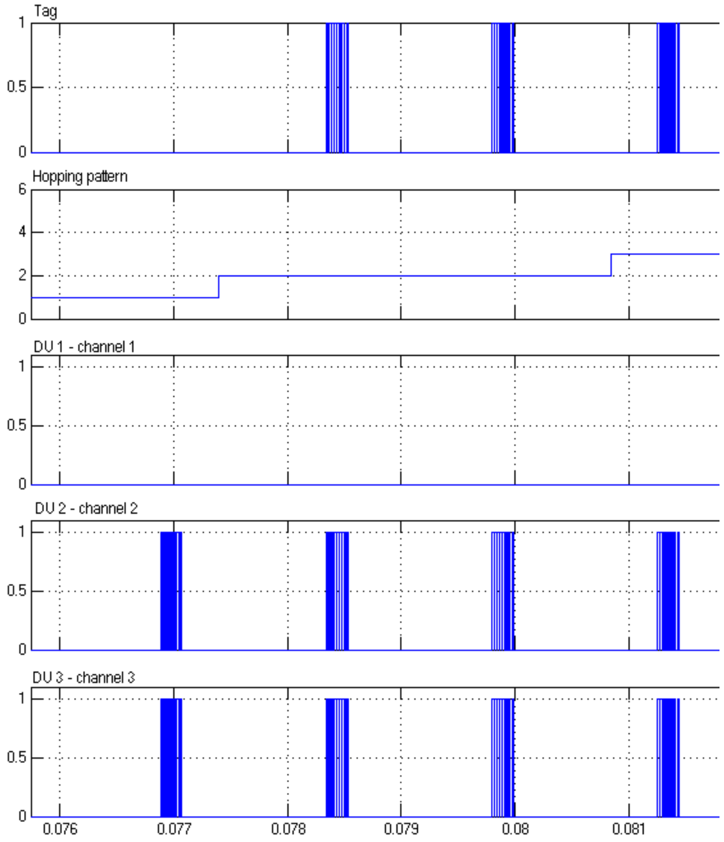


Figure 5.2: Simulink scope of the location sensing procedure on a tag with three DUs

The above described Simulink model of the positioning system gave a very good understanding of the physical processes on the air interface. The required high resolution of the system led to an unusable simulation time on the provided computer systems (e.g. the simulation of 1s with three DUs took almost 4 hours). Due to this fact, simulating different system parameters for the communication channel and additionally developing and testing a positioning algorithm based on this model could not be accomplished without extensive proprietary development of own Simulink blocks. Therefore this approach was not considered for further development.

5.2 Simulation Setup - Standard Set

Most of the following simulations were based on a simple configuration. The room is 50x50 meters large and every spot is covered with at least two DUs in order to enable positioning. For a reasonable result the DUs were placed in such a way, that in approx. 93 % of the area a third DU could be received. The range of each DU was set to 25 meters. Each corner was equipped with a DU. Additionally a DU was placed in the room centre $(25, 25)$ and at the centre of the border lines. This setup with 9 DUs results in a regular pattern. In order to be able to test different situations, a 10th DU was placed at the intersection point of two other DUs at the point $(12.5, 3.96)$. This DU should especially improve the result in the mid-left area. Figure 5.3 shows the detailed arrangement of the Standard Set. Numbers (1-10) indicate the position of a DU, letters (a-e) indicate reference positions for later simulations. The reference positions were chosen after several different considerations.

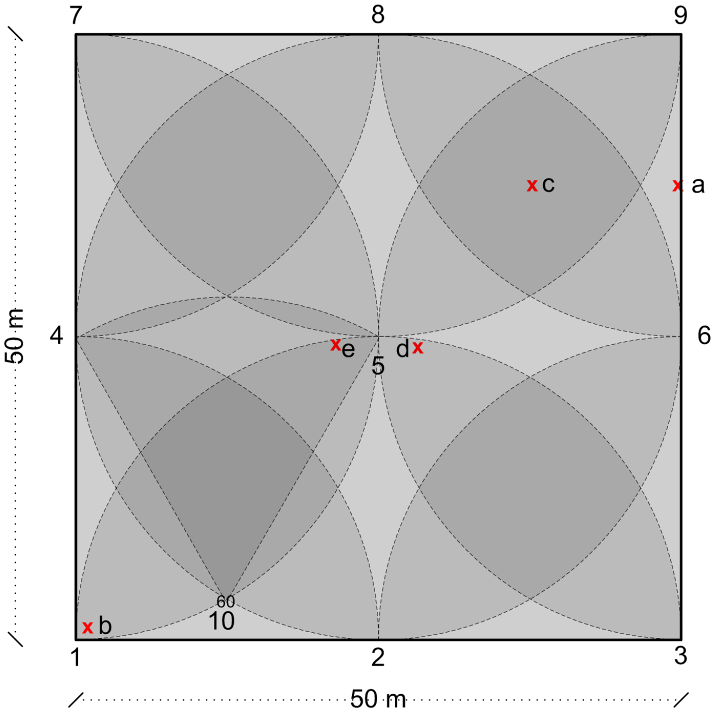


Figure 5.3: Arrangement Standard Set

Position (a) lies at $(50, 37.5)$ and represents a position at the border, where only two different DUs (6,8) are regularly received.

Position (b) at $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$ describes a standard position with 3 DUs (1,2,4) 1 meter away from the origin.

Position (c) $(37.5, 37.5)$ is a place where 4 DUs (5,6,8,9) can be received under constant conditions.

Positions (d,e) are close to each other and can receive at least 5 DUs (2,4,5,6,8) because they are placed close to the centre. The only difference is that position (e) is additionally in range of DU 10. The coordinates are (18.52, 24.14) for (d) respectively (31.48, 24.14) for (e).

5.3 Simulation Model

After the very detailed simulation of the communication model in Matlab Simulink a better performing simulation method was found by implementing the model in Matlab Script. The following variables are introduced for changing the parameters of the environment:

- relative Position of tag and DUs
- angular range of each DU, given by start- and end-angle of the range in degree
- output power of a DU P_t in dBm
- area of operation as dimensions of a rectangle in m
- a matrix representing the probability of occurrence of reflection (p_{ref}) and scattering phenomena ($p_{scatter}$)
- angular resolution of a DU res in units per degree
- number of signal repetitions per unit of a DU nr_{rep}
- beam width α_{beam} of the antenna on a DU in degree
- time t_{packet} for sending one packet on a DU in ms
- time t_{hop} for listening on a channel during the hopping on a tag
- value mem setting the available memory size on a tag in number of tuples

- number of passes *cycle* for one simulation, where one pass is given by $max_{rot} \cdot nr_{rep} \cdot res$ generated values per DU, where $\max_{rot}$ is the maximum rotation angle of all DUs.

The simulation program was divided into several procedures for optimal representation of the real environment. After determination of the angles and distances between a certain position of the tag and the DUs, the idealised values received from every DU are computed based on the given parameters. On the resulting data matrix a function representing attenuation is applied. Attenuation is introduced by means of a calculated bit error rate (BER) depending on sending frequency, distance and output power of the DU. As further error the multipath phenomena of scattering and reflection are added according to the set parameters. Also, a hopping pattern is applied on the matrix to result in a set of data representing the received tuples at the tag. A Matlab implementation of the positioning algorithm processes the received tuples (rawdata) and calculates the position.

The following chapters describe the purpose of the different functions and give some details about the considerations which lie behind the simulation scripts.

5.3.1 Idealised Data

The first procedure of the simulation program results in a cell array consisting of one matrix for each DU, representing the tuples sent simultaneously via the air interface in a specific time period. For obtaining this cell array, some initial steps are carried out. Firstly, the distances *dist* and the true angles between each DU and the tag, γ_{real}, are calculated and stored. The positive y-axis is taken as pole for the angles, increasing clockwise. Secondly, the rotation in degree for each DU rot_{DU} is calculated based on the given values. The maximum occurring rotation is then stored separately in the value max_{rot}. As final step of the initial preparation, the DUs are divided into two groups: those, for which the tag's position is within their angular range (*in-range*) and those whose rotations does not reach the tag (*out-range*). This separation is practical for the further computing to increase the performance of the simulation script. Additionally, DUs with exactly the same position as the tag are also added to the *out-range* list, because no meaningful data can be gained. It is necessary to consider this condition for simulation purposes, whereat it is impossible to occur in a real setup.

After these preparatory steps, the main loop is filling up a matrix for each DU and storing it in a cell array. Each line of the matrix represents one packet which consists of one tuple and additionally the real expected angle between DU and tag for further computation. For the simulation, the number of lines in every array, respectively the number of packets sent during one scanning interval of the tag, is defined by $nr_{rep} \cdot res \cdot max_{rot} \cdot cycle$. Because all DUs send their data simultaneously, the overall scanning interval (t_{scan}) for a tag is given by the number of packets at one DU multiplied with the time for sending one packet, t_{packet}. This results in the following equation:

$$t_{scan} = nr_{rep} \cdot res \cdot max_{rot} \cdot cycle \cdot t_{packet} \tag{5.1}$$

As defined in Section 4.1.2, each tuple stores three values: the frequency of the DU, a timestamp and the actual AoA (γ) of the DU.

For the simulation, a DU-ID was taken instead of the frequency to identify data from different DUs. DU-IDs are set as integer numbers increasing from 1. Another purpose of the DU-ID field is to set whether the actual packet can be received by the tag or not. If a packet cannot be received, the DU-ID is set to zero instead of the certain DU-ID. This is the case for packets which are in the *out-range* group. The DU-ID field will also be used for indicating that a packet was affected by one of the propagation errors, which are mixed into the idealised data as further step.

A relative timestamp for each tuple is calculated by simply adding t_{packet} to every prior timestamp of the actual DU, starting with an initial value. The initial value, a random number between zero and t_{packet} in order of $10^{-5}s$, is computed separately for every DU and has the purpose to simulate asynchronism between the different senders.

The third field is used for the actual AoA of the DU, where angles are given in radiant. Angles, for which the beam of the antenna does not reach the position of the tag, are not calculated but set to zero in order to increase simulation performance. This means that in this phase just angles within the limits from $\gamma_{real} \pm (\frac{\alpha_{beam}}{2})$ will occur within the matrix.

As a result of the function, a cell array is generated representing the simultaneous sending of packets by all DUs during the scanning interval. Every packet has a valid timestamp, whereas only packets which can actually reach the tag have also set DU-ID and angle of arrival. Because the DUs are rotating in asynchronous fashion, the timestamp of the packet, where the antenna beam reaches the tag, is set randomly within the actual rotation span rot_{DU} of the specific DU. Due to the fact that the whole function is intended to produce idealised data, exactly the amount of $nr_{rep} \cdot res \cdot \alpha_{beam}$ packets can be received by the tag and for those DU-ID and AoA are set.

As stated before, t_{scan} is calculated based on the maximum pivoting range of all participating DUs max_{rot}. Therefore, the rotation of the antenna within its range rot_{DU} is repeated until the value of the timestamp reaches t_{scan}. Considering this, the maximum amount of received data max_{data} from a certain DU is expressed in the following equation:

$$max_{data} = nr_{rep} \cdot res \cdot \alpha_{beam} \cdot \frac{max_{rot}}{rot_{DU}} \tag{5.2}$$

The generated data represents the idealised number and order of the tuples sent from every DU to the tag. So far, no restricting factors like the beam range were considered. This limiting factors or propagation errors are mixed into the idealised data array within the next procedure, which is discussed in detail in the upcoming chapter.

5.3.2 Propagation Errors

Wireless communication always suffers from several impairments, which affect data communication. The simulation model includes the following interferences, which all directly influence the idealised data set:

- Path loss in indoor environment
- Noise
- Scattering
- Reflections

Path loss and noise are the two dominating factors for a successful operation. Furthermore scattering and reflections bring up additional difficulties, because they interfere with the idealised channel and add unwanted information. The way how those impairments are treated within the simulation is described in detail in following section.

The energy of electro-magnetic waves decreases as the distance between transmitter and receiver grows. This effect is known as path loss. The received power P_r can be expressed as:

$$P_r = A \cdot d^{-\alpha} \tag{5.3}$$

where A is a constant defined by the transmitting power, the used frequency f and antenna specifications. α denotes the exponent of decrease. In logarithmic form and inserted for A, the path loss can be expressed as:

$$PL_{dB} = -32.45 - 20log(f) - \alpha \cdot 10log(d) \tag{5.4}$$

For communication in free space (e.g. satellite communication) α is set to 2. Indoor environments are harsher and therefore the signal will disperse much faster. Following

[23] the value for α is set to 2.7 in our simulation model. To account additionally for higher absorption a floor loss value FL [24] is introduced and set to $-10dB$. The received energy at the tag is therefore calculated in units of dB, where P_t is the EIRP of the transmitting antenna and the antenna gain of the receiver is set to $G_r = 1$:

$$P_r = P_t + PL + FL \tag{5.5}$$

In order to be able to calculate the prevailing bit error rate (BER) out of the Signal-To-Noise Ratio (SNR), noise is simulated by calculating thermal noise and adding summarized additional noise (like co-channel interference, crosstalk and the fact, that the ISM-band is used by other applications like 802.11b or Bluetooth) following the chapter *Line-of-sight transmission* of [25, p. 110-118]. Thermal noise in watts (Johnson noise) is given by:

$$N = kTB \tag{5.6}$$

where k is Boltzmann's constant, T the temperature and B the bandwidth of the signal. For normal room temperature $T = 293\,K$ and a signal bandwidth $B = 350\,MHz$ the thermal noise is expected to be $N = -118.5\,dBm$.

