Masterarbeit

Implementation of an Acoustic Indoor Localization System for Wireless Sensor Networks

Christian Rathgeb

Institut für Technische Informatik
Technische Universität Graz
Vorstand: O. Univ.-Prof. Dipl.-Ing. Dr. techn. Reinhold Weiß

Begutachter: O. Univ.-Prof. Dipl.-Ing. Dr. techn. Reinhold Weiß
Betreuer: Dipl.-Ing. Philipp Maria Glatz, Bakk.techn.

Graz, im März 2010

Abstract

The goal of this work is a program for the localization of wireless sensor networks (WSNs) in an indoor environment. Up to now, the main point of research was primarily put on outdoor scenarios, which require a lower accuracy. Many indoor applications are only useful with spatial information. This makes it necessary to provide a simple but also exact positioning system to enable location based services. Power awareness is a major design goal for WSN applications, so the concept takes primarily a low power profile into account.

For the distance calculation, the different propagation speed of radio and audio signals is used for a time difference of arrival measurement. The necessary hardware is integrated in state of the art hardware platforms and the combination of both transmission media gives better results than only relying on electro-magnetic waves traveling at speed of light. The accuracy is affected by many factors, for example, not deterministic time delays of hardware and software. There are many variants to improve the measurement results, whereby statistic methods are supposed to suit best. Existing projects result in an accuracy of 10 centimeters, which implies a temporal accuracy of approximately 3 milliseconds.

With the gathered distance data, a final position calculation can be achieved. Different algorithms are discussed and compared for hardware requirements, amount of needed infrastructure and computational effort. Best suited is a infrastructure-based self-localization based on multilateration. Therefore, an over determined equation system is formed with the positions and distances to all reference nodes and afterwards solved by the singular value decomposition.

The presented implementation of the localization system allows, in combination with modular design and adaptive behavior, the active control over result precision by defining the amount of energy available to fulfill the task. This is primarily realized by redundant distance measurements and more complex position calculations.

Keywords: wireless sensor network, acoustic ranging, time difference of arrival, pulse sequence detection, localization, multilateration, power dissipation

Kurzfassung

Das Ziel dieser Arbeit ist ein Programm zur Lokalisierung von Sensorknoten im Indoor-Bereich. Bis jetzt wurde hauptsächlich an Outdoor-Szenarios geforscht, welche eine niedrigere Genauigkeit erfordern. Allerdings benötigen viele Applikationen eine hoch auflösende Position um ihren Zweck zu erfüllen. So ist es unumgänglich, eine einfache aber trotzdem genaue Positionsbestimmung innerhalb eines Gebäudes zu ermöglichen. Da Energieeffizienz bei Wireless Sensor Networks einen hohen Stellenwert einnimmt, wird bereits beim Konzept speziell darauf Rücksicht genommen.

Zur Entfernungsbestimmung wird die unterschiedliche Ausbreitungsgeschwindigkeit von Schall- und Elektromagnetischen Wellen ausgenutzt. Die dazu benötigte Hardware ist in vielen aktuellen Plattformen enthalten und die Messung einer Zeitdifferenz ergibt genauere Ergebnisse, als die Ankunftszeit einer Funknachricht zu detektieren. Die resultierende Genauigkeit wird von mehreren Faktoren beschränkt, darunter auch nichtdeterministische Verzögerungen von Hardware und Software. Um das Ergebnis genauer und robuster gegen Störgeräusche zu machen, werden die Daten mit statistischen Methoden gefiltert. Existierende Projekte erreichen Genauigkeiten von 10 Zentimeter oder umgerechnet 3 Millisekunden.

Mit den gemessenen Entfernungen kann anschließend die Positionierung der Knoten vorgenommen werden. Verschiedene Ansätze werden in einer Literaturstudie anhand von benötigten Ressourcen und Komplexität verglichen. Am besten geeignet ist eine Infrastrukturbasierte Selbstlokalisierung mittels Multilateration. Dafür wird ein überbestimmtes Gleichungssystem aus Positionen und Entfernungen der Referenzknoten erstellt und mittels einer Singulärwert-Zerlegung gelöst.

Das präsentierte Lokalisierungs-System ermöglicht mittels modularem Design und adaptivem Verhalten eine aktive Steuerung der erreichbaren Genauigkeit durch Vorgabe der zur Verfügung stehenden Energie. Dabei wird eine erhöhte Genauigkeit vor allem durch redundante Entfernungsmessung und komplexere Positionsbestimmung ermöglicht.

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....

date Christian Rathgeb

Danksagung

Diese Masterarbeit wurde in den Studienjahren 2008/2009 am Institut für Technische Informatik der Technischen Universität Graz durchgeführt.

Als erstes möchte ich mich bei Philipp Glatz für seine professionelle und freundschaftliche Betreuung dieser Arbeit bedanken. Vor allem durch die zahlreichen Gespräche innerhalb seiner Forschungsgruppe konnte ich mein Wissen im Bereich der drahtlosen Sensornetzwerke auch über die Grenzen der Arbeit hinaus erweitern.

Weiters möchte ich Prof. Reinhold Weiß und Prof. Christian Steger für die Ermöglichung der Abschlussarbeit und die guten Rahmenbedingungen am Institut für Technische Informatik danken.

Ich danke auch meinem Freundeskreis, der mir eine unvergessliche Studienzeit bescherte. Sie waren immer für mich da und mit deren Unterstützung im Studienalltag und der Freizeit, verging die Zeit in Graz außerordentlich schnell.

Der größte Dank geht an meine Eltern und Großeltern, ohne deren moralische und finanzielle Unterstützung dieses Studium sicher nicht möglich gewesen wäre.

Graz, im März 2010 Christian Rathgeb

Contents

List of Figures

List of Abbreviations

Chapter 1

Introduction

This thesis deals with a specific form of mobile ad-hoc networks (MANETs) called wireless sensor networks (WSNs). These networks consist of small scaled and energy efficient mobile devices that include sensors and are connected with a wireless link. The used devices are normally named *motes* or more generally *nodes*.

There are several differences between WSNs and the more general term of MANETs and the main field of application is one of them. Every MANET device is usually operated by a person who uses the networking feature in the background only. These devices are normally larger in size and can therefore support a greater energy source. If the source is depleted, the device can be reloaded at nearly no costs. By contrast, the wireless sensor nodes are distributed without an direct operator and should take care of their available energy for the whole expected lifetime.

The first fields of application for WSNs were all military, for example battlefield surveillance or sniper detection. After this the first mentionable research topic was agricultural monitoring, which motivated many famous universities to start research on WSNs. During the last years the enhanced WSN applications became more and more closely involved with pervasive computing aspects. These new application scenarios move WSNs from wide outdoor to relatively small indoor environments. This movement raises many insufficient solved problems, and finding a motes position inside a building is one of them.

Within a WSN a single node is only a small part of it. The real benefit comes with the growing size of the network. This leads to the next research goal, which is minimizing the costs of a node. This is mainly done by utilizing common of the shelf (COTS) components when designing new hardware for WSNs. Another possibility is not to use cost-intensive technologies when there are cheaper alternatives. When power awareness and price policy are defined to be the main goals when designing new platforms for WSNs, there is still another to mention. The size of a node should also be minimized. The main reason is the relation to ubiquitous computing where the technology should merge with the environment. Very small devices are a prerequisite for this goal.

1.1 Motivation

The main goal of this thesis is to develop an energy efficient and adequate precise positioning system, which can be used by indoor wireless sensor networks. As already mentioned military and environmental applications are traditional research topics for WSNs. Aside these, especially the indoor environment opens many additional fields of applications. For example health care in hospitals, medical elder care at home or home automation [Akyildiz02]. In a hospital scenario, this technology can for example be used to monitor expensive equipment to guarantee an optimal utilization or even to track the patients' movements between single therapies, also with the goal to increase the capacity of the facility.

Another promising topic is home automation, which has started yet but with other technologies. For example wireless light switches, intelligent refrigerators or complex heating installations with different sensors and actuators. Let us take a closer look on heating, ventilating and air conditioning (HVAC) installations. These systems usually need a very complex wiring to connect all environmental sensors and the actuators in form of radiators, ventilators and so on. The WSN technology would provide improvement especially for collecting the sensor data.

A further application for WSNs is as support for special task forces in rescue scenarios as mentioned in [Davids08]. This project combines a lot of research topics but the availability of sensor data in form of temperature and smoke detection and the position estimation is essential. The sensing task can easily by accomplished by sensor networks but the part of position estimation is still not solved sufficiently.

Another application area is industrial warehouse management. Useful information for such a system could be the exact temperature of reefer cargo, the current payload and position of the mobile stackers, or the current stock size. The great benefit when using WSN in general for this task would be the refresh period of the gathered information. Current systems based on bar code or radio frequency identification (RFID) technologies only make snapshots when moving through static gateways. When looking on the specification of higher class tags from the EPCGlobal organization, especially the Class 5 tags, the objectives and application areas between RFID and WSN technology come close together.

Knowing the nodes position would be a benefit to all the so far mentioned applications. A WSN that detects the position of the single nodes automatically has many advantages. The original motivation for this research was the high effort to deploy a large-scale WSN. For example in agricultural monitoring the gathered data of temperature and air humidity is only meaningful when combined with the exact coordinates in the field. Utilizing the global positioning system (GPS) is not the optimal solution for WSN because GPS receivers have a high power dissipation. Other disadvantages are the form factor and the price of such receivers [Bulusu00]. For this reason, alternative methods have to be found. One possibility is to make a heterogeneous network consisting of a few nodes that find their position on their own, either through calibration or absolute positioning systems like GPS. These devices are called anchor or reference nodes. Most of the network consists of nodes, which find their position based on these anchors.

When moving to indoor scenarios, the usage of GPS is infeasible, so other alternatives have to be found to provide reference points in the network. For this reason almost every indoor positioning system makes usage of a fixed infrastructure [Hightower01]. Most systems that do not rely on infrastructure cannot match the requirements to support pervasive computing aspects in indoor scenarios. The most challenging research goal is to reduce the number and complexity of the needed infrastructure on one hand, but still provide the same accuracy on the other.

When using an infrastructure, there is still the problem to position the mobile nodes. A main differentiation is to rely on ranging or not. So called range-free positioning algorithms are divided into proximity based and connectivity based methods [Pletzer08]. Since proximity based methods heavily rely on infrastructure only the connectivity based remain. Nevertheless, these algorithms have two major drawbacks, first the lack of accuracy because the range is estimated through hop-count or area elimination processes. The second is to find a trade off between high infrastructure costs or tremendous calculation overhead. Finally only range-based techniques remain.

In most cases, this means to include additional hardware exclusively for the ranging process. One exception would be the received signal strength indicator (RSSI) which can be gathered from most physical transceiver chips used in wireless sensor networks. Disadvantage is the corruption of this value by the environment. Ferromagnetic metal has a huge impact on the system accuracy, but there is the possibility to use sound waves as transmission media. The speed of sound is very low compared to electromagnetic waves, which propagate with the speed of light. This simplifies the correct and precise detection of the sonic pulses. A next step could be to move the detection into discrete hardware, which further reduces the workload of the microprocessor.

When deciding to use audible acoustic waves, instead of ultrasonic for example, another benefit can be noticed. The price of COTS hardware to produce and detect audible frequencies is less than for ultrasonic [Fogh04]. Since a long term goal of WSN research is to make single nodes as cheap and small as possible, this aspect is very fundamental. In addition, the form factor of microphones is less than that of typical ultrasonic receivers. Nevertheless, there is a major drawback when using audible frequencies. Unpredictable noise is more often present generated by the environment, in indoor applications mostly created by human interaction.

There is still the question which parameter should be used for optimization in a positioning system for wireless sensor networks. To optimize only in accuracy is nevertheless striking but in most of the cases neither power efficient nor cheap to deploy because of high effort in calibration or amount of needed infrastructure. Another possibility is to build a very power efficient system, a guiding idea for WSN technology. A better solution is to optimize for an accuracy-power-product as it is aimed by this project. As an additional feature, the parameters should be dynamically changeable during runtime to get the most accurate information with the currently available energy.

1.2 Further Outline

This master thesis describes the design and implementation of an acoustic indoor positioning system for WSN, especially for the Mica2 platform. The goal is to develop a very simple, and for this reason power efficient, positioning method utilizing a time difference of arrival (TDoA) method between audible and electromagnetic beacon signals together with a multilateration approach.

Chapter 2 provides background information about range-based distance measurements, available transmission media and different approaches to calculate the according position information. Further, the difficulties for synchronizing time in wireless sensor networks are explained. Another topic is the difficulty to measure and estimate power consumption in WSNs.

In Chapter 3 the concept and design of the positioning system are explained. After specifying the intended fields of application and possible test setups, this thesis describes the chosen infrastructure based network topology of the system. After this section, the used ranging method and afterwards the algorithm for position calculation are stated. A final section provides information about how the power profiling was accomplished and mentions some ideas for integrating the implementation into a middleware system.

Chapter 4 describes the finally implemented parts of the system. The software can be split into the ranging and positioning part. Both are described in detail including the needed test programs necessary to prove parts of the concept, specify constants or gain detailed test data.

In Chapter 5 the evaluation and experimental results are explained. They are divided into the main system blocks for ranging and positioning. After presenting the final results, some influences are discussed which have an impact on the precision or meaningfulness of the results.

Finally Chapter 6 summarizes this project and gives some ideas for future research.

Chapter 2

Background and Related Work

For traditional sensor networks the knowledge of the sensors' position is very essential, because most of the gathered data has no meaning without the context of where the data was taken from [Savarese02]. With the shift of application scenarios to indoor environments, this argument is still valid. Some of the constraints are changing, so the typical dimension of the network normally scales down whereas the needed positioning accuracy increases. This point is very important when using indoor WSNs for pervasive computing domains.

This is why there exist so many different solutions for the localization problem. It is very challenging to design a system that achieves good performance in indoor and outdoor scenarios, because both environments have different noise parameters for example. For range-based systems, a good decision is to decouple the ranging process from the position calculation [Whitehouse02]. This approach has two advantages. First, the two methods can be designed and optimized individually. Second benefit is that one of the two system parts can be exchanged easily by another algorithm or method.

First, this chapter gives an overview about common methods to obtain ranging information in wireless sensor networks. Afterwards the problem of clock synchronization in an ad-hoc network is illustrated, which commonly exists for various ranging methods. After that, the acoustic ranging process is described precisely including the effort to identify and cancel different noise factors. Furthermore, common positioning approaches are discussed, differentiated by the use of infrastructure, the kind of needed ranging data and the strategy for calculating the results. Finally the method to measure or estimate the needed energy of a WSN algorithm is stated.

2.1 Ranging Methods

This section will discuss common techniques to calculate distance values. First, there is the possibility to use physical laws that depend on the distance such as radio signal strength. Further the distance can be calculated based on bearings or angular information. Finally, different approaches to calculate a distance from time measurements are presented.

2.1.1 Received Signal Strength

One of the most common approaches for range estimation is based on the RSSI, because most modern transceiver chips grant access to the measured RSSI value. During the propagation of a radio message, the environment attenuates the signal. In ideal environments the signal strength falls off exponentially [Ji04].

$$d = \frac{\lambda}{4\pi} \cdot \sqrt{\frac{P_T \cdot G_T \cdot G_R}{P_R}} \tag{2.1}$$

This relation is described by the free space propagation model in Equation (2.1) which is translated to directly calculate the distance between transmitter and receiver [Pletzer08]. λ stands for the wavelength, P_T and P_R for transmitter and receiver power and G_T, G_R for the antenna gains. This model does not match with the reality. Two major drawbacks are not concerned in this formula, the case of multi path propagation due to signal reflections and disturbing obstacles that attenuate the signal additionally. For this reason, the modeling of the propagation behavior is a challenging research point and various models exist in literature [Kang07, Bahl00].

After specifying a propagation model, the path loss parameters can be found. This is normally done offline and the parameters are fixed. This process can be done automatically by a self calibration protocol [Kang07]. First, the beacon, which know their position try to range their distances with RSSI. The deviation can be used to configure the propagation model.

If the ranging process is integrated into the positioning system, another approach can be used. When considering a typical scenario for signal strength based indoor positioning, several beacons are placed across the building. In this case, every point which receives at least three different beacon signals has a typical but not necessary unique RSSI value signature. These signatures can be measured during an offline phase for several points, typically uniformly distributed. This information including the corresponding coordinates is stored in some kind of database. At runtime, positions are calculated as follows. After measuring the current RSSI signature, the database is queried for close matches. The matching positions are averaged based on their analogy to the actual measurement. This system perfectly models the signal attenuation relating to walls and permanent obstacles, but neither considers the direction of the receiver nor the presence of people which has great influence on the measured RSSI. The stated method is used in the RADAR project by Microsoft Research [Bahl00]. The system operates with a laptop computer equipped with a 2.4 GHz industrial, scientific and medical (ISM) band network card.

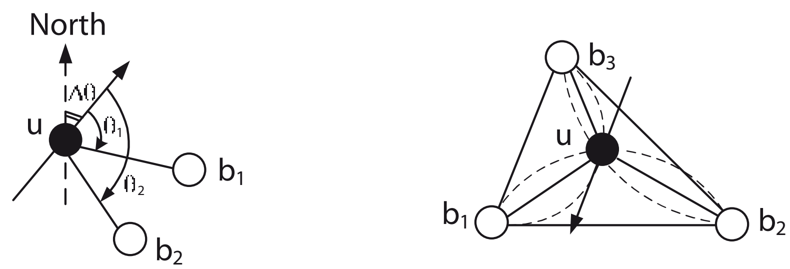
To improve accuracy of RSSI based ranging another method can be used. Attenuation and reflection of radio signals is strongly based on the used frequency and the transmission power. This features are used by the MOTETRACK framework [Lorincz07]. The transceiver chip has the ability to send on different frequency bands at different power levels. When combining this signal variation into the signature database, a further improvement in accuracy can be achieved.

2.1.2 Angle of Arrival

With the angle of arrival (AoA) approach distances between nodes can be calculated using trigonometric functions with given angles to or between reference nodes. Algorithms that calculate the position of nodes based on AoA often make additional use of distances between nodes. This is the cause why this method resides in this chapter.

It is difficult to obtain AoA information in resource constraint networks without complex additional hardware [Battelli07]. As signal media, radio or acoustic waves are used mostly. When designing such a system, either the receiver has to be capable to detect the incoming signal direction or the transmitter has the ability to control the direction of the emitted signal. The latter case is used in the very high frequency omni directional radio range (VOR) system for aircraft navigation [Niculescu03a]. Fixed base stations emit an omni directional signal, which carries the current direction information. An aircraft receiving two VOR signals can calculate the distances to the base stations and the own position. The same approach can be used in sensor networks [Nasipuri02]. This system is based on beacons that emit a continuous radio signal, which rotates at a fixed angular speed. All beacons are synchronized and have different offsets for the transmission direction. As proposed by Nasipuri, three different beacons are sufficient to calculate the distances to the beacons and calculate their position. The main error results from the assumption that the emitted beam width is zero.

There are two common AoA scenarios when the receiver can measure the direction of the incoming signal [Rong06]. They differ in the capability of the mobile node to estimate its own direction to north via a electronic compass. In Figure 2.1(a) the orientation of the mobile node is known. After measuring the bearing to at least two beacons, the position can be calculated. When no orientation is available as depicted in Figure 2.1(b), the relative AoA information can be transformed to the included angles between the beacon can be calculated and afterwards their distance to the mobile node.



(a) AoA with known orientation needs only two beacons.

(b) AoA with three beacons gives the orientation as additional result.

Figure 2.1: Two topologies for different angle of arrival techniques [Rong06] with absolute or relative angular information.

The construction of a AoA capable antenna is mostly associated with high hardware complexity. One possibility is to use antenna arrays. When analyzing the single signals of the array with multiple signal classification [Schmidt86] the AoA can be calculated.

Another possibility is based on ultrasonic beacons and is used in the Cricket Compass project [Priyantha01]. In this case, two receivers separated by a defined distance measure the exact time difference between the sensors. Based on this time the AoA can be estimated, but there is the possibility of an ambiguous result which can be solved by using 5 receivers placed in a 'V' shape.

Audible frequency ranges can also be used for AoA measurements [Girod06]. In this case four microphones are mounted on a square platform of 10 cm with one device raised above the others. With this arrangement, the azimuth and zenith angle of arrival can be detected although the necessary computation needs an embedded processor capable of running Linux and specific signal amplification hardware. Yet the accuracy is around 1.5° for both angles.

2.1.3 Time of Arrival / Time of Flight

Another possibility to obtain distance information is the measurement of the time between emission and reception of a signal. This method is called time of flight (ToF) or time of arrival (ToA), where both terms are used in literature for various measurement strategies. The main challenge is to detect the time of signal arrival as accurate as possible. Acoustic signals are commonly used for transmission because of the relatively low propagation speed in contrast to the speed of light for radio transmission.