The value of the additional noise is determined empirically. In order to account for temporal and spatial changes in the receiving conditions the value is normal distributed with mean $\mu = 15dB$ and variance $\sigma^2 = 2dB$. Adding thermal noise and additional noise leads to a mean noise floor value of approximately $-103.5\,dBm$. With these values it is possible to calculate the $\frac{E_b}{N_0}$ value with a bandwidth of $B = 350\,MHz$ and a bitrate of $R = 250kbit/s$.

$$\frac{E_b}{N_0} = \frac{S}{N} \cdot \frac{B}{R} \tag{5.7}$$

Based on [26], [27] and the product specification [28] of the used transmitter the BER was modelled with a similar function. Equation 5.8 led to a satisfactory result and is shown in Figure 5.4.

$$BER = \frac{1}{2} \cdot erfc(\sqrt{\frac{E_b}{N_0}}) \cdot 10^{\sqrt{1.21 \cdot \frac{E_b}{N_0}}} \tag{5.8}$$

The BER is the dominating factor for the whole communication process, because it primarily affects the transmitting range of the whole system (see Section 5.4.2). Therefore the BER is calculated for every transmission from a DU to a tag. With this value an error vector is created and then OR-added to the idealised data, which was generated by the functions described in Section 5.3.1. In this context the term nominal range of a DU is defined. The nominal range corresponds to distance from a DU, where the BER is approximately equal to $4 \cdot 10^{-4}$. This means, that the nominal range of a DU can be

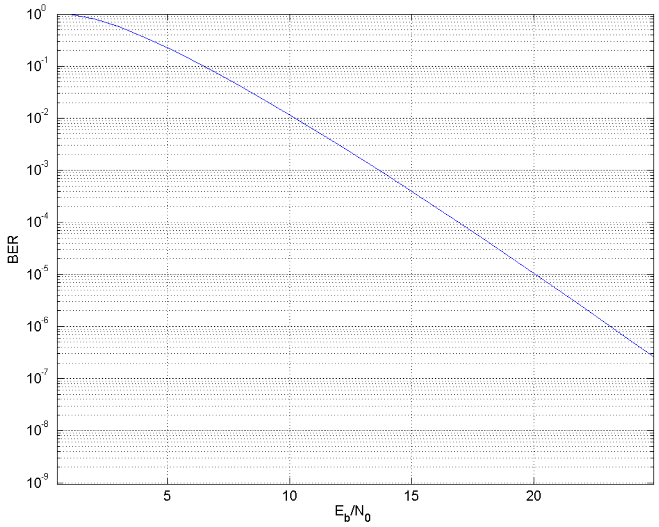


Figure 5.4: Simulated BER model for a GFSK Demodulator

adapted by configuring the output power P_t.

Multipath propagation can cause severe deterioration especially in indoor environments and must therefore not be neglected. Two different multipath effects (scattering and reflections) were modelled into the simulation. Firstly scattering from different objects in a room is mixed into the idealised data. Scattering occurs if the signal hits an obstacle, which is in the size of the wavelength. In the case of the ISM-band all objects in the size of 12.5 cm lead to a scattered wave. Scattered packets can be sent maximally $nr_{rep}/2$ times in a sequence in the simulation and are added to the idealised data with the preset probability $p_{scatter}$ for each DU. Reflections are simulated in the same way and can be controlled by the parameter p_{ref}. The number of repetitions of reflected data is normal distributed with $\mu = l_{ref}$ and $\sigma^2 = 0.3$, where l_{ref} is an adjustable parameter. This makes it possible to adjust the frequency and the duration of reflections for each DU. DUs which radiate directly on permanent obstacles are assigned higher reflection coefficients than DUs which can radiate into a free area.

Multipath propagation is characterised by a longer propagation path and therefore leads to a delayed arrival. The time shift due to multipath propagation for a nominal range of 25 meters is maximal $1.3\mu s$ and therefore much smaller than the used time resolution. This does not result in changes of the timestamp information in the idealised data. Because multipath propagated signals travel a longer distance, the attenuation is higher and a separate BER has to be calculated. The covered distance is simulated Rayleigh distributed with $\mu = 1.2507$ and $\sigma^2 = 0.0172$. This means that no multiple reflections are considered in this multipath model.

If a multipath signal reaches the tag completely (this means not corrupted by a high BER) and no other information is present on the channel (the DU-ID is zero), the interference is considered constructively and a new tuple is added. On the other hand if the channel is occupied (non-zero DU-ID) the interference is considered destructively in any case and leads to a reception error, which sets the DU-ID to zero within the cell array.

After this step the matrices for each DU consist of the idealised data modified by the effects of normal and multipath propagation. All matrices represent therefore all information, which is present on the air interface.

5.3.3 Hopping

As final procedure of the simulation script, a hopping pattern is applied on all data available on the air interface at the tag's position. The data is represented by the modified cell array.

Due to the fact that every matrix within the cell array represents exactly one channel a hopping sequence can be applied. Valid packets within the channels are indicated by a non-zero DU-ID, which makes it possible to filter out useful packets.

A packet has to fulfil the condition to be sent entirely within the time span of one hopping time to be received as valid packet at the receiving tag. If the period of sending a packet coincides with the change of the channel, this condition is not fulfilled anymore and the received fraction would be discarded by the tags internal CRC. Figure 5.5 illustrates the hopping pattern for $t_{packet} = 1.72ms$ and $t_{hop} = 3ms$. The random starting times of the different channels due to not synchronised DUs are realised by introducing a randomly chosen initial time value as described in Section 5.3.1. In the hopping procedure within the Matlab script, a relative timer t_{act}, representing the actual time at the tag, is introduced. t_{act} counts milliseconds and starts at zero, like the timestamps added to the tuples at the DUs.

The timestamp of the first packet (TS_1) is then compared with t_{act} and $t_{act} + t_{hop}$. If TS_n is bigger than t_{act} but still smaller than $t_{act} + t_{hop}$, the timestamp of the next packet, TS_{n+1}, is also compared with the same value of $t_{act} + t_{hop}$. In the case that TS_{n+1} is smaller or equal to $t_{act} + t_{hop}$, the packet would fit completely into one hopping period and therefore is de facto received by the tag. This comparison is carried out for every following timestamp TS_n, until the condition of being smaller or equal to $t_{act} + t_{hop}$ is not fulfilled anymore. In this case the local timer t_{act} is increased by t_{hop} and the channel number is changed. The described comparison then continues until t_{act} has reached the value of t_{scan}. The change of the channel number can be done in linear or in random fashion, depending on the specific simulation scenario.

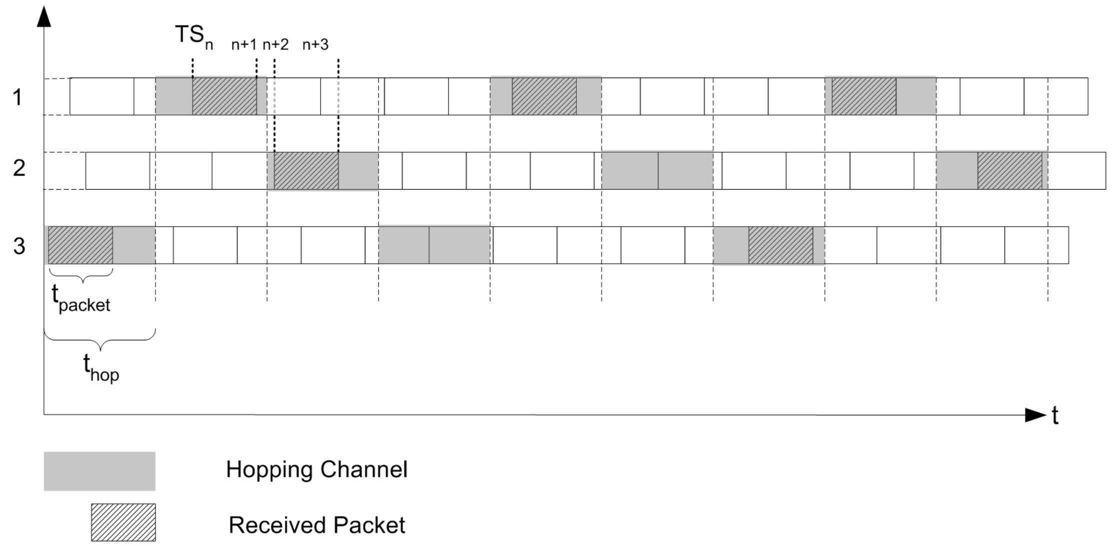
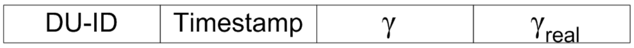


Figure 5.5: Hopping principle

All actually received packets by one tag are finally sorted out based on the validity (non-zero) of their DU-IDs. As final result, a four columned matrix is created with lines representing tuples (illustrated in Figure 5.6) sent by different DUs, ordered by their timestamps. The angle γ represents the received AoA.



DU-ID	Timestamp	γ	γ_{real}

Figure 5.6: Representation of a tuple in the simulation program

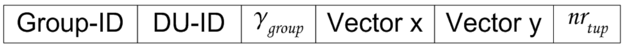
5.3.4 Implementation of the Positioning Algorithm

The positioning algorithm uses the list of tuples for calculating the most reasonable position based on the considerations in Section 4.3. To obtain an acceptable result, the algorithm performs the following basic steps: First of all, the rawdata are split into groups to reduce the amount of tuples to the most logical values. Out of these groups, intersection points are calculated, which are checked on their plausibility and weighted due to known facts. In the end, the weighted intersection points are used for calculating the final position.

5.3.4.1 Sort Data First step is the sorting of the data by DU-ID, timestamp and angle to provide the grouping function with an ordered list.

5.3.4.2 Grouping and Best Value Estimation In an idealised scenario without multipath and information loss, rawdata contains for each DU the maximum amount of angles in the range of $\gamma_{real} \pm \frac{\alpha_{beam}}{2}$. In this case, a grouping would not be necessary. It is enough to calculate $\frac{\gamma_{min}+\gamma_{max}}{2}$ to get the final angle and therefore the best value.

It is assumable that the rawdata contains diverse information concerning the angle to one DU originated from multipath propagation and broad beamwidth characteristics of the antenna. These additional tuples are not useful for the computation and should ideally be recognised and discarded. A reliable assertion cannot be done and therefore rawdata are classified into logical groups. The grouping does not result in one group with the correct angle, but it reduces the amount of total values and filters out several scattering signals. Groups that resulted from multipath signals are indented to be recognised during the plausibility check later on and are discarded there. The list of groups has the structure illustrated in Figure 5.7. The Group-ID is an incrementally increasing value to indicate the group. The DU-ID is taken from the rawdata in order to be able to assign a group to a DU. γ_{group} indicates the calculated mean angle for the group. Additionally, the vectors according to this angle are calculated. The x-value of the vector is $\frac{\cos(90° - \gamma_{group})}{|v|}$ and the y-value is $\frac{\sin(90° - \gamma_{group})}{|v|}$. The angle has to be subtracted by 90° because the y-axis is the pole of the DUs and not the x-axis, as used in ordinary trigonometry. The values are also divided by its absolute value to obtain the unit vector. The last field in the matrix is the number of signals summarised to the particular group.



Group-ID	DU-ID	γ_{group}	Vector x	Vector y	nr_{tup}

Figure 5.7: Structure of grouped tuples

In order to obtain relevant groups all possible combinations of groups are compared to the available values. For this the implementation first generates a template $m \times n$ matrix which represents all possible groups that one DU can comprise, where m is given by $(rot_{DU} - \alpha_{beam}) \cdot res - 2$ and n equals $\alpha_{beam} \cdot res$ (for a $rot_{DU} = 360°$ m is set to $360 \cdot res$). Table 5.1 illustrates the template for a resolution of 1/°, spread of 15° and a rotation angle of 360°.

Next, a matrix (*group matrix*) of the same dimension is generated and filled up with zeros. The algorithm continues by hopping through the template matrix. At every cell of the template matrix, the according value is compared to all angles of the rawdata of the particular DU. If the value in the template matrix matches the angle in rawdata, a counter increases the cell in the group matrix at the respective position of the template matrix.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
358	359	0	1	2	3	4	5	6	7	8	9	10	11	12
359	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Table 5.1: Template matrix

After finishing this iteration, the sum of every row of the group matrix as well as the absolute maximum (row_{max}) is calculated. Every row which has a sum that is above $row_{max} \cdot groupTol$ is taken as group, where $groupTol$ is a tolerance value. For $groupTol$ 85% turns out as reasonable value during the simulation. The final result of the grouping is consequently a list of all interesting groups for the further calculation with the amount of tuples.

When the groups are known, a best value has to be found that represents the whole group. Within one group this best value is calculated as mean value of γ_{max} and γ_{min}. Another possibility would be to calculate the mean value of all angles which leads to a less accurate result, because γ_{max} and γ_{min} represent the border of the group and cover therefore a bigger field. In certain cases, groups that are very close to each other are calculated. If a group has the same γ_{max} or γ_{min} value than a previously generated group, the group with less number of tuples is discarded. If two or more γ_{group} exist which have a distance less than 2° from each other, those are summarised to one angle by taking into account the number of tuples.

Figure 5.8 illustrates one result of the grouping algorithm in a 50x50 m^2 area and an actual position of (23, 10). The left hand side shows all the received raw data and the right side what remains after grouping the values. In this example all the multipath signals are discarded, resulting in an accuracy of 10 cm, which is above the average results. The beacon placement follows the Standard Set in Figure 5.3.

5.3.4.3 Triangulation Tiangulation uses concepts from trigonometry, explained in Section 2.2.2, to calculate the intersection points between the resulting vectors of the groups. Therefore the algorithm needs at least two signals from different DUs. Due to the fact that the vectors are already calculated during the grouping it is fairly easy to use this information for calculating the intersection point and additionally the intersection angle between the two vectors. This angle is important for analysing the plausibility of the intersection point, explained in the next section.

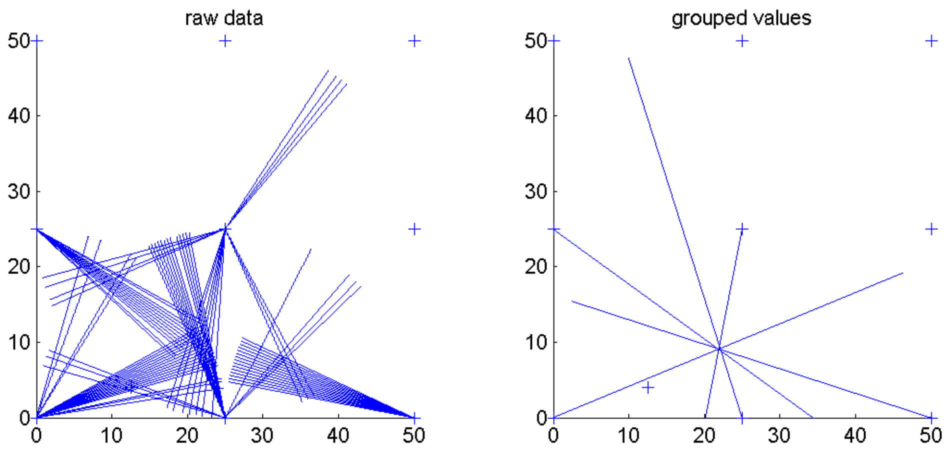


Figure 5.8: Grouping raw data

5.3.4.4 Plausibility Check The calculated intersection points are verified regarding their plausibility by several criteria. It is thereby possible to discard (weight=0) definitely wrong information or assign weights for remaining intersection points. Figure 5.9 shows the sequence diagram of how the checking is performed.

Intersection Area Test: This test checks if the intersection points lie inside the intersection areas of the DU's, from which signals are received. The weight is set to 0 if the points lie outside the intersection area, otherwise it is set to 1 by default. For angles close to the boundary of the antenna range an extra check is forced, because the accuracy of these values is less valuable due to the angular spread. For instance, in the case that a value group lies close to the border and it does not contain the maximum amount of values, it is not predictable which angles are missing. Tests have shown that setting the weight to 0 leads to a better accuracy but it has the disadvantage that values are discarded. That means, that some positions can not be calculated due to the information loss. For this reason, a function x^3 between 0 and 1 is introduced, that weights values near the border with nearly 0 and values farther away with weights in a rising manner.

Direction vector test: Negative vectors can be recognized by checking the dilation λ and μ of the vectors. If one of these values is negative, the intersection point is weighted with 0.

Number of Tuples Weighting: The amount of received tuples per value group (nr_{tup}) is an indicator for the quality of the intersection point, which results from the best value vectors. It is assumable that the groups resulting from reflections consist of

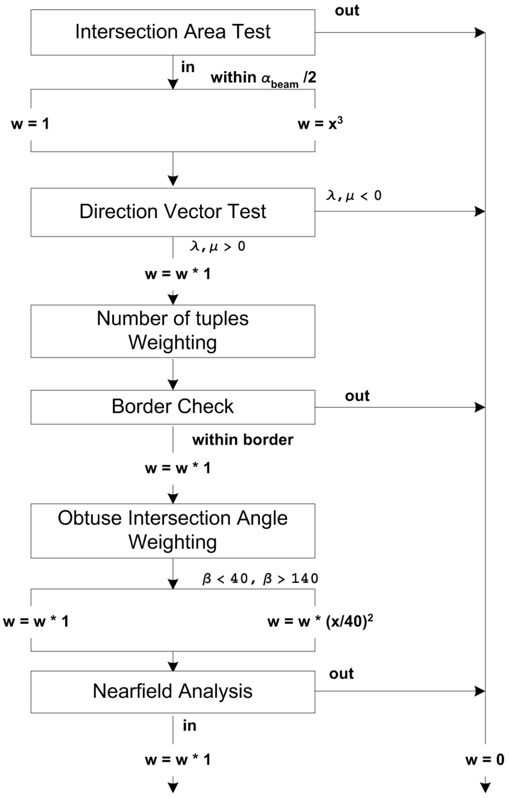


Figure 5.9: Plausibility check

less tuples. For this reason, intersection points with lower nr_{tup} are assigned lower weights, which reduces their influence on the later result.

For calculating the weight of an intersection point, first the number of tuples per group ($nr_{tup\,i}$) are normalised to the maximal occurring rotation angle max_{rot}. These normalised numbers are multiplied and divided by the square of the maximal possible number of values for a DU with max_{rot}, expressed by $\frac{max_{data}}{nr_{channels}} \cdot \lambda_{hop}$, where $nr_{channels}$ is the occurring number of DUs respectively channels, and λ_{hop} gives the rate of lost packets due to the hopping pattern, which will be discussed in Section 5.4.3. The assigned weight for an intersection point, which always lies between zero and one, is consequently calculated by the following equation:

$$w = \frac{nr_{tup_1} \cdot \frac{rot_{DU_1}}{max_{rot}} \cdot nr_{tup_2} \cdot \frac{rot_{DU_2}}{max_{rot}}}{(\frac{max_{data}}{nr_{channels}} \cdot \lambda_{hop})^2} \tag{5.9}$$

Border Check: Intersection points which lie outside the operation area are discarded. For positions near the border sometimes intersection points are calculated that lie outside the area but very close to the real position. For this case a tolerance factor of 0.5m is assigned, which was empirically determined.

Intersection angle weighting: The intersection angle β between two vectors gives a direct information about the quality of the intersection point. Figure 5.10 shows the behaviour of the position error (ΔPos) for a shift of ±1° for one vector compared to the intersection angle β. For this calculation two sides of a triangle and the angle between them were kept constant. The intersection angle β was then shifted ±1° and the error of the position (ΔPos) was determined. As it can be seen ΔPos reaches a minimum at 90° and stays relatively constant in both directions. The error grows to a major value in the border areas at 0° and 180°. Therefore the weight is applied in similar manner to counteract the introduced uncertainty of ΔPos. The doubled minimum error at 90° was chosen as threshold for a reasonable result, which corresponds to 40° respectively 140°. In both border areas the weight is assigned in quadratic manner according to the findings of Figure 5.10. Table 5.2 summarises the weighting due to an obtuse intersection angle.

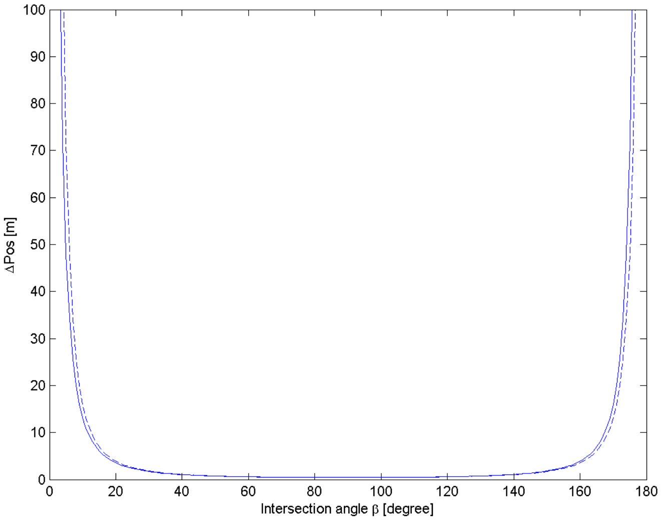


Figure 5.10: Error due to a shift of ±1°

$0^\circ \leq \beta \leq 40^\circ$	$w_{obtuse} = (\frac{\beta}{40})^2$
$40^\circ < \beta < 140^\circ$	$w_{obtuse} = 1$
$140^\circ \leq \beta \leq 180^\circ$	$w_{obtuse} = (\frac{180-\beta}{40})^2$

Table 5.2: Allocation of weights for obtuse intersection angle

Nearfield Analysis: By grouping the intersection points within an adjustable radius, a decision of the reliability of the grouped intersection points can be done. It is more reasonable that the real position is within the area of more intersection points.

The grouping is done by constructing a circle around every intersection point and counting the sum of weights. The intersection points within the biggest group gets the weight 1, whether the others are set to 0. Reasonable values for radius will be discussed in detail in Section 5.5.2.

5.3.4.5 Final Position Calculation The final position is a weighted mean value of all intersection points with the use of the weights assigned during the plausibility check. At least two DUs have to be in range for calculating one intersection point and taking this point as calculated position. This represents the worst case. More than two DUs are very important for the calculation of a more precise value. The following equation shows how the position value is calculated, whereby n is the number of intersection points within the nearfield with the highest sum of weights :

$$x = \frac{\sum x_n \cdot w_n}{\sum w_n},\ y = \frac{\sum y_n \cdot w_n}{\sum w_n} \tag{5.10}$$

5.4 Simulation of the Communication Protocol

After developing the above described simulation model for the positioning system, the next step was to test and optimize the proposed settings with different simulation setups and experiments. The aim of this simulation part is to analyse the overall performance of the communication protocol, described in Section 4.2 with respect to system variables (hopping time t_{hop}, number of repetition nr_{rep} , hopping pattern) and propagation phenomena. This subchapter will describe in detail the conducted experiments and will discuss the results.

5.4.1 Impact of Hopping Pattern

In Chapter 4.2.3, different hopping patterns were described, which were reviewed throughout this experiment. As indicator, the number of received values was taken. The simulation was carried out on the Standard Set, tag position (c), which means 4 DUs within range (nr_{rec}). t_{hop} was adjusted to 3.44, which gives a λ_{hop} of $\frac{1}{2}$ (for a detailed discussion about λ_{hop} see Section 5.4.3). μ_{rec}, the expected number of overall packets received by a tag during one *cycle*, is calculated by the following formula:

$$\mu_{DU} = \frac{max_{data} \cdot \lambda_{hop}}{nr_{channels}} \tag{5.11}$$

$$\mu_{rec} = \sum_{i=1}^{nr_{rec}} \mu_{DUi} \tag{5.12}$$

With α_{beam} chosen as 15 and nr_{rep} as 9, the expected number of received packets for a res of 1/° was calculated for each DU, giving a sum of 60.75, according to Equation 5.12. The following box plot illustrates the values of a series of $n = 2000$ simulations. The boxes have lines at the first, second and the third quartile. The whiskers show the extent of $1.5 \cdot IQR$ (Inter quartile range). Outliers are indicated as +. The results show, that the median for all three simulation runs is very close to the expected value. A big difference was found for the statistical spread of the different hopping patterns. Where the plots for both, the linear hopping pattern and the linear hopping pattern with random start points, show very narrow statistical spread, the random hopping pattern shows a four times wider spreading of values. This means the probability for having about one third more values is the same than it is for having about one third less. The same pattern was also explored for the number of values for each DU.

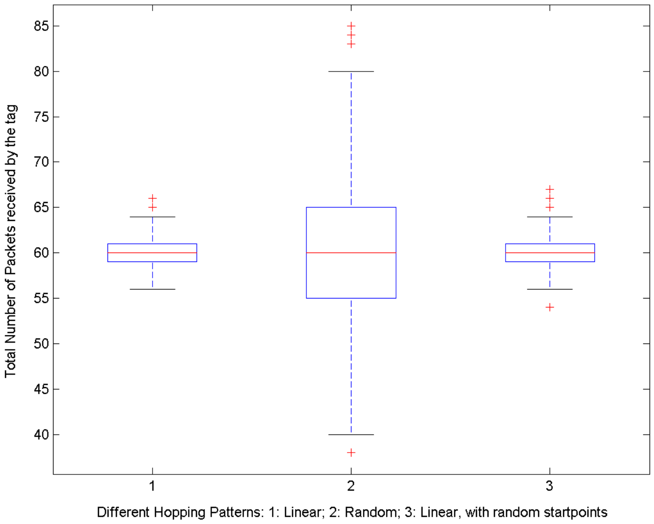


Figure 5.11: Comparison of different hopping patterns

Due to the fact, that the system should be as predictable and stable as possible, the behaviour of the random hopping pattern is not desirable. As best performing hopping pattern, the linear pattern was chosen. Although it generates almost the same statistical spread as the linear pattern with random start points, it is easier to realise and comprehend. Additionally, it has less outlier than the latter pattern.

For this reason, in all upcoming simulations, the linear hopping pattern was used, which is the recommended solution.

5.4.2 Impact of Attenuation

In order to be able to make a statement about the behaviour of a receiver, when it is moving away from a DU, the following experiment was conducted. A DU was setup ($t_{packet} = 1.72\,ms$, $\alpha_{beam} = 15°$) and the output power set to -2 dBm, for a nominal range of 25 meters. One receiving tag was configured with $t_{hop} = 17.2\,ms$ for just this DU, in order to be able to receive a high number of packets only from this DU (a detailed discussion about the effects of t_{hop} follows in Section 5.4.3). The tag was continuously moved away from the DU from the point $(0, 0)$ to $(50, 50)$, which accords to a distance of 70.7 meters. A measurement was conducted every 7.07 cm and repeated 100 times for one point. One measurement counts all the successfully received packets from the DU. The expected peak value (μ_{DU1}) for this scenario referring to Equation 5.11 is calculated as 121.5 packets per pass. The y-axis is normalised to this peak value. Figure 5.12 presents the result of this experiment. The mean values of each point are plotted against the distance between tag and DU. The results show clearly, that the reception quality is excellent for the first 23 meters and decreases then rapidly until it reaches zero at approx. 40 meters. Based on the data from the simulation it is possible to divide the area around a DU (with nominal range of 25 meters) into four different quality zones. Table 5.3 shows how the quality zones are defined in general.