Measuring the round trip time is a very simple possibility for distance estimation [Pletzer08]. The mobile node transmits a beacon signal and concurrently starts a timer. A reference node receives this beacon and replies with a beacon signal after a predefined time Δt. When the mobile node receives this signal, it stops the timer at t_1. When considering propagation speed of air with approximately $c_{air} = 343\ m/s$ the distance d can be estimated with Equation (2.2):

$$d = \frac{t_1 - \Delta t}{2} \cdot c_{air} \tag{2.2}$$

This method has drawbacks in accuracy and power dissipation. Two times a signal has to be sent precisely at the defined time and has to be captured with high accuracy. This results in four time measurements, each of them containing independent noise. The second disadvantage is that the beacon signal has to be sent twice. When using acoustic devices, a pulse consumes more power than a radio transmission that can be received in the same distance. Therefore, the amount of acoustic beacons has to be minimized. A system relying on round trip time measurement based on electromagnetic waves is the distance measuring equipment (DME) system used for aircraft navigation in combination with the already mentioned VOR. Range finders based on ultrasonic pulses are another possibility for distance measurement using ToF [Jordt06]. In this case, the propagation time to an obstacle, capable of reflecting the signal, and back is measured and the according distance is calculated.

A second variant mostly denoted as ToA is based on the time of a single signal propagation. In this scenario all nodes have to be tightly synchronized [Girod01, Kushwaha05]. Mechanisms to achieve such a synchronization are described in Section 2.2. When measuring ToA the time of signal departure t_t has to be either correctly predicted before execution or measured after execution. The obtained timestamp in respect to the synchronized network time has to be transmitted to the receiving node. The receiving mobile node has to measure the exact timestamp t_r when the signal arrives. Now the distance can be calculated following Formula (2.3):

$$d = (t_r - t_t) \cdot c_{air} \tag{2.3}$$

This variant has the benefit that only two exact time measurements have to be done. This reduces the noise of the ToA values. In literature, the term time of flight is often bounded to another approach. A Beacon sends a radio message concurrently when emitting an audible or ultrasonic pulse. The receiver starts a timer when receiving the radio message and stops it when detecting the pulse. Although the radio time of flight is not taken into account, this approach is typically referred as TDoA system described in the following section.

2.1.4 Time Difference of Arrival

The TDoA approach for distance estimation can be divided into two basic principles; both of them deal with a time difference measurement. One method is to take advantage from the propagation speed of two different media. These are normally the speeds of light and sound. The difference is so high (factor of 10^6) that most systems neglect the time of flight for the signal traveling with speed of light.

$$d = (t_2 - t_1) \cdot (c_{light} - c_{air}) \tag{2.4}$$

Radio and acoustic messages are mainly used as media type, for example in the Cricket, Active Bat and AHLos projects [Whitehouse02]. All these projects use ultrasonic pulses for the distance measurement, whilst there are only few systems that use audible sound [Fogh04, Farrokhi05, Kwon05]. The distance calculation nevertheless works similar in both cases and is following Formula (2.4). A simple timing diagram is depicted in figure 2.2.

The network needs no form of time synchronization, because the receiver only counts the time between radio message reception (t_1) and acoustic signal detection (t_2). However, this benefit introduces more inaccuracies. There is the possibility that the radio package is not sent in the correct moment because of the shared media and the carrier sense multiple access (CSMA) control or the transmission fails due to bit failures and has to be resent.

$$\sqrt{(x - x_1)^2 \cdot (y - y_1)^2} - \sqrt{(x - x_2)^2 \cdot (y - y_2)^2} = c \cdot (t_1 - t_2) \tag{2.5}$$

Another TDoA approach is to simultaneously send two beacon signals from different distances [Xiao06]. The resulting time difference is proportional to the difference of the distances of both reference points. When three reference points exist, from each combination of two points a difference measurement can be made. This results in three hyperbolic equations, as formulated in (2.5), which can be used to calculate the unknown position

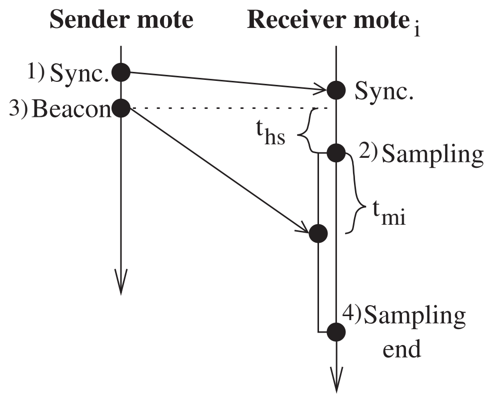


Figure 2.2: Timing diagram of a TDoA measurement which is based on two signals with different propagation speed [Fogh04].

(Figure 2.3). The advantage of this approach is that only a single media is used for positioning, so that the uncertain radio message time stamping can be avoided. A negative aspect is that the nodes have to be synchronized.

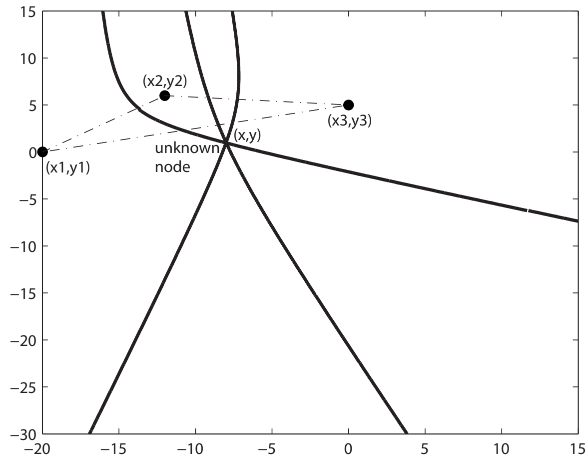


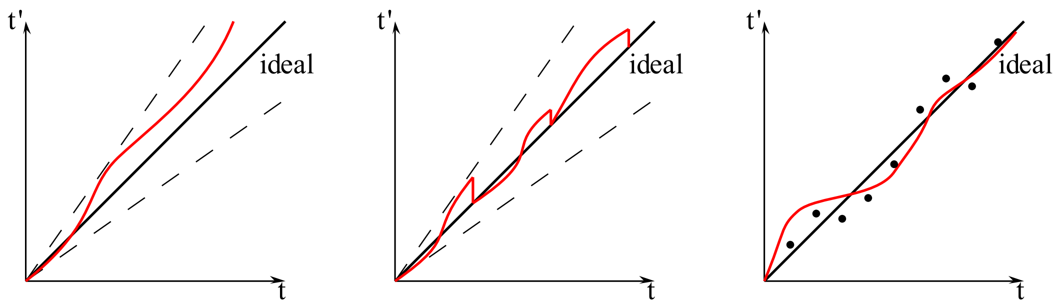
Figure 2.3: A TDoA measurement based on receiving two signals from different anchors. The captured time difference gives a hyperbolic equation on which the node can reside [Xiao06].

There is a possibility to avoid the synchronization problem [Hongyang05]. In this case more than three time difference measurement are taken. These differences are partly redundant, in the form that one distance is already known as range between two anchor nodes.

2.2 Time Synchronization in Ad-Hoc Networks

Time synchronization is an essential task in WSNs because many applications need to bind their sensing results to a specific timestamp [Römer01]. There are different requirements to the precision of synchronization. For example in a typical environmental monitoring application, an accuracy of seconds is enough to specify the results. By contrast, algorithms for acoustic source detection or distance measurement need tightly synchronized clocks, as proposed in Section 2.1.3. More complex radio stacks for wireless sensor nodes also require synchronization due to a time division multiple access (TDMA) organized media access control. Systems that have calculations with very low duty cycle are also targets for these algorithms, because the time of wakeup has to be counted very precisely to keep the power consumption low.

The problem of time synchronization splits into two tasks. The first covers the problem of offset calibration. Since typically all nodes in a WSN are switched on at a different time, a mechanism has to inform new nodes about the current relative or absolute timestamp in the network. An absolute time can be taken from nodes with equipped GPS or from gateway nodes which have access to network time protocol (NTP) based information, for example data sinks connected to computers.



(a) uncorrected clock drift (b) periodical offset correction (c) phase locked loop

Figure 2.4: Two approaches for the correction of clock drifts.

The second problem is clock drift, which arises due to inaccuracies of the used clock crystals. Measured in parts per million (ppm) this value is around $10-100$ *ppm* for typical WSN hardware [Römer05]. There are two approaches to cope with that. Either the clock drift itself is estimated and the time calculation is then correctly scaled to prevent further drifts, or the offset of the local clocks is synchronized periodically. The long time average error is accordingly lower with the first approach, as seen in Figure 2.4.

The problem of high time variance when sending radio messages in WSNs accounts for clock synchronization also. Typical latencies for WSN hardware are summarized in Table 2.1 [Maróti04]. A method to avoid the nondeterministic factors is medium access control (MAC) time stamping. In this case, the current time is saved just after sending the first byte and when receiving the first byte. With these times, the latencies of send, access and receive are almost eliminated. Most of the remaining factors depend on hardware or

Action	Time	Deterministic
SW send	0 - 100 ms	no
channel access	10 - 500 ms	no
encoding	100 μs	yes
byte alignment	0 - 400 μs	yes
transmission	10 - 20 ms	yes
propagation	< 1 μs	yes
decoding	100 μs	yes
interrupt	5 - 30 μs	no
SW receive	0 - 100 ms	no

Table 2.1: Typical latencies for wireless message transmissions

message size, so the variance of transmission time is much lower using MAC time stamping.

The reference broadcast synchronization (RBS) system [Elson02] considers offset and clock drift correction. A beacon signal is sent to all single hop neighbors. Now the algorithm considers only the relative reception times of the nodes. Since the propagation time is nearly negligible, the differences of arrival can be used to find a global time offset and each node can calculate the clock drift using 8 successive measurements using linear regression. The flooding time synchronization protocol (FTSP) also uses linear regression to adopt the own clock drift [Maróti04]. This algorithm relies only on MAC time stamping for offset synchronization. The energy efficient time synchronization protocol (ETSP) especially accounts for the strict power constraints in WSNs. The approach combines the RBS and the time-sync protocol for sensor networks (TPSN) approach. After analyzing the algorithms, constraints could be calculated which specify the better suited algorithm for the current scenario and network neighborhood [Shahzad08].