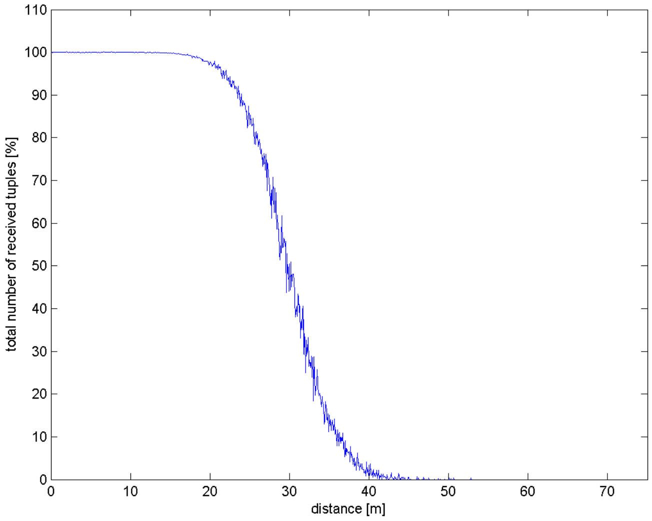


Figure 5.12: Effects of distance on quality of reception

For the simulated model this results in a range of Zone 1 of 23.5 meters and a good quality reaching up to 27.5 meters. A schematic drawing (Figure 5.13) shows the range of all zones for a DU with a nominal range of 25 meters. It can be seen, that 50 % of the packets

	Number of packets in %	Quality
Zone 1	$> 90\%$	excellent
Zone 2	$> 70\%$	good
Zone 3	$> 50\%$	critical
Zone 4	$> 25\%$	poor

Table 5.3: Definition of quality zones

are expected to be received successfully even 5 meters away from the nominal border. All tags beyond zone 3 will not receive data which are relevant for further processing.

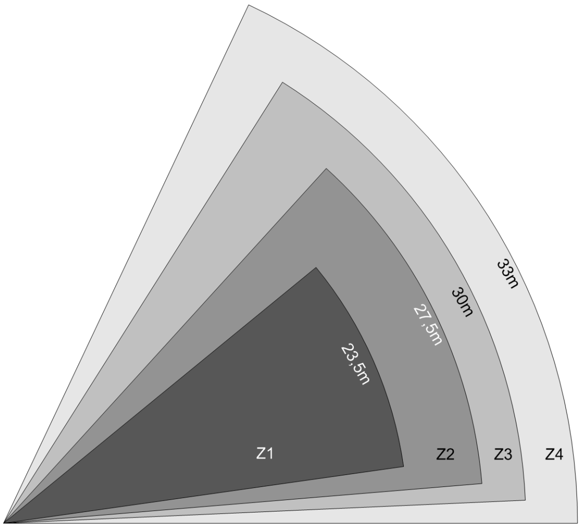


Figure 5.13: Quality zones of a 25m-DU

5.4.3 Impact of Hopping Time

This experiment should clarify the optimal setting for hopping time t_{hop}. Optimal in this context means, which hopping time leads to the highest number of received tuples, which can be passed to the positioning algorithm. It is obvious that with a higher hopping time, the tag is able to receive more values while listening on one channel. With a hopping time twice the packet time $t_{hop} = 2 \cdot t_{packet}$ at least one packet during the hopping time is received successfully. This means the rate of successfully received packets (λ_{hop}) is $\frac{1}{2}$. In general λ_{hop} can be expressed as:

$$\lambda_{hop} = 1 - \frac{t_{packet}}{t_{hop}}, \; t_{hop} \geq t_{packet} \tag{5.13}$$

With this value and the knowledge on how many different channels are present ($nr_{channels}$) it is possible to calculate the expected peak value of received tuples for one cycle (full 360° rotation) for one DU (μ_{DU}) from Equation 5.11. The term $\frac{rot_{DU}}{360°}$ is introduced for normative reasons, in order to be able to compare all kind of DUs.

The experiment was based on the Standard Set (see section 5.2), although all distances were halved. All measurements are taken at five predefined positions (a-e), which differ mainly in the number of receivable DUs (nr_{rec}). Each measurement was repeated 50 times in order to average out values and discard extreme. Figure 5.14 shows one result of the experiment. The sums of received packets from each DU for each position are compared for different hopping times. The results of the experiment are compared to the theoretical values of μ_{rec} (dashed line), which were calculated from the equations above. The x-axis is normalised to a multiple of t_{packet} and the y-axis represents the total number of values received by a tag. All DUs are normalised to a 360° radiator. This means, that DUs, which cover a smaller area are only included once. The figure shows clearly, that the theoretical calculations and the results of the experiment agree with each other. The theoretical maximum is exceeded in nearly every case, which can be explained by the fact, that also unexpected DUs can be received very weakly. It can be observed, that in every case the amount of received tuples increases while the hopping time increases. If the quality of the values is not considered, higher hopping times should be preferred.

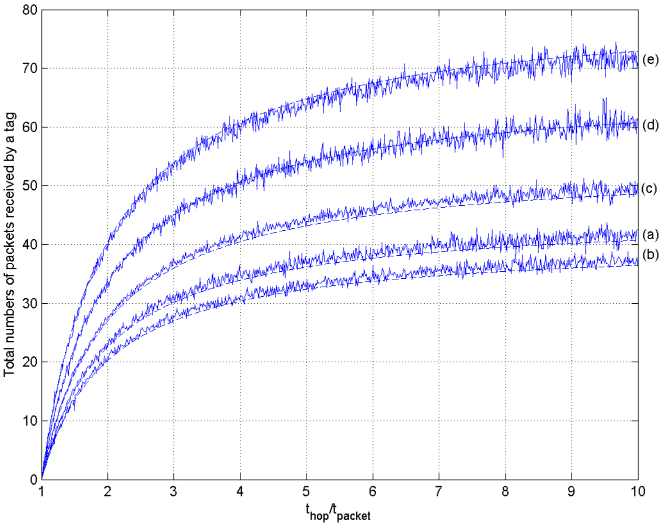


Figure 5.14: Effects of hopping time t_{hop} on the total number of receivable packets

For a good performance of the positioning algorithm it is crucial to process different values instead of several times the same value. This fact was taken into consideration, when only different values were counted during the simulation. Figure 5.15 illustrates the amount of different values from DU 5 for the reference positions (c),(d) and (e), which lie all within an acceptable distance (Zone 1) of DU 5. DU 5 is most suitable for comparison reasons, because the data of DU 5 is only transmitted once during one cycle. The values from DUs which transmit more often show a comparable behaviour but introduce also

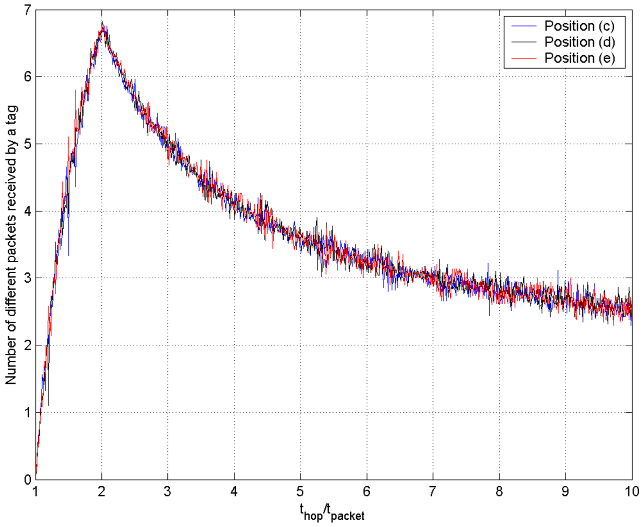


Figure 5.15: Effects of hopping hime t_{hop} on the number of different received packets

side effects, which result in a higher margin of deviation. In contrast to the total amount of received values, the amount of received different values is decreasing with a higher hopping time. After a steep rise until the doubled packet time, the amount of different values fades out. The time between two consequent listening periods on the same channel grows with increasing hopping time. This fact prevails the advantage of listening longer for packets on the same channel. For all three positions the peak value lies consistently at the doubled packet time. Because a high variety of values is preferred to a overall high number of values, this simulation leads to the recommendation to set the hopping time twice the packet time ($t_{hop} = 2 \cdot t_{packet}$).

5.4.4 Impact of Signal Repetitions

This experiment should clarify the impact of different values for nr_{rep} (the number of signal repetitions per unit) on the system. As indicator for this experiment, the received number of values per tag was taken. For this purpose, a special DU and tag placement was adjusted. All DUs were given the same coordinates in a centre. The rotation range of the first DU was set from 0 to 20 degrees, for the second from 0 to 40 and so forth, increasing the rotation by 20 degrees each further DU. Tags were placed on a circle with radius 5 meters from the centre, the first on 10 degrees, the second on 30 degrees and so forth, again increasing by 20 degrees for every next tag. This setting is visualized in Figure 5.16. Although this DU placement is not capable of obtaining positions, it is very good for comparing the number of received packets at every tag. The tag on the first position (10 degree) is able to receive data from all DUs, the tag on the second position

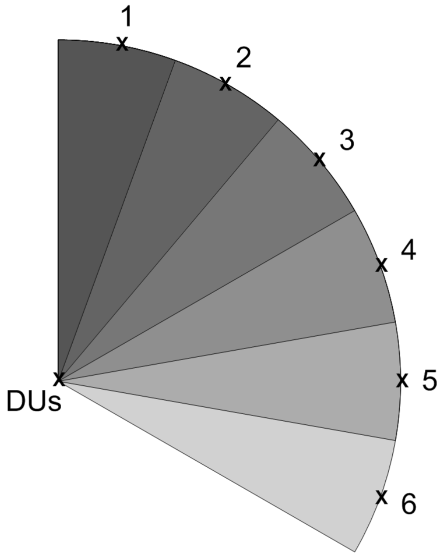


Figure 5.16: Setting with 6 DUs

(30 degrees) is able to receive data from all DUs except the first one and so forth. Finally, the last tag can just receive data from one DU.

In the simulation run, the number of tuples received by every DU was counted on each tag during one pass. As stated in Chapter 5.3, one simulation pass is depended on the maximum rotation angle of all DUs, max_{rot}, and the number of packets send per unit, nr_{rep}. The former dependence states, that there is more data sent for DUs with a low radial range than for DUs with larger rotation angles. Hence, to make all values comparable, the counted numbers were normalized to the highest appearing rotation angle max_{rot} by dividing them through the value of $\frac{max_{rot}}{rot_{DU}}$. After this angular normalization, the values were added up to nr_{norm}, the number of received packets at the tag. Finally, this number was normalized again due to the dependence on nr_{rep}. A higher value for nr_{rep} means a higher number of generated values, and therefore nr_{norm} was normalized by dividing it through nr_{rep}. By this way, nr_{norm} represents the number of received packets per max_{rot} degrees and per max_{rot} values for a tag.

The simulation run was carried out with six tags respectively six DUs, like the setting in Figure 5.16. t_{packet} was set to 1.72 ms, t_{hop} to 3.44 ms, α_{beam} to 15° with res of 1/° and nr_{rep} was going from 1 to 50. λ_{hop} is consequently $\frac{1}{2}$. nr_{rec} gives the number of receivable channels at the position. The expected value for each tag and one *cycle* was calculated with the following formula:

$$nr_{norm} = \frac{\alpha_{beam} \cdot res \cdot \lambda_{hop}}{nr_{channels}} \cdot nr_{rec} \tag{5.14}$$

In this simulation setting, this results into:

$$nr_{norm} = \frac{15}{2} \cdot \frac{nr_{rec}}{6}$$

The expected value for the first tag is therefore 7.5, which represents the maximum. The values for the other tags lie $n \cdot \frac{1}{6}$ of the maximum value beneath, independent of nr_{rep}. This independence is given due to the fact that the values are normalised. Figure 5.17 shows the average result of 200 simulation passes including the expected values indicated as dotted lines. As it can be seen, the simulation confirms the theoretical consideration that nr_{norm} is independent from nr_{rep}.

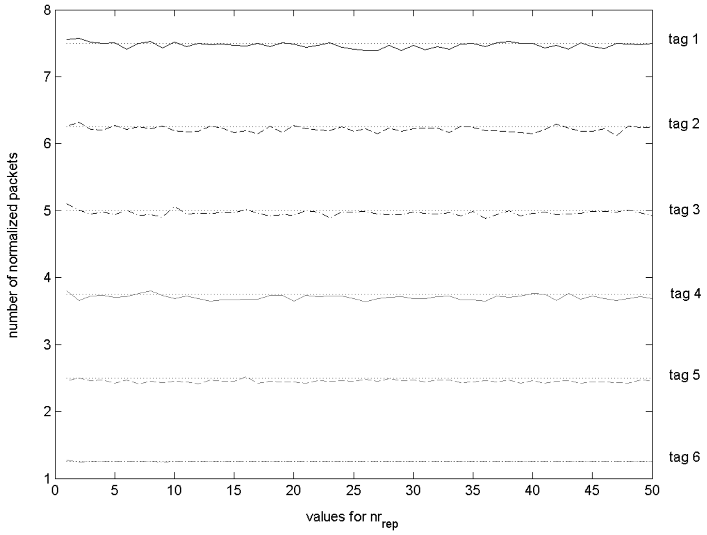


Figure 5.17: Effects of nr_{rep} on normalised received packets

Nevertheless, the total number of not normalised values is expected to increase linearly with an increasing value of nr_{rep}. The expected number of overall packets received by a tag during one *cycle*, μ_{rec}, is calculated by Equation 5.12. The simulated overall values are illustrated in Figure 5.18, where also the expected values are indicated as dotted lines. As it can be seen, the simulation confirms the stated formula. It has to be considered, that for the purpose of this simulation, *mem*, the memory of one tag, was set to infinity. In the real system and in later simulation, this memory is limited to 1kbyte giving 128 tuples, which are stored in FIFO fashion. The 128 tuple threshold is indicated in Figure 5.18 as dash-dotted, horizontal line. Equation 5.12 can then be used to adjust the maximum appearing μ_{rec} to the value of *mem*. μ_{rec} can be changed either by changing nr_{rep} or increasing *cycle*. Due to the given guidelines, the minimal time for one step of the rotation is 15 ms, which results in a minimal value of nr_{rep} of 9 for the real system.

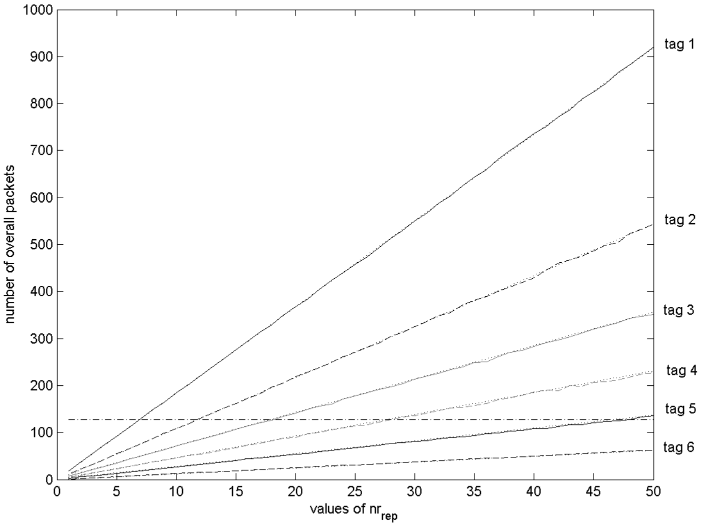


Figure 5.18: Effects of nr_{rep} on overall received packets

As final result of this simulation, the minimal possible value for nr_{rep} is recommended. As it was shown, there is no advantage of increasing the value, but there might be a drawback when exceeding the available memory. Anyhow, if for some reasons the available memory would not be utilized by far, the μ_{rec} can be increased by increasing *cycle*. Hence, for later simulations and the first prototype the minimal possible nr_{rep} of 9 is used.

5.4.5 Resulting Recommendations

As a result of the previous experiments on the communication protocol, some of the parameter values which were introduced in chapter 4.1 have been discovered. These settings are the proposed settings for a first prototype and are also the base for the upcoming experiments regarding the positioning algorithm. As hopping interval t_{hop} a value of two times t_{packet} is recommended, which results in the most diverse rawdata according to the experiment in chapter 5.4.3. The scanning interval t_{scan} was investigated to be reasonable when set to the time for one rotation of the maximal rotating DU max_{rot}. With $\omega = 64,6°/s$ this results in a maximal t_{scan} of 5.573s for a 360° radiator. This setting ensures reception of tuples from all possible DUs, when provided a proper hopping pattern.

The angular velocity ω, which is calculated by $(nr_{rep} \cdot res \cdot t_{packet})^{-1}$, is set to its maximum with 64,6°/s. This velocity is calculated with the minimal t_{packet} of 1.72 ms and the minimal possible nr_{rep} of 9, as discussed in Section 5.4.4.

The values for position interval t_{pos} respectively data exchange interval t_{dx} can be chosen according to the particular needs. It is recommended to set both intervals to the same value, because with higher t_{dx} it is very likely to exceed the tags memory capacity of 1kB respectively 128 tuples, which would mean information loss. For further application with possibly higher memory resources, it could be of advantage to choose t_{dx} as a multiple of t_{pos}, due to the fact that more rawdata usually leads to higher accuracy.

In this context also the power consumption of the tag has to be considered. The actually used battery has a capacity of 230 mAh. According to the specification of the currently used transponder, for sending data about 14 mA and for receiving about 19 mA are consumed. This results in approximately 780 positioning and data exchange processes for t_{scan}=5.573s and mem=1kB. When in *sleep mode*, the tag consumes 0.0037 mA, which is negligible compared to the consumption in the active phase. With 780 positioning operations it is obviously not possible to carry out real-time tracking of tags. With the current capabilities, this positioning system can be mainly used for static operations or *snapshots*. Beside the possibility of increasing the number of possible positioning processes by deployment of batteries with higher capacities it would be much easier to save power by decreasing t_{scan}. For the current hardware this can only be done by decreasing max_{rot}. With proper placement providing a max_{rot} of e.g. 180°, the number of positioning procedures would already be doubled, whereas in certain cases the average error would be likely to increase due to an introduction of additional DU-border areas. For further applications, it is also imaginable that t_{packet} could be decreased, which would lead to a shorter t_{scan} respectively a higher number of positioning processes as well.

5.5 Simulation of the Positioning Algorithm

In this chapter the positioning algorithm is optimised before carrying out simulation on different setups. These show the influence of certain adoptions on the performance of the algorithm. After incorporating the findings, a fictious scenario illustrates the capabilities of the system. At the end of the simulation a detailed statement about the accuracy of the system is made, which is one of the most important key figures of the whole system.

5.5.1 Border Shift Algorithm

A major error is introduced, if the processed group consists only of values, which originated only from a part of the full antenna spread. This case especially applies to boundary regions of every DU. For instance, if the boundary coincide with a wall, tags are only able to receive a reduced number of tuples from a DU. In worst case the tag is only able to receive the half of α_{beam}, if the tag is exactly situated at a angular boundary. Because of the missing values in the group, the calculated best value is computed falsely. More precisely, in an ideal case, the shift of the best value can be expressed as:

$$\Delta = \pm\frac{\alpha_{beam} - (\gamma_{max} - \gamma_{min})}{2} \tag{5.15}$$

The simulation recognises if a value group lies at an angular border of a DU, simply by checking if a value group starts or ends close (within 3°) to an angular border. In this case the range of the group ($\gamma_{max} - \gamma_{min}$) and the shift of the best values (Δ) are determined and added to the calculated best value. If the value group lies at the start boundary of a DU, Δ is taken negative, due to clockwise operation. Figure 5.19 illustrates the influence of the border shift. Both images show the average result of 20 simulation runs on the Standard Set. The right plot originates from test runs without the border shift algorithm, whereas the left hand side shows the results of test runs with the implemented border shift algorithm. In order to be able to identify the impacts, α_{beam} was set to 30°. Black areas indicate zones with an error bigger than 1 meter. The benefit of the border shift algorithm can best be seen from the marked area in the right hand side plot. DU 10 has an angular border in the direction of 30° in this region. The accuracy in the upper part of the marked area in the simulation with the border shift algorithm increases significantly. Although the improvement is considerably high, the border areas are still less precise than the core region of DU 10. This can be explained by the fact, that the algorithm has to process rawdata, which is corrupted by multipath and incomplete due to hopping.

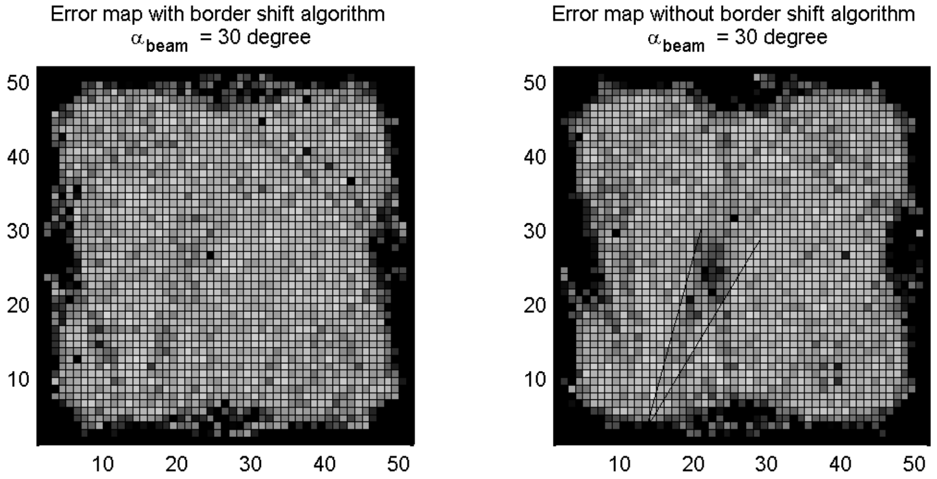


Figure 5.19: Impact of introduced border shift algorithm

Furthermore the improvements at the border are noticeable. Overall the mean error decreased from 81.89 cm to 76.33 cm and in the inner region from 39.42 to 36.63.

5.5.2 Optimisation of the Near Field Analysis

As described in Section 5.3.4.4, the positioning algorithm includes a near field analysis. The aim of this analysis is to find the intersection point with the highest rated neighbouring intersection points and discard single outliers. The weights of all neighbouring intersection points within a certain radius are summed up and compared to each other. The objective of this experiment was to find empirically the optimal setting for the radius in the near field analysis.

The experiment setup was based on the Standard Set. The values for the radius were chosen to $r = 0.5m$, $r = 0.75m$, $r = 1m$, $r = 1.5m$ and $r = 2m$. From all simulations, which were repeated 65 times, the average overall error, the average *principle* error and the error within the pivoting range of DU 10 were calculated. Before the calculations, all errors exceeding 1 meter were reduced to a value of 1 meter. Those values were not included for the calculation of the *principle* error, in order to be able to make a meaningful statement for the inner regions (without the border areas). The percentage of area, which is included in this mean value is quoted in braces. The average error in the pivoting range of DU 10 was included to evaluate the behaviour of the near field analysis in areas, which are densely covered. Table 5.4 illustrates some of the results from the experiment. The presented figures show clearly that a a radius of 0.75m gives the highest accuracy.

	avg. error [cm]	avg. principle error [cm]	avg. error in DU10 [cm]
$r = 0.5m$	48.7	23.8 (90.3%)	26.4
$r = 0.75m$	46.9	22.5 (90.5%)	25.9
$r = 1m$	47.5	23.3 (90.9%)	26.2
$r = 1.5m$	48.2	25.3 (90.9%)	29.1
$r = 2m$	49.8	27.4 (91.2%)	31.8

Table 5.4: Comparison of error figures for the near field analysis

Additionally to those figures the plots of the particular error maps emphasise the fact that a low radius (0.5m) leads to accurate result, with the drawback of a high number of outliers. A wide radius, on the other hand, increases the overall error, but reduces outliers. The error map for $r = 2m$ shows a uniformly distributed overall error over the whole map, whereas a very low error rate and outliers are dominant within the error map for $r = 0.5m$. Based on the results of this experiment, all following simulations were conducted with $r = 0.75m$.

The value of the radius is suited for scaling the overall system behaviour. Generally, it can be said, that for areas, which are covered densely by DUs it is preferred to set the radius to a low value, in order to profit from the high accuracy. For environments with sparsely covered DUs it is recommended to raise this value in order to avoid outliers. For further implementations this value has to be adapted to the particular requirements of the environment.

5.5.3 Illustrative Example

The described implementation is reconsidered by means of an illustrative example, which gives a better understanding and a deeper insight into the operation of the positioning algorithm. The exemplary position was calculated within the Standard Set. The tag was positioned at the point (38.5, 17) in the right lower quarter. At this position, which usually shows precise results, uninterrupted signals from four DUs (2,3,5,6) are expected. Figure 5.20 shows the test room of 50x50 meters and all signals, which successfully receive the tag. Those are indicated by a full line in light grey colour and are drawn from the position of the particular DU (represented as +) in the direction of the AoA. The picture gives a comprehensive overview of the rawdata model, described in the last preceding sections. The signals from DU 5 in the centre of the room include all phenomena and are suitable to review the configuration of rawdata. If a DU is in range the tag is usually able to receive a *full* group of consecutive tuples with a certain distance, due to hopping. In this example the *full* group consists of 7 values, what exactly corresponds to the expected value, given from 5.11. Due to $\lambda_{hop} = \frac{1}{2}$ only each second packet is received. In contrast to DU 5, the signals from DU 3 are closer together, due to the fact that DU 3 transmits four times

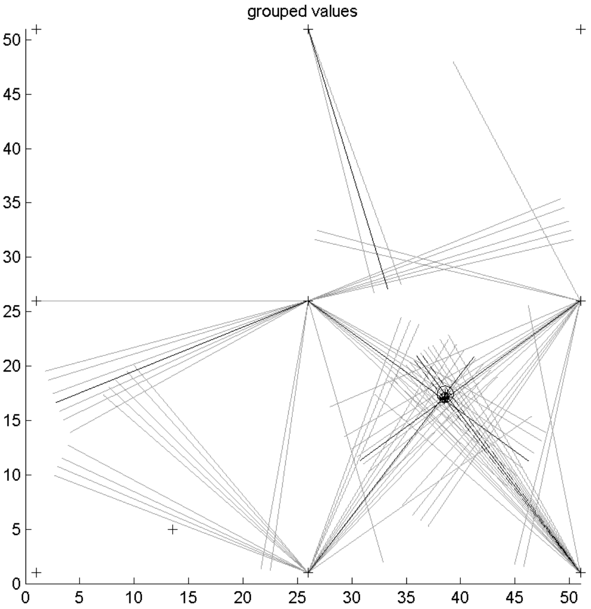


Figure 5.20: Example overview

more often than DU 5. The implemented propagation model incorporates additionally reflections and scattering. The signal of DU5 was reflected two times (approx. 70° and 240°) within a successive group and scattered four times as single value.

After sorting the incoming rawdata, the positioning algorithm builds logical groups of the rawdata and estimates a best value. A best value is generated for each group with the highest number of tuples from the particular DU. Additionally all groups containing 85% values of the maximum are included into the calculation. The estimated best values are indicated by full black lines. For DU 5 the group with maximum number of tuples (7) is the *correct* received group. The reflected group in the direction of approx. 240° consists of 6 tuples and is therefore also considered for further computation.

The calculation of the best value for the signals of DU 3 results in two best values (indicated by dashed lines), which are within 2 degrees. These values originate from one group, containing a gap in the sequence of values. To simplify the further processing these values are unified to one single value (full black line).

Overall, the complex and abundant information from all DUs are filtered and reduced to few single values. In this case overall 77 tuples are successfully received and only 6 values chosen to calculate intersection points, which are passed on to the plausibility check. Single multipath signals, weak signals from DUs, which are out of range and outliers are discarded in this process.