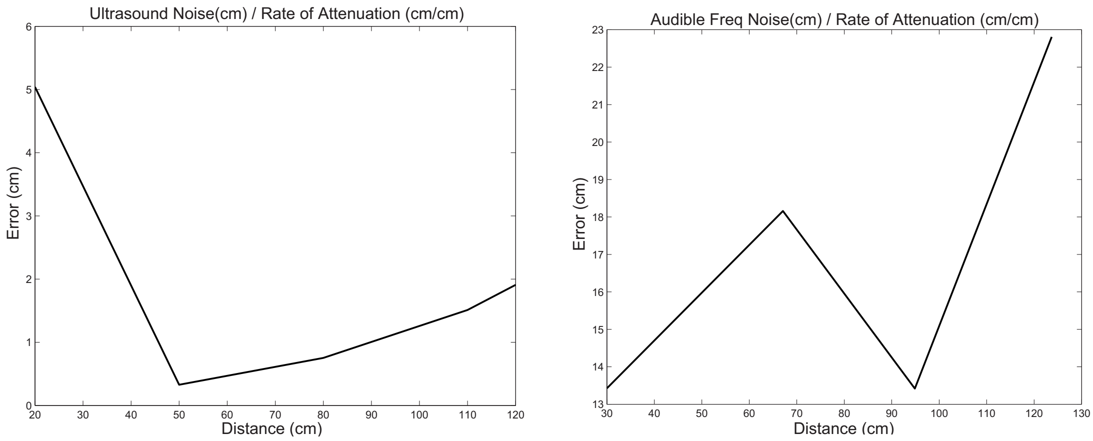
2.3 Acoustic Ranging

Acoustic based localization methods are widely spread, especially for fine-grained positioning requirements. The advantage over RSSI based methods is the accuracy. All acoustic methods require a line of sight connection between the nodes to work properly. Detecting or filtering non line of sight (NLoS) measurements that result from reflections is a very cunning task. Acoustic signals can be categorized into the ultrasonic and audible frequency domain, both of them having their own benefits and drawbacks. The challenges of signal detection and filtering still remain the same.

2.3.1 Ultrasonic Pulses

The frequency range of ultrasonic is defined as from 20 kHz up to 1 GHz. Most of the hardware used for range estimation uses only frequencies up to 200 kHz, for example echo sounding used for measuring water depth on ships. Typical COTS ultrasonic transducers operate at 40 kHz. The most common platform for ultrasonic based ranging in wireless sensor network is the Cricket project [Priyantha00]. The platform enhances the Mica2 platform with an ultrasonic transmitter and receiver. The first drawback of this technology is the form factor of the transducers, which is higher than audible frequency hardware.

A second disadvantage is the high directionality of both the transmitter and the receiver. For echo ranging this fact is a benefit, because the point to which the distance has to be measured can be controlled more easily. For the use as beacon one transmitter with a cone of around 120°, is not enough [Whitehouse02]. The signal power degrades rapidly outside the central axes of the transmitter and this fact is similar to the sensitivity of the receiver. One solution to the high directionality is using multiple transducer pairs mounted onto a single node, but this approach increases costs, size and power consumption of a mote.



(a) using ultrasonic pulses (b) using audible frequency pulses

Figure 2.5: Comparing the ranging error for different distances between the two acoustic frequency domains [Whitehouse02]. Ultrasonic provides a better accuracy than the audible frequency pulses.

An enormous benefit of the ultrasonic technology is the good accuracy, as depict in Figure 2.5(a). The measurements have been produced by common Kobitone transducers, and represent an average of 100 values. This fact makes the ultrasonic ranging technology ideal for scenarios where high accuracy in a narrow defined area is necessary. Further, there is the possibility to construct tracking applications, because the high accuracy reduces the number of samples and therefore the measurement time for a precise result.

The distance estimation with ultrasonic pulses is mainly supported by TDoA or ToA measurements. The maximum range for such systems is about three meters, but can be as far as ten meters, depending on the available hardware and power. The Cricket localization system for indoor applications, mentioned earlier, uses TDoA based measurements between radio packets and ultrasonic pulses, emitted from a number of fixed infrastructure nodes [Priyantha00]. These beacons are mounted on the ceilings, which reduces the impact of the transmitter directivity. The system achieves a good accuracy for positioning and for simple tracking purposes. Another design concept is privacy because the mobile nodes calculate the position on their own, allowing them to decide who should get the position information. A related approach is used in the Active Bat system [Harter02]. The mobile nodes are only equipped with an ultrasonic emitter and a radio transceiver. The sonic receivers are scattered on the ceiling and are all directly connected to a data sink. Since all receivers are synchronized several ToA measurements can be made and transformed to the emitters' position in three dimensions. The system is very accurate with measurement errors of under 9 cm, but suffers on the enormous amount of needed infrastructure which has to be placed and wired.

The ad-hoc localization system (AHLoS) also uses TDoA for distance estimation, with the difference that mobile nodes can calculate their position even on an ad-hoc basis [Savvides01]. The specially developed platform called Medusa is equipped with an array of ultrasonic receivers each of them heading in a different angular position. Experiments show a precision of 2 cm up to a range of three meters. For error correction and filtering the same approaches as in the infrared-based Active Badge system [Ward97] are used.

A system for very accurate object tracking based on the Cricket motes is presented by Ansari [Ansari07]. The estimated distances are gathered from TDoA measurements. A novel design is an automatic ultrasonic receiver calibration, which adjusts the analog signal gain and therefore the threshold for signal detection. To support tracking the system must be capable of filtering linearly changing distance measurements and this is done by using Bayesian filtering techniques. When using four beacon nodes mounted on the ceiling the two dimensional position of a train moving with constant speed can be estimated with an error of only 3 cm. This leads to a distance estimation of much less than 1 cm. Another project tries to improve the tracking capabilities of the Cricket framework in combination with inertial navigation based on a gyroscope and an accelerometer, with the motivation to reduce the needed amount of anchor nodes [Popa08]. The achieved result is an average error of 20-40 cm, depending on the density of the infrastructure.

2.3.2 Audible Sound

The second domain of acoustics is the audible frequency domain, which goes up to 20 kHz thus defined as the range that is hearable by the human ear. COTS hardware capable to emit or detect this type of frequencies is cheap and easy to buy [Palmer02]. For a few Euros, an adequate hardware can be integrated on wireless sensor nodes, as found on the MTS310 sensor board for Mica nodes. The main problem when using audible frequencies is the existent ambient noise, mainly produced by machines or human interaction. To cope with this problem and make the ranging process more robust against these interferences especially Section 2.3.3 gives solutions.

Another problem when using a COTS emitter in form of buzzers or speakers is to emit a sharp edged signal. Most hardware needs to tune into the intended frequency and amplification leading to a signal form as shown on Figure 2.6. This typically leads to less precise measurements in comparison to ultrasonic ranging methods. This can be seen in figure 2.5 where the audible ranging error is highly above the ultrasonic error, based on the same test bed and experimental setup. The measurement error grows logarithmically over

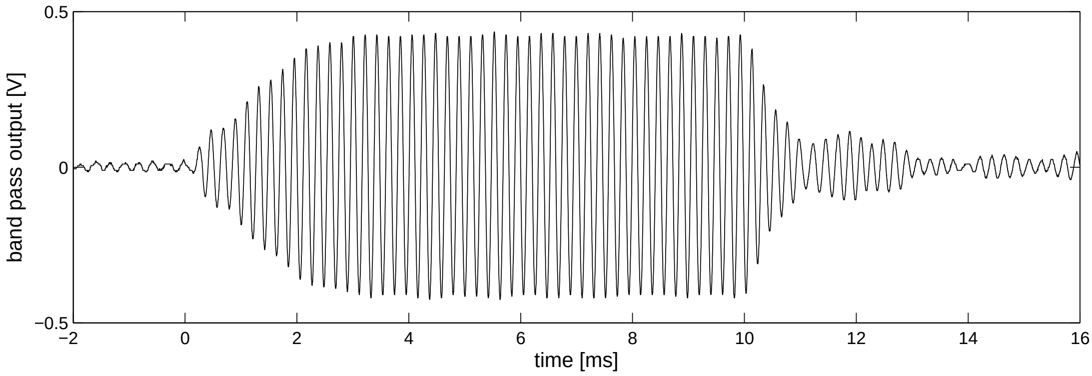


Figure 2.6: Output signal of the Mica2 sensor board microphone receiving a 10 ms audible pulse. The tune-in phase has a duration of 2 milliseconds.

the distance [Whitehouse02], which is in fact a drawback to RSSI ranging. Although both radio signal strength and the sound volume decrease in a logarithmic order over distance, the RSSI error suffers only from linear increasing error.

As it is with ultrasonic the most used ranging techniques for audible sound emission are TDoA and ToA. To provide better signal detection and direct control of the emitted wave shapes several acoustic ranging projects need high computational power. This gives the possibility to emit a linear frequency modulated (LFM) pulse (Figure 2.7) also designated as Chirp [Farrokhi05]. Multi path effects are very hard to filter out in an indoor scenario, but after the calculation of a cross correlation function (CCF) between the received signal and a local copy, this can be managed. Every multipath component results in a peak of the CCF, where the first peak is the accurate time of arrival. After additional filtering, the resulting error of this system is less than 5 cm for up to twenty meters of distance. This is accomplished by two computers equipped with a sound card and a DAQ-card. A very similar approach is done by Lewis Girod [Girod01], which is also PC based. In this case a special sound pattern was formed. The transmitter modulates a known bit-sequence using binary phase shift keying (BPSK) at 12 kHz. The used bit-sequences can be distinguished by a CCF even if they overlap in time. This makes it possible to emit different beacon signals at the same time. Further, this benefit can be used for concurrent distance measurements to different anchors making position estimation faster and more power efficient.

There also exist solutions that can be processed on wireless sensor nodes directly. Due to the cheap hardware, no sharp rising signals can be produced and the accurate detection of audible frequency pulses in noisy environments is very difficult. A specific signal pattern has to be produced to reduce the impact of false detections and Gaussian noise [Sallai04]. Such signal patterns are built with multiple acoustic pulses emitted from a buzzer as a fixed frequency. The duration of the pulses and the silence between is variable, but must be known by the receiver. The system proposed by János Sallai uses fixed length pulses with different times of silence in between. Increasing the number of chirps gives an improved signal to noise ratio (SNR). The detection is implemented in software as a band-pass filter

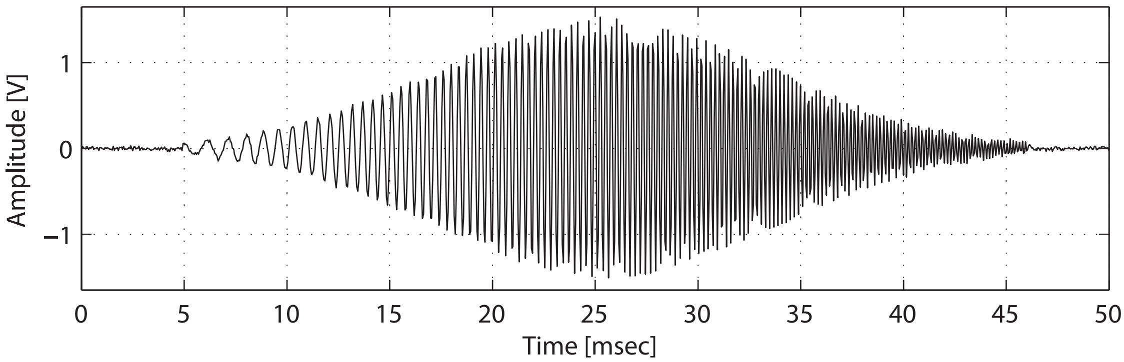


Figure 2.7: Signal form of a LFM pulse used for acoustic ranging in reference projects [Kushwaha05]. To detect the frequency shift an additional co-processor has to be used.

and after outliers correction and calibration the ranging error is a bit less than 10 cm up to a distance of nine meters.

The same approach for signal measurement, reading a continuous ADC data stream from the microphone and process data online by means of signal processing, can be found at Aalborg University [Fogh04]. The signal consists of only one pulse with undefined length and the challenge is now to find the exact beginning. This is supported by a mean squared error estimator and a Kalman filter both realized in software directly on the mote. The calculated position estimates have an error of around 5 cm on a playground of one square meter.

To this point, all approaches were supported by a fixed infrastructure. For large-scale outdoor scenarios, the concept of a mobile beacon equipped with GPS to support localization is feasible. These approaches are mainly based on ToA measurements and achieve good accuracy. Within the Thunder project [Zhang07] a helicopter is used as mobile reference node. The unknown nodes also sample the analog microphone signal and calculate the according distances as up to 150 meters. Over distance, the error rises from 50 cm to about 2 meters. The same concept of a mobile reference is developed at Vanderbilt University, but uses a special expansion board equipped with an 50 MHz digital signal processor (DSP) for the Mica2 mote [Kushwaha05]. In this case the ranging error decreases to about 25 cm for up to 40 meters distance.

When using the Mica2 platform with expansion board, the available analog tone detector with band-pass could be used to detect an incoming audible pulse. Although this approach is rarely used it achieves rather good results [Kwon05]. The system, developed for large outdoor scenarios, is equipped with a louder but still power efficient buzzer, which grants maximum distance ranging of 30 meters with environmental noise present. The main difference to other approaches is that in this case the plausibility of the interrupts from the signal detector have to be evaluated not the form or the signal itself. That means that a filter has to cope specifically with the false detections and outliers. The proposed statistical approach has an error of less than 20 cm in most of the cases.

2.3.3 Signal Generation and Detection

This section deals especially with the generation and detection of audible frequency signals. The discussed approaches are all capable to run on typical wireless sensor nodes and are therefore based on COTS hardware components and limited computational power.

Most of the research projects use the Mica2 platform form UC Berkeley which is equipped with an Atmel AtMega 128L micro controller running at 7.3 MHz and comes with a analog to digital converter (ADC) with 10 bit resolution. This type of node is very flexible and has the possibility to extend its functionality with expansion boards. For acoustic ranging an expansion boards exists with a fixed-frequency sounder, a microphone and a phase locked loop (PLL) tone detector with a band-pass on the input. The control of the sounder is binary so the only possibility to control it is switching on and off. The signal of the microphone is amplified with an adjustable gain and fed to an ADC port of the controller. The same signal is also filtered and given to the tone detector whose output signals an external interrupt of the micro controller. The center frequency and roll-off factor of the band pass are fixed and therefore affected by variances of the hardware components.

As already mentioned the generation of the signal cannot be influenced very extensive. Due to the fixed frequency, only the duration of the pulse can be adjusted. The common technique is to generate more than one pulse with specified durations and times of silence in between them. This so called pulse sequences are widely used for acoustic localization in WSNs [Sallai04, Kushwaha05, Kwon05]. The duration of the pulse can be varied but must be long enough that the signal can be detected by the receiving node and the sounder can tune in. Too long pulses need more energy or can produce multiple detections of the same pulse and should therefore be avoided. The duration of silence between the single pulses should be different or even varied for each pulse sequence. Figure 2.8 shows the pulse sequence as emitted (black) and as received (grey). For every pulse, the time difference is exactly the same.

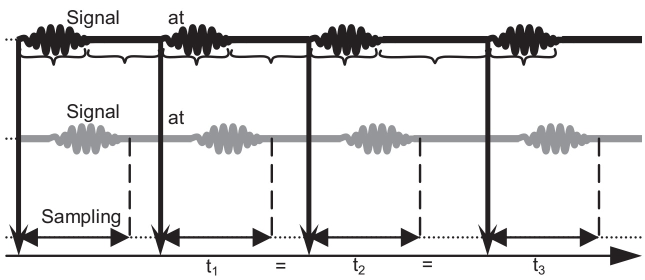


Figure 2.8: Example of a pulse sequence commonly used in literature with different times of silence between the pulses [Sallai04]. The time difference for each pulse is equal making the detection time more precise.

There are mainly two possibilities to detect an arriving audible frequency pulse. The first is the usage of an ADC to read a discrete data stream. This data is evaluated in real time by means of digital signal processing. This approach has the advantage that the time of arrival can be detected more precisely in most of the cases. This is accomplished by digital filters that can filter out Gaussian noise and the negative effect of the tune in process from the sounder [Fogh04]. Due to the high precision, there is no need of a pulse sequence, which saves energy for the emitting node and reduces the time needed for ranging. The drawback is the high computational effort for the signal processing.

Another alternative for signal detection is using an analog filter connected to a tone detector. In this case, the processor has no work to do but serving the interrupt from the detection circuitry. This reduces the power consumption because the analog components need less power than a micro controller operating at full workload. The used hardware is only designed to detect the presence of a specified frequency. More complex and noise resistant waveforms consisting of LFM or BPSK encoded information, as illustrated in Section 2.3.2, cannot be detected. Further, the false detection rate is very high with such systems. To counter this problem such systems are mainly used in combination with pulse sequences [Kwon05]. The more pulses are used for a sequence the higher is the improvement in SNR because Gaussian noise produced by the environment is independent for each signal pulse. The gain of SNR when using N signal pulses is stated in Formula (2.6):

$$\Delta G_{dB} = 10 \cdot \log N \tag{2.6}$$

The pulse sequences result in a series of external interrupts for the micro controller, which have to be time stamped. If the known pattern of pulses can be found in the dataset, a valid detection can be estimated [Sallai04]. When the time of silence is varied, the impact of the common phenomenon of multi-path propagation can be minimized.

2.3.4 Sources of Error

Depending on the type of signal detection, different metrics can be used to produce better ranging results. These methods generally compensate noise, average the system latencies or detect and eliminate false detections and outliers. Due to the very specific methods that depend heavily on the test and application setup, this section first introduces the different error sources and afterwards describes different existing solutions for filtering and error minimization.

The first error source is the system latency produced by the transmitter and receiver. When using TDoA methods this divides into radio transmission, acoustic pulse generation and detection, and the processing latencies. The magnitude and variance of a radio transmission are already stated in Table 2.1. When using ToA or ToF approaches the same latencies remain, although the error compensation moves to the time synchronization algorithm. As in Section 2.2 mentioned already, MAC time stamping strikes out most of the non deterministic delay factors. To reduce the interrupt response time on microprocessors only one guideline can be given. Since typical controllers cannot queue interrupts, all interrupt service routines have to be kept as short as possible. The longest routine gives the upper bound of the introduced latency. Further, the payload of the radio messages should be minimized which is relatively simple when using a TDoA approach because the synchronization message only has to contain an identification number.

The delay of the acoustic signal divides into the transmission time itself and the hardware latencies of transmitter and receiver. The delay for controlling the sounder is very deterministic because on most controllers this is simply toggling a general purpose input/output (GPIO) pin. The receiver side has a more complex behavior especially when using a PLL tone detector [Whitehouse02]. The greater the volume of the received signal is the faster the PLL detects the tone. The volume of the received signal depends on several factors. There is the linear impact of the distance, the frequency difference of the emitted signal and the band pass center frequency and the orientation of the microphone in respect to the sounder [Girod01]. The first factor has not to be concerned; the second can be calibrated whereas the last impact is unknown in most cases.

A third factor of delay comes from computation durations. For example, this means the time that goes by between executing the sounder activation command and getting the current timestamp from the hardware counter. These inaccuracies are fixed and only depend on the program code but should be concerned during program development.

Another error source is the speed of sound, which is dynamic depending on the environment. Factors of speed variation are temperature, humidity, pressure and CO_2 concentration [Cramer93] whereas the main impact comes from the ambient temperature. The influence of pressure grows for high temperatures but is negligible for the targeted operating range. Further, the pressure change in static deployment is very small. The same arguments apply to the CO_2 concentration. The impact of humidity should be compensated especially for higher temperatures. Finally the effect of ambient temperature has to be corrected [Farrokhi05]. The following linear approximation is valid for temperatures from around 0°C up to 30°C:

$$c\,[m/s] = 331.4 \cdot \sqrt{1 + 3.66 \cdot 10^{-3} \cdot \theta\,[°C]} \approx 331.4 + 0.6 \cdot \theta\,[°C] \tag{2.7}$$

Another problem are false detections caused by noise. Due to the use of audible frequencies, many sources of interference exist in an indoor scenario. Many effects are present in a wide spectrum that includes the frequencies of the acoustic transmission. This can cause false detections on the receiver side. For robust range estimation, different measurements have to be made. Outliers' correction can therefore be done by means of statistical methods. In this case, a rolling or weighted average can be used to discard values that differ too much. Another possibility is to calculate the median of the gathered data. A pulse sequence also reduces false detections, because the chance that the noise has the same pattern as the sequence is vanishingly small.