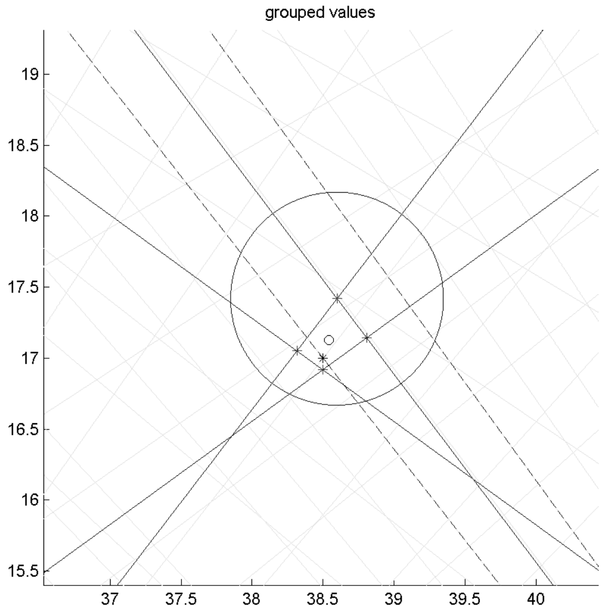


Figure 5.21: Example detail

Figure 5.21 shows a detailed close-up of the real position (indicated as $*$) within the same simulation. After weights are applied to all possible intersection points on the basis of the plausibility check, the near field analysis calculates the intersection point with the highest weighted values in the surrounding. The result of the example is shown as a circle around the *winning* point. Once more the amount of information is reduced by the means of plausibility. 15 possible intersection points are in this case cut down to four intersection point, which lie indeed close to the real position (indicated as $\circ$). Again single, multipath originated outliers are discarded and not considered for further computation, like the 240°-multipath group from DU 5. Only intersection points (indicated by $+$) within the circle are taken for the calculation of the final position. With the values of those four positions a weighted average is built, which corresponds to the calculated position. In this case the algorithm performed above average and calculated a position only 9 cm away from the real one.

As it can be seen by this example the main purpose of the positioning algorithm is to filter out valuable information and discard obvious wrong information. Furthermore secondary information, due to system parameters like antenna beam width or angular speed, is used as additional aid to ensure precise results.

5.5.4 Impact of different Parameters on the Accuracy

According to Section 5.2, five reference points were chosen to show the accuracy behaviour in different areas of the Standard Set. In Table 5.5, the mean principle error (in cm) of 1000 cycles is shown with different setups. The plus-sign indicates the use of a memory capacity of $mem = 512$ respectively 4kB, which was calculated to be sufficient for storing the maximum expected number of tuples (according to Equation 5.12).

Setup	(50, 37.5)	$\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$	(37.5, 37.5)	(31.48, 24.15)	(18.52, 24.15)
1. ideal	93.03	34.42	11.97	17.67	14.87
2a. normal	183.32	427.02	24.48	24.69	22.25
2b. normal +	169.74	393.42	23.87	23.81	20.79
3a. resolution	130.1	272.15	25.96	20.34	22.87
3b. resolution +	120.41	258.56	17.24	15.68	13.34
4a. spread	223.94	450.45	32.72	32.65	38.63
4b. spread +	234.68	480.43	26.76	23.38	21.15

Table 5.5: Mean error of five reference points

1 The *ideal* setup is based on the Standard Set with no multipath effects and an unlimited memory capacity. All further setups in this simulation are also based on the Standard Set.

2a In the *normal* setup, the parameters proposed in Chapter 5.4.5 are used, e.g. $res = 1/°$. Memory and beam width were set according to the actual hardware specification to $mem = 128$ and $\alpha_{beam} = 15°$. This setup can be seen as the reference setup for upcoming setups.

2b In *normal+* the memory is upgraded to $mem = 512$.

3a In this setup the *normal* setup was altered to $res = 2/°$.

3b In order to compare the results of the previous setup, this setup additionally introduces increased memory size ($mem = 512$).

4a With this setup the impact of an expanded antenna beam ($\alpha_{beam} = 30°$) is investigated.

4b The memory capacity was additionally increased to $mem = 512$ for comparison to the previous setup.

In the following paragraphs, the results are analysed. The focus lies at the positions (c), (d) and (e), because they lie in the important inner area. The other two positions are used to investigate the accuracy behaviour at the border of the area.

Position (a), (50, 37.5) lies directly at the border of the area and receives only signals from two DUs. For this reason it provides high errors, as expected. This imprecise accuracy is firstly based on the fact, that for data from two DUs just one intersection point is calculated, which increases the error accordingly. Secondly, the signals at this points are in an obtuse intersection angle towards each other which has an additional negative impact on the correct calculation of the position. Finally, a third source of error is given at this point, because at the border at most half of the possible signals can be received. This means, if a dominant multipath group exists, it is predictable that the positioning algorithm weights this group more than the correct "half-size" group. To reduce this effect, an enhancement of the algorithm was introduced in section 5.5.1. Even though this point was expected to provide the most inaccurate positions, it was discovered that point (b) had worse values, because in position (a) it is very likely to receive additional tuples from DU 8 and 5, which means a considerable increase of the accuracy.

Position (b) $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ lies very close to the left-bottom corner (with a distance of 1 meter) and is consequently in a region where bad results are expected, although it lies in the range of three DUs. This experiment reveals, that multipath has a crucial negative impact on the calculation of the position at this point. In Setup 1 it can be seen that without multipath an acceptable result of 34.42 cm is achieved. On the other hand, the results for the experiments including multipath signals debase dramatically. The deterioration due to multipath phenomena is much more serious than it is for point (a), because in case of (b) it is very unlikely to receive additional packets from other DUs.

Position (c) (37.5, 37.5) represents a position where exactly four DUs can be received under constant conditions which results in a very precise accuracy. Even though two intersection points are calculated with an obtuse intersection angle (DU 5 and 9 respectively 6 and 8 are facing each other) four intersection points can be calculated whose vectors are ideally perpendicular to each other, which means the most accurate results with the applied mathematical calculations. This circumstances lead to the best overall result in the *ideal* setup.

Position (c) also demonstrates the general effects of the different setups. Setup 3a has in fact good values due to an increased angular resolution of 2/°. On the other hand a lot of values are discarded, because a higher resolution also leads to a higher number of received values, whereas the memory capabilities are limited to 128 tuples, managed in FIFO fashion. The result without necessity to discard tuples can be seen in Setup 3b with an improvement of approx. 9 cm. Also compared to the *normal* setup with a resolution of 1/°, the effect of increasing the resolution can be seen with an accuracy-improvement of about 7 cm. Setup 4a results in an increase of the error of approx. 8 cm compared to the *normal* set, due to the doubled α_{beam}. This setting also generates the doubled number of tuples, which means that higher memory capacities are required. The results show, that without memory restrictions an accuracy decrease of just 2 cm occurs compared to the normal setup, even though the antenna spread is doubled.

Position (d) (31.48, 24.15) is in range of five DUs (2, 4, 5, 6, 8). The results are mainly similar to those of point (c). For the *ideal* setup without multipath, a less accurate value is gained because the angles between the vectors are not as ideal as they are for position (c). The main purpose of this position is to build a comparison to point (e), which lies on position (d) mirrored on the vertical centre line, and for that reason also lies within the range of DU 10.

Position (e) (18.52, 24.15) lies consequently in range of six DUs (2, 4, 5, 6, 8, 10). The results show, that in general the accuracy is better than the comparable point (d) due to more positioning information. Nevertheless, in the cases of Setup 3a and 4a a worse accuracy was investigated due to the limited memory capabilities in these setups. The increase of the error for this case is especially dramatic, because DU 10 has a very low rotation angle with just 60°. Therefor it passes by six times during one scanning interval, which is calculated for 360°, and "pushes" a lot of relevant data of other DUs out of the FIFO memory. The investigation of point (d) and (e) proves the scalability of the system, where the available memory resources still have to be considered.

After analysing the five reference points, the findings are summarised within the next paragraphs and proven with additional simulations. These simulations provide complete error maps representing the mean errors of several simulation passes on the certain point over the whole area with resolution of 1x1m.

First of all, it can be recognised that the mean error in the ideal setup lays at approx. 15 cm, which can be referred to as the "systematic error" of the system for an area of 50m x 50m. This error results mainly from reception of inexact angles, but also from information loss on the air interface.

With multipath introduced, the mean error for the *normal* setup is increased to the value of 24.4 cm. The complete error map for mean values of 50 passes within the *normal* setup is illustrated in Figure 5.22 and can be seen as the reference map for all upcoming error maps.

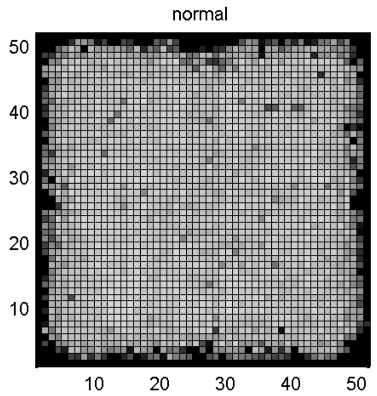


Figure 5.22: Reference error map, *normal* setup (res=1/°, α_{beam}=15)

In general, it was investigated that the positions (c), (d) and (e), which lie far away from the border areas, have an acceptable accuracy in average. It was predictable that positions near the border, like (a) and (b), are very critical for the algorithm, which was proven by this simulation. It was also revealed that it is desirable to increase the memory on the tag for certain DU placements or settings providing a high number of tuples. If the memory is sufficient, it is found that areas within the range of more DUs obtain better results, which shows that the system is scalable depending on the number and the placements of DUs. The comparison of the values of positions (a) and (b) respectively (c) and (d) showed that also the intersection angle between the resulting vectors can play a crucial part in the accuracy of the results, especially when there are not much other DUs in range. Obtuse intersection angles therefore should always be considered when planning the DU placement.

Another outcome of the reference point simulation is that a doubled resolution leads to about 30% more accurate results, provided sufficient memory. In contrast, doubled beamwidth of the antenna increases the error just very slightly when ignoring memory shortcomings. For proving this results, complete error maps with res=2/° respectively α_{beam}=30 were generated. In Figure 5.23 and 5.24 the mean error maps of 20 passes are illustrated for *mem* set to 128 and 512 each.

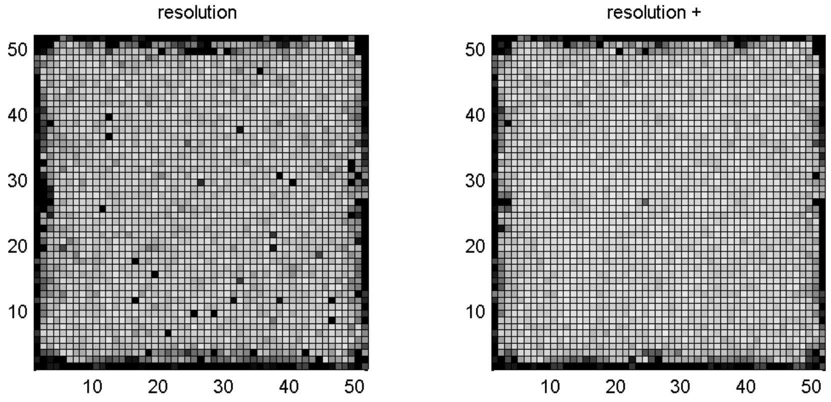


Figure 5.23: Error maps $resolution(res = 2/°)$

These simulation runs reveal that even though the above investigated conclusions are true, there is a main difference in the border areas of the DUs. Compared with the reference error map of the *normal* setup (Figure 5.22), a higher resolution leads just to a slightly more narrow inaccurate stripe at the borders, whereas a doubled antenna spread doubles the inaccurate stripe at the borders. An experiment, in which the advantage of $res = 2$ should compensate for the drawbacks of a doubled α_{beam} is illustrated in Figure 5.25.

The finding of this experiment revealed that even though the mean error could actually be decreased to a slightly lower value as for the *normal* setup, the inaccurate stripe at the borders could hardly be narrowed. Figures 5.23, 5.24 and 5.25 furthermore evince one

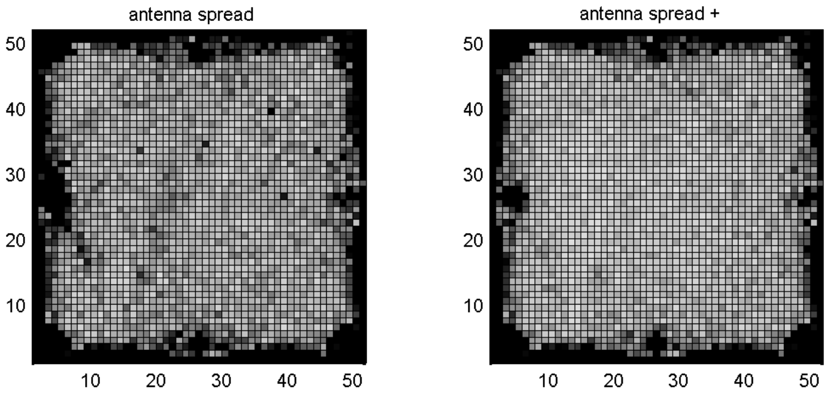


Figure 5.24: Error maps *angular spread* ($\alpha_{beam} = 30°$)

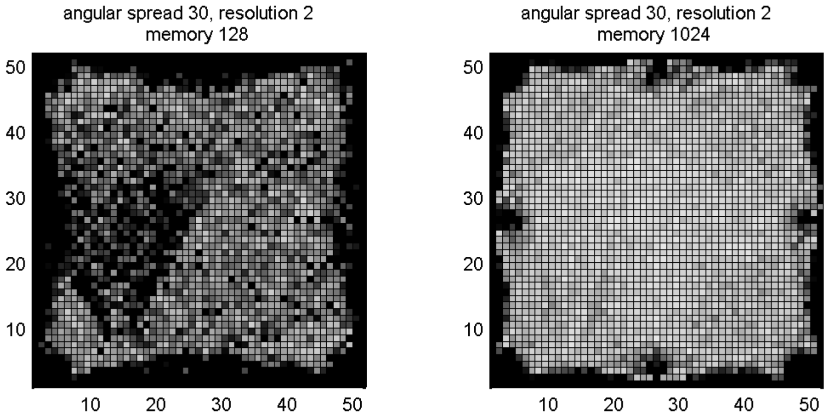


Figure 5.25: Error maps, $\alpha_{beam} = 30°$, $res = 2/°$

more time the importance of sufficient memory. As it can be seen for all three error maps with limited memory, the number of outliers increases with any setup providing a higher number of generated values. This is especially obvious in the left hand image of Figure 5.25, where the range of DU 10 introduces an area with very inaccurate values. Due to the fact, that the chosen setting produces 4 times more values than the *normal* setup, this area is most vulnerable because of the low rotation angle of this DU, as already described above in the discussion of position (e). These outliers can be eliminated by providing the necessary memory capabilities at the tag, as proven by the right hand figures.

5.5.5 Scenario Assembling Hall

In the following a scenario is presented that illustrates a possible application for the positioning system. The scenario is based on ideas from a swedish production site of an international company. Field of operation is an assembling hall, illustrated by the building plan in Figure 5.26. The building is divided into three parts: assembling area, warehouse and loading zone.

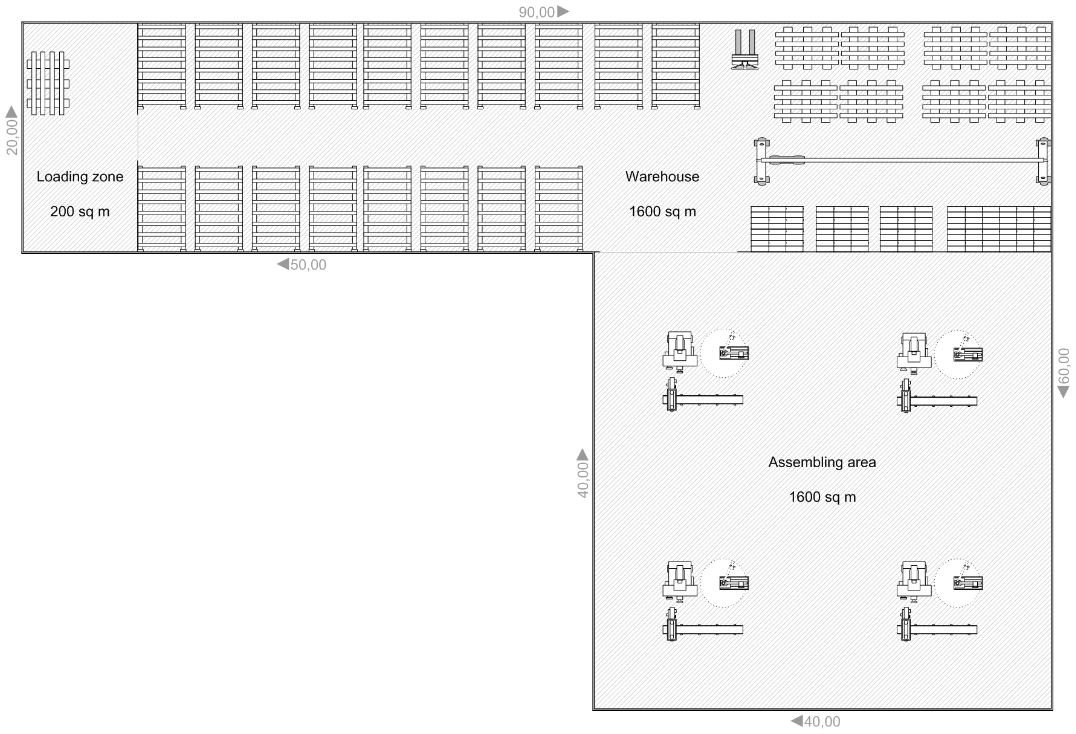


Figure 5.26: Scenario building plan

The assembling area is that part of the building where the production takes place. It is a 1600 m^2 room with different machines and tools. Aside from that, the room is sparely equipped and only few obstacles can disturb the signals.

The warehouse is the storage area of the building with also 1600 m^2 space. Different kinds of racks and containers are placed here as well as heavy machinery for arranging the goods.

The loading zone has an area of 200 m^2 and serves as station for loading goods. At five gates trucks can dock and pick up palettes. The retention period of the goods is short and the throughput high resulting in a fast changing environment.

RFID-Tags are already attached before goods come to the assembling area. After leaving the assembling area the parts are stored in the warehouse. Last station is the loading zone which serves as terminal for the goods where trucks can load wares. A retail shop next to the hall uses software that is able to connect to the tag and read out positioning information. Also the truck driver is using this software to look up the gate number on which the desired product is delivered.

Based on the diverse demands of the different sections of the building a different accurate position is desired. In the assembling area it is sufficient to gain rough localisation in order to obtain proximity to different work benches. In the warehouse higher accurate positions are required for an effective goods retrieval. The loading zone needs a high accuracy, because of the short retention period, the high throughput and the fact that the gates lie close to each other. As a result of these thoughts, 21 beacons are placed as illustrated in Figure 5.27.

The system requirements do not comprise real-time tracking. Therefore every 30 minutes a snapshot of the stock is taken and stored in a database from where the position information can be read. The highest occurring pivoting range within this setup (max_{rot}) is 180°, which leads to a scanning interval t_{scan} of 2.79s. Concerning the battery lifetime, theoretically 1560 positioning procedures respectively 32 days operating time are possible within this scenario.

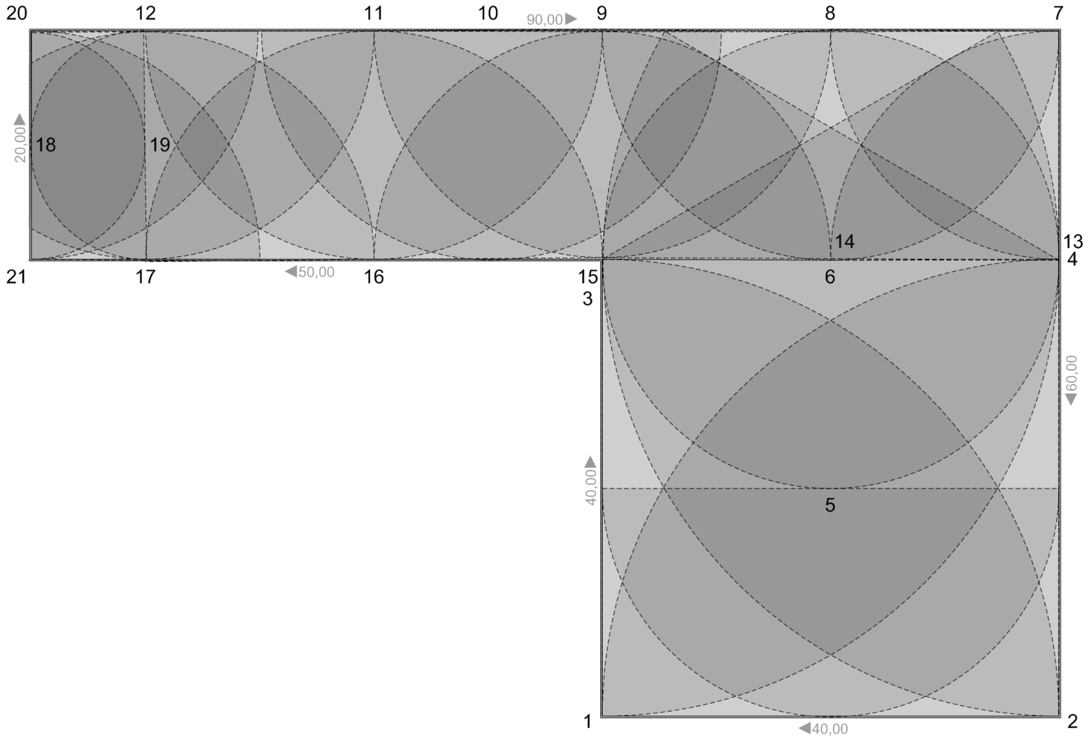


Figure 5.27: Scenario beacon placement

Due to the fact that in the assembling area mainly work benches with small machinery are situated, scattering effects are more dominant than reflexions. Position information in this room is not of big relevance, it is enough to determine proximity to one of the work benches with tolerable errors of up to 2m. Therefore a DU placement consisting of a few DUs with a high output power is chosen. DUs 1-4 have a nominal range of 40m, which results in a coverage of nearly the whole room. Additionally, two DUs (5, 6) with a nominal range of 20m are added to support a better coverage of the main area. As it was calculated during the simulation run for Figure 5.28, the resulting mean accuracy lies at 77 cm, without including the error prone border area.

The warehouse comprises basically 11 DUs (7-17). Neighbouring DUs like three DUs from the loading zone (18, 20, 21) and two from the assembling hall (3, 4) are also partly sending out signals to the warehouse area. Because of the dense arrangement of palettes and containers a high number of metal surfaces causes a very high probability for occurence of reflections. Nevertheless, the accuracy of the warehouse lies in average below 55.1 cm for the inner area, whereas for regions with higher coverage, an increasing accuracy can be recognised. The accuracy in this section shows that a proper DU placement can compensate for the shortcomings of an error prone environment.

The loading zone, where a high accuracy is desirable, is covered by six DUs (11, 16, 17, 18, 19, 20). Because of the low density of objects combined with the advantage of having no solid westbound wall this area is very unlikely for occurance of dominant multipath phenomena. This circumstances results in a mean accuracy of 20,6 cm.

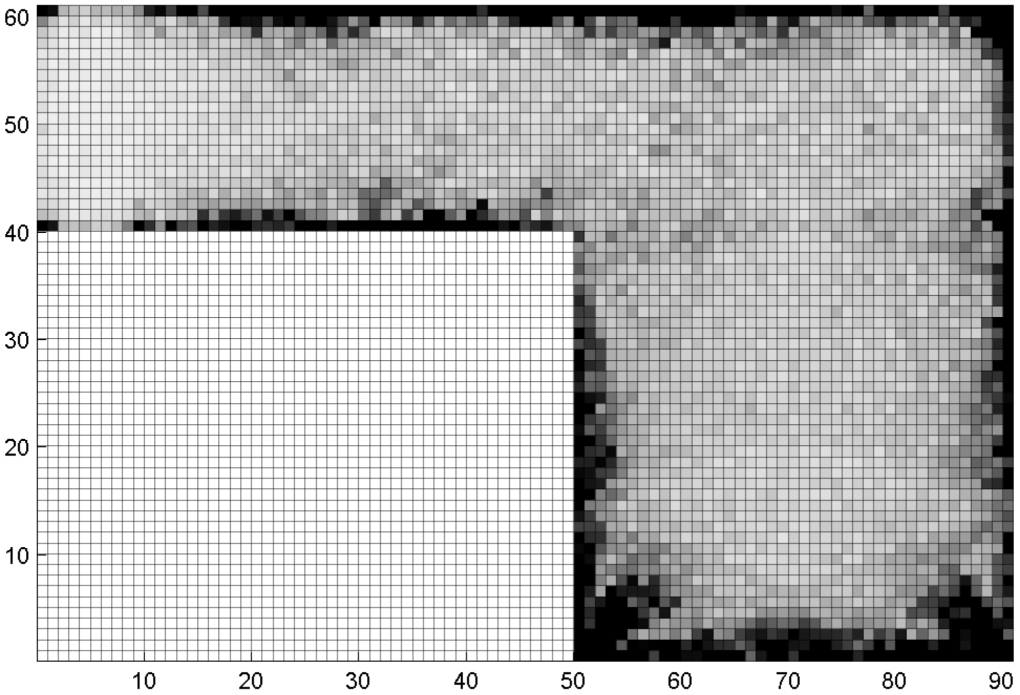


Figure 5.28: Scenario error map

This fictious scenario illustrates the capabilities of the proposed positioning solution. By varying DU placement and transmission range it is possible to adapt to different environments and needs. The solution for the assembling hall is able to provide seamless localisation with respect to particular requirements.

5.5.6 Accuracy Rating

A possibility to verify the correctness of a position algorithm is to compare an accuracy map with the corresponding error map. An accuracy map is generated by adding the weights of the intersection points within the near field to an accuracy value. This calculation is carried out for any position of the error map. In ideal case, the error should be low where the accuracy value is high and vice versa. Figure 5.29 shows the resulting accuracy map compared with the error map of the above described scenario. Darker colours indicate higher accuracy respectively lower errors, whereas bright colours mean low accuracy respectively high position deviation.

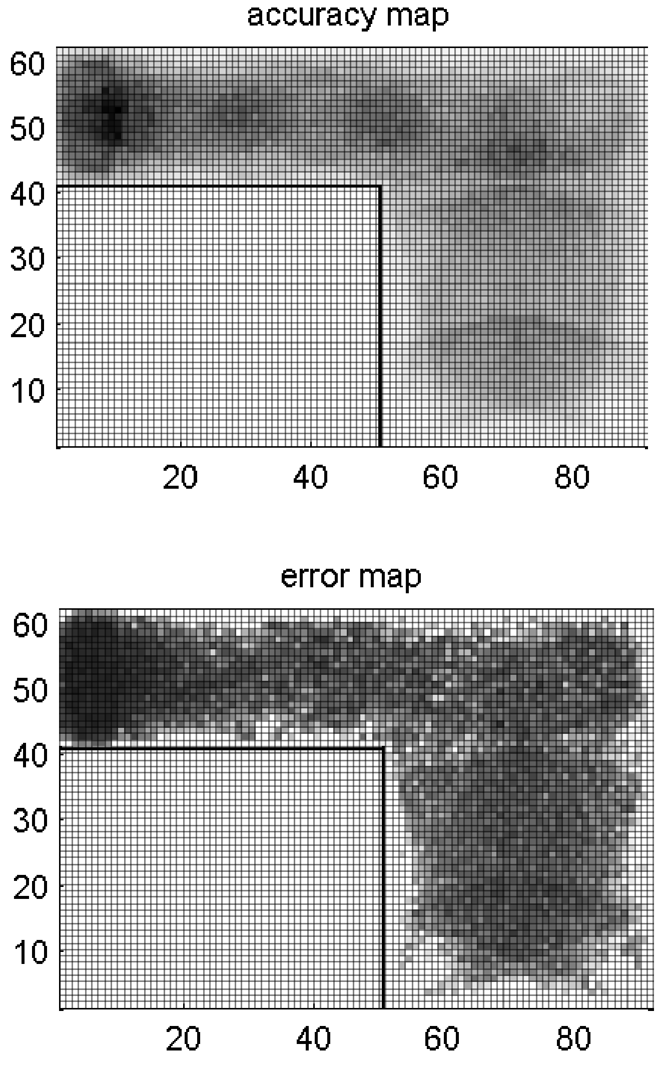


Figure 5.29: Comparison between accuracy and error map

As it can be seen, particularly the border areas of the warehouse evince very similar patterns. A closer look reveals a high similarity especially in the assembling area, where low accuracy leads to high errors, due to the placement of the DUs. Also the scalability of the system is proven again when comparing the loading zone on the far left side of the building. Here, a dense DU placement gives very high accuracy values, whose plausibility is manifested by very accurate results in this area.