The most challenging error source is based on multi-path effects and NLoS measurements. In the first case, the receiver detects the same pulse more than once. In such a case, the first received signal is the best choice because the direct way is always the shortest. If the line of sight is blocked by obstacles, this technique results in enormous error rates. Without redundant measurements or a additional transmission media such as infrared a robust detection of this negative effect is not possible [Sallai04]. Nevertheless, it is possible to calibrate a receiver system to dynamically detect occurring NLoS measurements by a relationship between the mean result and the standard deviation [Jeong01]. Another error can occur when the mobile node is moving. Special tracking algorithms consider this so called Doppler shift with special filters [Ansari07].

2.3.5 Approaches for Data Filtering and Error Minimization

This last subsection of acoustic ranging covers different solutions for error minimization designed for use on typical sensor nodes. All of the algorithms presented try to compensate some of the earlier mentioned error sources with respect to the available computational power and hardware specifications. More complex solutions, that need central processing or additional DSP hardware, are not mentioned in detail because the goal of this work is a distributed and power efficient ranging solution.

The absolute position estimation by Mads Fogh [Fogh04] is based on Mica2 nodes and uses a TDoA based distance measurement. The signal detection is software based and uses the 10 bit ADC from the micro controller sampling at 20 kHz. The time difference is measured between the reception of a synchronization radio message and the exact detection of the acoustic pulse. The ADC values are fed to a programmed band pass. The time of arrival is detected based on the steepest incline of the filtered signal. This measurement method introduces Gaussian distributed noise with an expected value of null. The according variance is minimized by a Kalman filter [Welch95]. In this case, a next prediction is the sum of the previous value and the according weighted measurement difference. The measurement also produces a nonlinear noise which is minimized by a linear minimum mean squared error estimator (LMMSEE). The algorithm also compensates the influences of air temperature and humidity together with constant system latencies. Results for the accuracy are only available for distances up to 1.5 m and are less then 5 cm for most of the cases.

An improved variant for a range estimator is the extended Kalman filter (EKF) which is used in combination with a special TDoA measurement, developed at Jiaotong University [Hongyang05]. This approach is based on nonlinear hyperbolic equations to calculate the distance and position of the nodes. This fact makes the usage of a linear estimator as the normal Kalman filter inadequate for error minimization. For such problem statements the EKF was designed [Julier97] and achieves good results even when the input data has a great variance.

The latter approaches detected only one arriving acoustic pulse per measurement result. A pulse sequence can achieve better results instead, as already mentioned in Section 2.3.3. A system based on Mica2 nodes and a ToF based ranging method comes from Vanderbilt University [Sallai04] and also utilizes the built-in ADC at a sampling rate of 18 kHz. To increase the SNR a pulse sequence with 16 chirps is used for a gain of 12 dB following Formula (2.6). All chirps of the sequence have the same length but variable intervals in between. Consecutive sequences have the same pattern. The raw data from the ADC is fed into a band pass finite impulse response (FIR) filter optimized for computation on a 8 bit micro controller. The filter has 35 taps with coefficients in the $[-4, 4]$ range and needs only 36 arithmetic operations. Pulse detection is made by finding the maximum of a rolling average on the filtered data stream. The pattern of the sequence is not used to filter out false positives but only to make the peak detections more robust. The range computation compensates the static hardware delays with data from offline tests and considers the current ambient temperature for speed of sound calculation. The resulting mean squared error is under 10 cm for ranges up to nine meters. The resulting error histogram (Figure 2.9) also depicts the Gaussian error behavior of measurement noise.

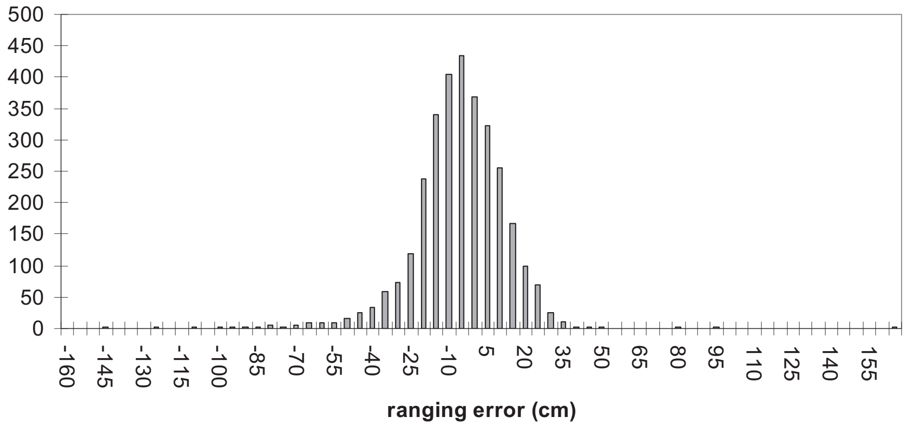


Figure 2.9: The error histogram of almost every acoustic based ranging project shows Gaussian distributed behavior [Sallai04].

A system especially designed for outdoor environments is postulated by Kwon. It basically uses the Mica2 platform with a the normal sensor board but the board is equipped with a louder sounder for increased ranging distance [Kwon05]. The ranging is based on a TDoA measurement between radio and acoustics with an additional clock synchronization. For improving SNR, a pulse sequence is used for acoustic transmission. The approach does not rely on signal processing, because the built in sound detector is used together with some sophisticated statistical methods instead. Due to the large amount of false detections of the tone detector, a special detection method based on number of detections per time interval specifies the detection time. These times are then buffered and compared to the a priori known pulse pattern. The afterwards calculated range estimates are checked for consistency in two ways. First bidirectional between two and afterwards for the triangle inequality between three different nodes.

A very similar approach was already stated by Whitehouse but this project concentrates more on the easy and robust calibration of large WSNs [Whitehouse02]. The method uses acoustic ranging and heavily concentrates on calibrating the dynamic errors introduced by the environment and especially from unit-to-unit variations. Many of the error sources described in Section 2.3.4 come from these variations and are complicated to avoid for large amounts of nodes because the deployment time would be enormous. The drawback is the complexity of the calibration algorithm. It is basically a linear equation system with four times as much unknown parameters as the number of nodes in the network. The calculation can only by accomplished by an external computer.

2.3.6 Summary and Comparison

The last pages presented different techniques of acoustic ranging in wireless sensor networks based on acoustics. The first decision, which has to be made, is the frequency range because ultrasonic and audible frequencies have different behaviors. First, the two domains are presented separately with their benefits and drawbacks especially regarding hardware matters. After that, the challenge of generating a signal robust to noise and false detections is described. The detection of such audible signals is a very cunning task, which needs the most know-how of a ranging algorithm. This comes from the different sources of error described in this section. There are delays caused from hardware and software, the variation of the speed of sound, ambient noise, multi path propagation and NLoS errors. Afterwards the section described solutions stated by different researchers to prevent or minimize these sources of errors. For example with digital signal processing, special signal encodings, signal detection filters, statistical methods or consistency checks of the gathered results.

Table 2.2 summarizes different projects and presents some of the key parameters. All algorithms were described in detail within this section. The listed values for the accuracy are for the maximum range and typically decrease for closed measurements. Most of the experiments were made under laboratory conditions with calibrated hardware. One exception is the project of Whitehouse, which specially targets on the elimination of calibration errors but introduces a higher error. In general it can be said that typical algorithms produce errors greater than 5 cm for close distance measurements and have an upper limit of 5 % error rate for high distances.

Researcher	**transmission media**	**ranging method**	**maximum range**	**accuracy**	**error minimization**
Fogh	audible	TDoA	2 m	5 cm	Kalman
Sallei	audible	ToA	9 m	10 cm	band pass
Hongyang	audible	TDoA	simulation	1 %	EKF
Kushwaha	audible	TDoA	30 m	30 cm	FIR filter
Farrokhi	audible	ToA	2 m	5 cm	CCF
Girod	audible	ToA	6 m	unknown	CCF
Ansari	ultrasonic	ToF	2 m	3 cm	Bayesian filter
Whitehouse	audible	ToF	3 m	5 %	calibration
Kwon	audible	TDoA	20 m	1 %	statistics

Table 2.2: Accuracy of Different Acoustic Ranging Projects

2.4 Positioning Approaches

There are many positioning algorithms that can be found in literature. This section will try to categorize them and present some of the approaches. Even when looking only for WSN implementations, there are still many solutions. Positioning in WSNs means to specify the current position for each node of the network. In most cases, this position information is defined within a coordinate system, but needs not to be. The result could also be the name of a known place or the proximity to some point. Depending on the precision of a positioning system, it can be divided into coarse and fine-grained approaches. In terms of a WSN a coarse position has enough precision to distinguish every single node from all others. If additional information can be retrieved, the position is fine grained.

The different approaches are categorized by the need of ranging data, if the algorithm is processed centrally or distributed on the nodes, the type of needed infrastructure and reference nodes and finally if there is the possibility to get positions via a multi hop connection. After this introduction, some implementations are described and compared.

2.4.1 Characteristics of Positioning Methods

Central vs. Distributed Computation

There are several reasons for computing a positioning algorithm centrally at a base station: fixed power supply and higher computational capabilities. Typical hardware platforms for sensor nodes are equipped with low power but also low performance micro controllers. If a positioning algorithm needs complex mathematics the program execution on these controllers would take too much time and energy. In this case it is better to transmit the data to a base station and send the results back to the mote. Although it has to be balanced if the cost of sending radio messages is less than the local calculation.

When the number of nodes in a network raises the centralized approach scales badly for the communication and computation overhead. For huge networks, every positioning approach should be distributed to avoid this problem. Typically, a distributed approach needs more complex routing protocols because every node eventually has to transmit data to all others. Examples for a centralized approach are the convex position estimation [Doherty01], mobile acoustic beacon localization [Kushwaha05], the RSSI based indoor localization system RADAR [Bahl00] or the directionality based location discovery scheme [Nasipuri02]. On the other hand, many distributed implementations exist. The ad-hoc positioning system (APS) framework [Niculescu04] is completely distributed, as the approximate point in triangle (APIT) approach [He03]. Some algorithms furthermore support both computation modes.

Range Free vs. Range Based Positioning

As already said, one main characteristic of a positioning algorithm is the need of ranging information. This means that there has to be a possibility to calculate a distance between two nodes resulting in a definite length value. The opposite would be an estimation of the distance of another node based on the radio communication range or try to position nodes based on proximity or their neighborhood graph.

The advantage of range-based approaches is their higher precision, at the cost of more hardware effort. Most range-based approaches achieve fine-grained positioning information. Many range free algorithms rely on either a very complicated computation or the presence of a complex infrastructure. To cover with the higher computational effort a central approach would suit best, as it is the case for the RADAR system. An often used parameter to replace definite distances is the hop count. This approach suits especially for coarse-grained positioning where the hop count to different reference nodes together with the typical radio range is used to calculate the position. Proximity based approaches are another coarse-grained range free variant. The presence or absence of different beacon signals is basically used to divide the area and specify the current position of the node [Bulusu00]. APIT can be seen as an improved proximity based method. Some positioning algorithms can operate with range based and connectivity based information, for example the APS framework. Another range free method is the proximity matrix method [Agha03].

Infrastructure Based and Ad-Hoc Positioning

To classify the method of the overall mechanism of positioning, literature describes several terminologies. The first is the need or absence of infrastructure. If a network relies on some nodes that have a fixed position and know the according information from the beginning, the positioning algorithm is infrastructure based. A simple base station normally accounts not as infrastructure because it implements only a data sink and does not assists the positioning. If a node of the infrastructure propagates the own position information to other nodes, it is called anchor node. Anchor nodes normally have a high trustiness and their amount and density significantly affects the precision and accuracy of the system. A special case of an anchor node is a reference node. It also knows the own position, but can be mobile during operation. The mobile acoustic beacon localization system [Kushwaha05] is an example, which needs only one reference node to position all unknown ones. A drawback is the higher complexity of a reference node, because it must update its position without the help of the network. In most cases this is realized with a high precision GPS system. The term of reference nodes is also valid for outdoor systems that rely on some nodes which are equipped with GPS modules.

Another terminology is the beacon node. Sometimes it is equally used to reference or anchor nodes but this is not true in general. A beacon node propagates the own position to all neighbors and accepts ranging in between. A beacon node is a reference during one or more ranging processes but not for the overall positioning system. With this definition, reference and anchor nodes behave like beacon nodes, but not vice versa. So it can be said that almost every positioning system is based on beacon nodes [Battelli07]. The use of beacon or reference nodes not necessarily makes the positioning approach infrastructure based. For example, an outdoor network relies only on one reference node to convert the calculated positions from a relative coordinate system into the globally valid world geodetic system 1984 (WGS84).

The counterpart of an infrastructure-based algorithm is ad-hoc positioning which comes from the MANET definition of wireless sensor networks and therefore disclaims the use of infrastructure. The development of ad-hoc algorithms is definitely more complex than relying on infrastructure but should be the long term goal for outdoor scenarios. Ad-hoc positioning is often equally referenced with multi-hop positioning. This technique allows calculating distances even when there is no direct communication between both nodes. Some ad-hoc positioning methods rely on this technique, for example APS. This leads to the point, that an ad-hoc system is not the opposite of an infrastructure-based approach. Theoretical solutions for ad-hoc positioning without infrastructure exist [Yu06] and also some implementations for Mica2 nodes [Priyantha03]. Some of these algorithms do not rely on multi-hop distance calculation but rather concentrate on the neighborhood. One approach is that every node finds a position with respect to the neighborhood. After this initial step the information of every node is shared and a common coordinate system is formed [Bachrach04, Kwon05].

Most positioning systems that rely on multilateration can be classified to use direct or single-hop position calculation in opposite to multi-hop distance calculation. Such a system basically always needs the support of an infrastructure. The number of needed anchor nodes or the density of these is a good quality factor of such a system, and should be as low as possible as long as the precision is acceptable. Yet some localization systems do not fit perfectly into this terminology. The Laplace approximated simultaneous localization and tracking (LaSLAT) system [Taylor06] for example has the goal to track mobile beacon nodes. To provide this feature a high number of nodes are deployed, which represent the fixed infrastructure. None of this nodes know their own position but all work together to transform the measured distances to the unknown beacon node into a valid position.

2.4.2 Approximate Point in Triangle

The first positioning system presented, is postulated as range free approach. The calculation is done with an area based comparison method [Agha03]. The goal of area-based methods is to determine the nodes position by narrowing an area down to the most possible location of the node. To make this happen, an imprecise calculation method is used more often to narrow the real node's position. The basic calculation can deliver very coarse grain results, for example reduce the possible position by intersecting the whole playground in half. Such methods are often used for acoustic event or source localization and presume a very high node density in the network [You06, Guha05]. In most cases there have to be more than 10 nodes inside the radio or acoustic communication range.

The APIT positioning system is based on the perfect point in triangle (PIT) test [He03] and uses neighborhood information and coarse RSSI values to find the nodes position. Additionally there has to be a high anchor node density. Taking three anchor nodes, their position spans a known triangle in a 2D coordinate system. Additionally there is an unknown node in this system which can measure the distances to all three anchors. The perfect PIT test can give information whether the node resides in or outside the triangle. If there is a direction that the node can be moved where all three distances to the anchors will increase, the node is outside the triangle. The drawback is that the nodes movement cannot be realized in a WSN.

If the node density of the network is high enough, a typical spanned triangle would include many unknown nodes. So these nodes can be taken to simulate the movement in either direction. So the approximate PIT test states as follows; if a node can find at minimum one direct neighbor that is outside the triangle, it assumes that the same is valid for himself. The APIT algorithm itself divides into four parts. First, the anchor nodes send radio beacons with their coordinates. This information is stored together with the appropriate RSSI value and shared among the neighborhood. Now each node has the information to perform the PIT tests, one for each combination of three anchor nodes. The results of each test is aggregated into a result grid describing the playground in a discrete manner. If the test was successful, every entry of the grid inside the triangle is incremented or decremented otherwise. After all tests, the position can be calculated by taking the center of gravity from the grid.

A great advantage to other RSSI based ranging and positioning systems is that this approach is very resistant to most radio interferences, because the values are only compared and not taken directly into account. Additionally no external hardware is used and the communication overhead is low in comparison to other positioning approaches. Nevertheless, the main disadvantage is the high anchor node density needed. The resulting error of this approach mainly depends on the neighborhood density and number of received beacons. When receiving 10 beacon signals and having 8 direct neighbors, the precision is half the own radio transmission range.

2.4.3 Multilateration

The most intuitive method for calculating a position based on different ranging estimates is multilateration. The result is calculated by finding the point where all distance measurements intersect. Multilateration is the most common technique for range based positioning approaches.

When looking at a two-dimensional coordinate system, a distance value and the position of a reference node define a circle on which the unknown node will reside. Taking another ranging, the estimate can be narrowed to two points, the intersections of the two circles. If one of these points cannot be disqualified by constraints, which is normally the case, there is the need of another range to a known reference station. If translated to three dimensions, a single ranging specifies a sphere. In this case, four independent ranging results are necessary to find a valid solution. Solving the two dimensional variant leads to a system of three equations (Equation (2.8)) with two unknowns (u_x, u_y). This can be solved to one solution, but only theoretically. Introduced measurement errors give a residuum around the intersection of all three equations, resulting in a infinite set of solutions [Boukerche07]. So this technique cannot be used with real measurements.

$$d_i^2 = (u_x - x_i)^2 + (u_y - y_i)^2 \quad i = 1 \ldots 3 \tag{2.8}$$

To make the positioning calculation more robust against ranging errors normally more reference nodes are taken into account. This finally leads to the term of multilateration. With this technique a high number of distances is transformed into an over determined equation system. As the solution of such a system cannot be solved directly, the problem of a resulting residuum has not that much impact anymore.

The method of linear least squares (LS) is the standard mathematic solution for solving over determined equation systems. First the equation system earlier mentioned has to be converted with a linearizing tool [Murphy07]. The result are $i-1$ linear equations with two unknowns. This system can now be solved by the LS method by applying the L2-norm

[Reichenbach06]. This means the solution should minimize the squared error as stated in Formula (2.9):

$$\underset{x \in \Re^n}{Minimize} \; \|Ax - b\|^2 \;\rightarrow\; (A^T A) \cdot x = A^T b \tag{2.9}$$

There are several solutions to this problem; three of them can solve the system directly. The decision depends on the form of Matrix $A^T A$. If it is well populated and non-singular the system can be solved by matrix inversion. If the matrix is singular or badly populated, the QR-decomposition can be used. If this technique also fails, the singular value decomposition (SVD) method always leads to an result [Murphy07]. The methods strongly differ in complexity and therefore computational effort. The normal equation solution is the simplest while the SVD algorithm is the most complex solution.

The calculation described so far is sometimes named direct or atomic multilateration process. That is because the input parameters cover only distances to direct neighbors of the unknown node. Additionally it is only possible to calculate a single position that depends from three or more surrounding nodes. Nevertheless, it is a simple method and therefore often used in fine grained positioning systems [Fogh04]. A prerequisite to this approach is that every node has contact to at least three reference nodes. That is not always the case when an ad-hoc network is taken into account. There may be areas where fewer references are available or even none is present. This fact makes direct multilateration unusable for particular parts of the network. To cover with this problem iterative multilateration can help [Savvides01]. First, every unknown node counts the beacons it can receive. The nodes with the highest number perform an atomic multilateration to calculate their own position. Then these nodes turn themselves to beacon nodes. After this, the remaining unknown nodes rescan the neighborhood including the newly found reference nodes and the process starts again. After a while, all nodes should be able to start a calculation. Still there is the problem or error propagation. When looking at convex network topologies, which has no outlying areas, this error does not grow very fast. Simulations proved that a reference node density of 10% results in positioning errors below 20 cm, if the ranging data was disturbed with 2 cm of white Gaussian noise.

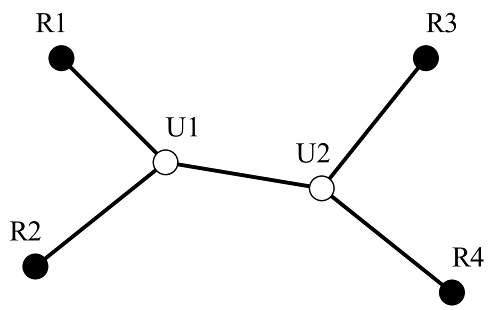


Figure 2.10: Problematic topology for direct or iterative multilateration. It can be solved by knowledge exchange between unknown nodes.

A further improvement to this algorithm is the n-hop multilateration [Savvides02].

Even in the iterative form, some areas of the network will be unable to position, because there are too few beacons. Therefore, it would be a benefit, that in such cases a node could find a position with the help of other unknown nodes and their distances to further references. A simple topology for such a problem is depicted in Figure 2.10. Both unknown nodes cannot calculate a position because both only hear two beacon signals, but when working together this can be managed. When all inter-node distances are known, an algorithm could solve such a multilateration problem in an iterative fashion. However, this needs high computational effort and can hardly be managed by a single node. So the number of neighbors and their distances to reference nodes has to be limited, by finding a trade off between computation time and success rate of the positioning algorithm.