The high conformity of the patterns not only verifies the position algorithm, but also gives the possibility to assign a quality level to the corresponding position. Such a rating is a very desirable additional feature, because in most setups a lot of different accurate areas occur. An accuracy rating could compensate for this shortcoming and give a reasonable recommendation about the quality of the resulting position. For a general predication which is valid for all setups, these accuracy values have to be normalised, which is not the case for the current implementation. An enhancement of this accuracy rating therefore is a suggested improvement, which was not carried out within this project work.

6 Conclusion

RFID is an emerging technology for the retail industry with a potential that is not fully utilised. Nowadays the focus lies on the integration of passive tags into the supply chain (e.g. EPC for globally identification of goods). Active tags offer, in contrast to passive devices, various options for new application fields due to higher memory and processing capabilities.

RFID systems gain a benefit by providing indoor position capabilities as part of the supply chain management. According to actual studies the majority of positioning systems are based on direct or indirect distance measurements by RSSI, ToA and AoA. Based on sophisticated indoor propagation models, RSSI can provide precise results by measuring the signal strength. The known velocity of propagation of certain carrier waves leads to highly accurate results for ToA-systems. Due to stringent hardware conditions both proposed concepts were not suitable for integrating positioning capabilities into the RFID system. Therefore an AoA concept was chosen for implementation in order to obtain directional information. Based on this information the position is calculated by applying methods of triangulation.

The main extensions of the designed system are directional units, covering the whole target area. These DUs rotate like beacons and send out AoA information within their pivoting range. The reception of signals from all DUs is guaranteed by fast frequency hopping on the tag. AoA data is temporarily stored in the tag's memory and periodically transmitted to readers, which are existing standard components. The information is redirected to a backend system, where the data is interpreted by a positioning algorithm. Finally, the algorithm computes position via a weighted mean value. The weights are assigned to resulting intersection points based on different plausibility considerations.

Based on this concept simulations were conducted to verify the approach. The system was implemented with the use of Matlab Script and different toolboxes. The first part of the simulation takes care of the generation of rawdata considering propagation errors, multipath effects and fast frequency hopping. The simulated rawdata build the interface to the implementation of the positioning algorithm.

The simulation framework was used to optimise system parameters within the communication protocol. The simulations gave a deep insight to the behaviour of certain parameters and resulted in recommendations for future implementations.

Furthermore, the positioning algorithm was tested taking into account the above gained results. After comprehensive simulations the impact of different parameters on the accuracy was analysed. For demonstration reasons an experimental setup was developed based on a fictive scenario. The positioning algorithm had to prove its performance and scalability in a warehouse environment. Areas with high density of DUs show an accuracy of 20.6 cm for 90%. The precision decreases continually for regions covered with less DUs. Six DUs are necessary to cover a room of $1600\,m^2$ with an average error less than 1 meter, which is still sufficient for proximity localisation. Multipath effects, which are usually dominant in an indoor environment, had a crucial impact on the accuracy.

Concluding it can be said, that the objective of the project could be satisfactorily fulfilled. It could be shown, that the developed system is capable of providing position information in an indoor NLOS environment using existing components and additional low-cost devices. The resulting average accuracy of up to 20 cm is a promising argument for a future experimental setup.

7 Future Work

The aim of the project was to prove the principal feasibility of a positioning system for RFID tags. During the conceptual phase a lot of ideas and enhancements had to be neglected, which could not be considered for this work. The most relevant ideas and further extensions to this project are proposed in the following chapter.

In order to verify the promising results of the simulations in the use case, the proposed concepts have to be realised. The results of a experimental setup should then be used for improving and adapting the simulation. Especially the effects and the amount of multipath propagation have to be tested in an experimental setup. Until now these values are only based on theoretical considerations.

The actual simulation program and the parameters are spread on several files. All values have to be set hard coded into the files. The development of a graphical user interface would enhance this shortcoming. Analysing the effect of different parameters, sketching the target area and placing DUs would become much more intuitively.

The design of an antenna for the DUs leaves a lot room for improvement. The system would benefit from an narrow main lobe, especially for the angular border regions. For the actual system is very simple servo motors are intended for the rotatable platform on which the antenna is mounted. Electrical steerable motors would improve the accuracy and stability of the movement.

As future advancement it would be possible to introduce an algorithm for semi-automatic coordinate discovery based on the ideas discussed in [20],[21]. This would simplify the installation process, because the exact position of a DU would be discovered automatically. The placement therefore becomes more flexible and can more easily be adjusted to the current needs.

All information sent on the air interface is timestamped. This information can be used for analysing the actuality of the received data. Furthermore timestamp information can be included into the plausibility check. The current version of the positioning algorithm does not take into account any time information. This would be an enhancement for future implementations.

The calculation time of the algorithm is critical for the whole communication process to meet real time requirements. Matlab Script was chosen for the actual implementation, because Matlab offers various possibilities for simulation purposes. This advantage turns into a drawback, when it comes to performance issues (e.g. the simulation of the Standard Set takes 35 minutes). An implementation in an advanced programming language would lead to an increased performance and would make simulations much faster.

The actual system was only implemented for 2D operation. Based on this concept an extension to 3D positioning can be made by introducing vertical DUs, which provide information about elevation. Because this would mean an increased number of necessary DUs, parts of the concept have to be updated. One option would be to distribute the vertical DUs in an SDMA manner and adapt the communication protocol to be able to distinguish between vertical and horizontal beacons.

References

[1] Metro Group (2004) Future Store Initiative. *http://www.future-store.org* (11th April 2004)

[2] A.T. Kearney (2003) Meeting the Retail RFID Mandate. *http://www.atkearney.com/main.taf?p=5,4,1,89* (11th April 2004)

[3] SearchMobileComputing.com (2004) Glossary. *http://searchmobilecomputing.techtarget.com/gdefinition* (04th February 2004)

[4] AIM White Paper (2000) Draft Paper on the Characteristics of RFID. AIM Inc. Frequency Forum White Paper. *http://www.aimglobal.org/technologies/rfid/resources/RFIDCharacteristics.pdf* (March 01, 2004)

[5] Savvides, Andreas et al. (2001) Dynamic Fine-Grained Localization in Ad-HocNetworks of Sensors. 7th ACM Int. Conf. On Mobile Computing and Networking (Mobicom). Rome, 166-179

[6] Hightower, Jeffrey and Borriello, Gaetano (2001) Location sensing techniques. A companion to Location Systems for Ubiquitous Computing

[7] Hightower, Jeffrey and Borriello, Gaetano (2001) Location Systems for Ubiquitous Computing. IEEE Computer, 34. 57-66

[8] Turing, G. et al. (1972) Simulation of Urban Vehicle-Monitoring Systems. IEEE Transactions on Vehicular Technology, Vol VT-21. 9-16

[9] Foy, W. (1976) Position-Location Solution by Taylor-Series Estimation. IEEE Transactions of Aerospace and Electronic Systems Vol. AES-12. 187-193

[10] Caffery, J. and Struber, G. (1999) Overview of Radiolocation in CDMA Cellular Systems. IEEE Communications Magazine

[11] Bahl, Paramvir and Padmanabhan, Venkata N. (2000) RADAR: An In-Building RF-based User Location and Tracking System. IEEE Infocaom 2000, Volume 2, 775-784

[12] Priyantha, Nissanka B. et al. (2000) The Cricket Location Support System. 6th ACM International Conference on Mobile Computing and Networking (ACM Mobicom). Boston, 32-43

[13] Harter, Andy et al. (1999) The Anatomy of a Context Aware Application. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking. Seattle, 59-68

[14] Want, Roy (1992) The Active Badge Location System. ACM Transactions on Information Systems (TOIS). 91-102

[15] Savarese, Chris and Rabaey, Jan M. (2002) Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks. Proceedings of the 2002 USENIX Annual Technical Conference. Montery, 317-327

[16] Buluscu, Nirupama et al. (2000) GPS-less lowcost outdoor localization for very small devices. IEEE Personal Communications Magazine, 7 (5), 28-34

[17] Doherty, Lance et al. (2001) Convex position estimation in wireless sensor networks. IEEE Infocom. Anchorage

[18] Niculescu, Dragos and Nath, Badri (2001) Ad-hoc positioning system. GLOBECOM 2001

[19] Enge, Per.; Misra, Pratap. (1999) Special Issue on Global Positioning System. Proceedings of the IEEE , Vol: 87, 3-15

[20] Savarese, Chris et al. (2001) Locationing in distributed ad-hoc wireless sensor networks. Proceedings 2001 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP 2001), vol. 4, 2037-2040

[21] Capkun, Srdjan et al. (2001) GPS-free positioning in mobile ad hoc networks. Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 9

[22] Nasipuri, Asis and Li, Kai (2002) A directionality based location discovery scheme for wireless sensor networks. Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications

[23] Dobkin, Dan (2002) Indoor propagation issues for wireless LANs. RFDesign: September 2002, 40-36

[24] Howard, Steven J. and Pahlavan, Kaveh (1990) Measurement and Analysis of the Indoor Radio Channel in the Frequency Domain. IEEE Trans. on Instrumentation and Measurement:39(5), 751-755

[25] Stallings, Wiliam (2001) Wireless Communications and Networking. Prentice-Hall, Inc., Upper Saddle River, New Jersey

[26] Robinson, Craig and Purvis, Alan (2003) Demodulation of Bluetooth GFSK Signals under Carrier Frequency Error Conditions. IEE Colloquium on DSP Enabled Radio, Livingston, Scotland

[27] Robinson, Craig et al. (2003) Characterisation of Bluetooth Carrier Frequency Errors. Proc. of the 9th IEEE IMSTW'03, Seville, Spain, 199-124

[28] Nordic VLSI (2002) Datasheet nRF2401 Single Chip 2.4GHz Radio Transceiver *http://www.nvlsi.com/index_popup.cfm?obj=misc&act=download&pro=64&prop=280* (2nd February 2004)

Symbols

α_{beam}	beam width of a antenna on a DU
β	intersection angle between two vectors
$cycle$	number of passes for one simulation
Δ	shift of the best value on border areas
ΔPos	position error for a shift of 1°
$dist$	distance between DU and tag
FL	floor loss
γ	actual AoA received by a tag
γ_{max}	maximum AoA within a group
γ_{min}	minimum AoA within a group
γ_{group}	average AoA, representing a group
γ_{real}	true angle between DU and tag
$groupTol$	tolerance value for row_{max} for best value finding
λ_{hop}	rate of successfully received packets
l_{ref}	mean number of repetitions for one reflection
max_{data}	maximum amount of received data from a certain DU
max_{rot}	maximum occurring rotation of a DU
mem	available memory size on a tag in number of tuples
μ_{DU}	expected peak value of received tuples for one cycle for one DU
μ_{rec}	expected total peak value of received tuples
$nr_{channels}$	number of DUs respectively channels in the current setup
nr_{norm}	number of received packets per max_{rot} degrees and values
nr_{rec}	number of receivable DUs
nr_{rep}	number of signal repetitions per unit of a DU
nr_{tup}	amount of received tuples per value group
ω	angular velocity of a DU
PL	propagation path loss
P_r	received power at the tag
P_t	output power of an antenna
$p_{scatter}$	probability of occurrence of scattering
p_{ref}	probability of occurrence of reflections
res	angular resolution of DUs
row_{max}	maximum occurring number of received tuples per value group
rot_{DU}	rotation of an antenna within its range
t_{dx}	time between successive data exchanges with a reader
t_{hop}	time for listening on one channel
t_{max}	maximum rotation time for one DU
t_{packet}	time for sending one packet on a DU in ms
t_{pos}	time between successive scans
t_{scan}	total time for listening on the channels
w	weight for an intersection point

Abbreviation

A	ACK	Acknowledgment
	AIM	Automatic Identification Industry
	AWGN	Additive White Gaussian Noise
	AoA	Angle of Arrival
B	BER	Bit Error Rate
C	CRC	Cyclic Redundancy Check
D	DU	Directional Unit
	DU-ID	Directional Unit Identifier
E	EIRP	Effective Isotropic Radiated Power
F	FDMA	Frequency Division Multiple Access
	FIFO	First in, First out
G	GFSK	Gaussian Frequency Shift Keying
	GPS	Global Positioning System
I	IC	Integrated Circuit
	IQR	Inter Quartile Range
	ISM	Industrial, Scientific, Medical
	ITU	International Telecommunication Union
M	ML	Maximum Likelihood
R	RFID	Radio Frequency Identification
	RSSI	Received Signal Strength Indicator
S	SNR	Signal to Noise Ratio
T	TDMA	Time Division Multiple Access
	TDoA	Time Difference of Arrival
	ToA	Time of Arrival
U	UHF	Ultra High Frequency
W	WLAN	Wireless Local Area Network