Especially for iterative processes, the robust positioning algorithm [Savarese02] introduces a so-called confidence value. This value indirectly represents the estimated error of the position calculation. In this implementation, the ad-hoc network tries to find the positions iteratively after a range free process for a first guess. The anchor nodes always have the highest confidence, all unknown nodes start at a low level. After accomplishing a multilateration, the node updates the confidence as an average of all beacon nodes, which were used for the calculation. As beacons with high confidence values are always preferred, this technique should prevent a high error propagation.

As the process for solving the LS problem is not simple and becomes even more complicated in 3D or by concerning neighborhood information, an approach close to area based positioning can give another solution. This is called the bounding-box method [Simic01] and initially was developed to work with range free and only connection based systems. The idea is to idealize the radio communication range to a square area around the node. To calculate the final position, first all square areas of the beacon nodes are merged additively. After this process, there should be a remaining residuum. The center of gravity from this remaining area is the resulting position. This approach definitely introduces more error than multilateration but needs far less computational effort. To cover with range based problems, this approach can be adopted be defining two different sized squares, one for the minimum distance and one for the maximum based on result and confidence values of the ranging.

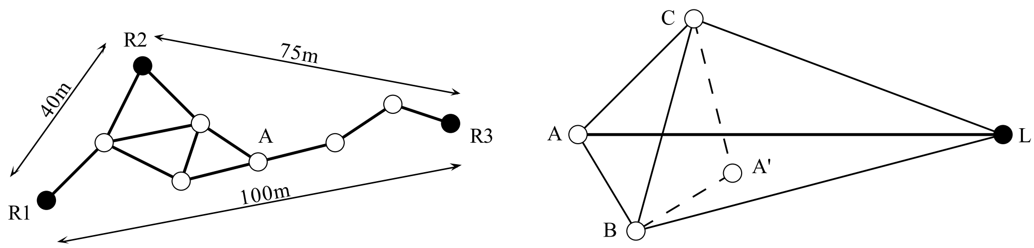
2.4.4 Ad-Hoc Positioning System

The ad-hoc positioning system (APS) is a framework developed by Rutgers University. The basic idea is to incorporate multilateration and distance vector (DV) routing to support multi-hop positioning in ad-hoc networks [Niculescu03b]. The system consists of two main parts, the APS algorithm itself and the underlying distance estimation. To cover with the already mentioned problem of no direct connection to more than three reference nodes, the distance to these nodes has to be estimated and the results are propagated through the network. This is based on the DV routing protocol which handles data exchange and tries to reduce the number of messages in the network. After having at least three distances to reference nodes, a atomic multilateration process is started as described in Section 2.4.3. Already positioned nodes can assist neighboring nodes during the positioning process, as it is done by iterative multilateration.

The precision of the APS algorithm depends on the used distance estimation. The framework includes several of these. The DV-hop method is the most simplest because it is a range-free approach. The basic idea is to count the hops between sender and receiver and conclude to a distance with that information. Reference nodes also do a hop-count in between them. After that, a correction term can be calculated and propagated to the neighboring nodes. The correction is calculated be dividing the sum of distances to other reference nodes through the sum of their corresponding hop counts. This gives a good estimate of the average hop size in meters. If a unknown node has to calculate its distances to other references, it takes the correction term of the nearest reference node. This progress is also shown in Figure 2.11(a) and gives the following result for the distance of one hop which node A uses for range estimation:

$$c_i = \frac{\sum d_{ij}}{\sum h_j} \quad \forall_{j \neq i} \qquad c_2 = \frac{40 + 75}{2 + 5} = 16.42\ m \tag{2.10}$$

Since DV-hop is range free, the error introduced for positioning is very high. Simulations result in errors of up to 50% of the radio range and increase to over 100% if deployed to hardware [Stoleru04]. However, the technique does not rely on additional hardware and is very simple. An improvement in distance estimation is the DV-distance technique. The basic principle is identical to the before mentioned, but it relies on accumulated distance measurements instead of a hop count. Finding a correction value is also necessary. The resulting positions have higher precision, but additionally rely on the distance measurement error.



(a) DV-hop relies on hop counting and correction factors calculated between reference nodes.

(b) DV-Euclidean calculates distances arithmetically with distance values of two neighbors.

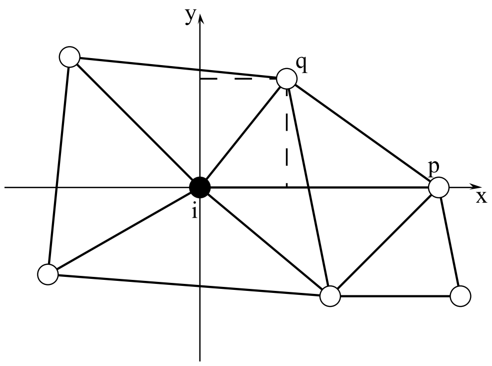
Figure 2.11: Basic APS range estimation techniques.

The third possible range estimator is DV-Euclidean. The basic idea is to find the distance to a reference node when a direct ranging is not possible. This is assisted by at least two neighbors with direct connection to the reference and redundant distance measurements. As shown in Figure 2.11(b), node A tries to calculate the distance to reference node L. At beginning all distances have to be gathered. With that information, the unknown range can be calculated. The ambiguity of the solution (A') can be solved by considering a third neighbor. If no obstacles are present, the nearer position cannot be valid, because this position would make it possible to measure the distance to the reference directly.

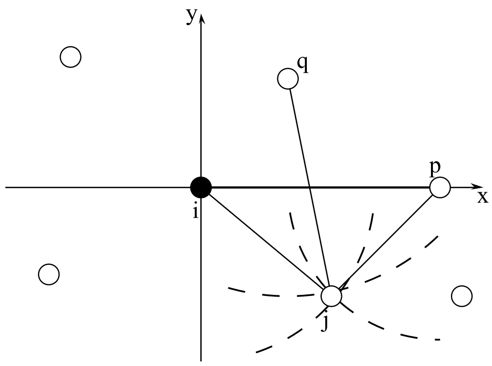
There is another range estimator, named DV-coordinate where every node builds an own coordinate system based on the whole neighborhood information. This technique is the most complex of the already mentioned, and is described in detail in the next section. Additionally the framework was extended to include AoA information [Niculescu03a]. Depending on the type of AoA information (see Section 2.1.2) there exist two more distance estimators, DV-bearing and DV-radial. The APS framework is also used slightly modified in other projects [Whitehouse02].

2.4.5 Distributed Least Squares Scaling

Another possibility to calculate node positions in ad-hoc networks with distributed computation is by introducing locally valid coordinate systems and corresponding transformations between them. The basic idea is constructing a local coordinate system that is only based on ranging estimates [Capkun01], and divides into three parts. After collecting information about the neighborhood, the axis of the new coordinate system are defined and finally all other nodes are integrated into this system. So the first step is finding all one hop neighbors and execute a ranging process. This information is then sent to all the direct neighbors. At the end of this step, every node has enough information about all nodes within a two-hop distance to continue.



(a) After gathering the neighborhood information, define the axis orientation for the local coordinate system.



(b) Finally calculate the according position for the remaining nodes with multilateration.

Figure 2.12: Building a local coordinate system needs ranging information from all two hop neighbors and divides into three steps.

The next step is to define the axis as depicted in Figure 2.12(a). For this, two randomly linearly independent nodes (p, q) are taken. One of these nodes is used to define the x-axis directly whereas the other is used to span the y-axis. So the first two node positions were found. The final step is to integrate the other nodes into this coordinate system. For this reason, every node is positioned with a multilateration algorithm from at least three reference nodes. Figure 2.12(b) shows the process that is finally done by a LS algorithm as described in Section 2.4.3.

As every node builds up an own system, there has to be some kind of transformation between them. To merge two local coordinate systems different factors have to be considered. First is the orientation of the axis which includes a rotation and an additional mirroring is necessary. Therefore, a connection between two nodes, which is present in both systems, has to be aligned. As both nodes claim to be at the origin, one system has to be translated based on a single shared node. High ranging errors and therefore wide residues in the multilateration process make it necessary to scale one of the systems to fit into the other.

The already mentioned DV-coordinate distance estimator of the APS framework uses this technique to build up local coordinate systems. So a range is estimated by transforming the beacon signal coming from a reference node at each hop to the local system of the next hop until it reaches the destination. After receiving three such beacons, their position is uniformly transformed into the own coordinate system and the positioning algorithm can be executed.

The distributed least squares scaling (LSS) algorithm is also based on such local systems [Kwon05]. This approach first calculates such a local coordinate system for every node in the network. After that step, the transformations have to be found. As the computational effort for the earlier mentioned transformation process is very high, a more sophisticated variant was found. A drawback of this algorithm is the lower precision. The transformation builds on aligning all positions of those nodes that share both coordinate systems. To ease this process only one point is used for the calculation. This point is given by the center of mass from all shared points in both systems. Furthermore, the process of scaling the systems is omitted. After a leader election, this node claims to be the origin of the global coordinate system. The final step is to align each node to the new coordinate system. After this step, every node knows its own position based on the now globally valid system.

Different to DV-coordinate the distributed LSS algorithm starts with a predefined amount of the possible inter-node ranging estimates. This has a major impact on the overall precision of the positioning. Simulations have shown that giving a third of the possible ranging estimates leads to a high number of completely false positioned nodes. When providing the whole ranging data the error is only 5% of the average inter node distance which is fairly good for distributed multi-hop positioning systems.

2.4.6 Summary

As presented in the last section there is a wide variety of positioning systems in literature. Each of them has its own advantages based on additional hardware effort, power consumption, failure robustness, speed and some other parameters. This fact shows that yet no perfect solution exists for the problem statement of positioning in WSNs. Some of the constraints were already defined in Chapter 1. The main goals are power awareness, low price and high robustness to failures. Indoor systems additionally claim for high accuracy and precision of the calculations. The requirement of complete autonomy from infrastructure is not so crucial instead.

The term of robust positioning has become popular in the last years and tries to position as much nodes of a network as possible without neglecting the precision of the results. To reach this goal, no new algorithms are invented but the combination of existing systems should lead to better results [Stoleru07]. The main know how lies in the strategy to decide which algorithm should be used for a specific topology problem. There is the possibility to run different approaches redundantly and compare the results, or to have a backup strategy if something goes wrong. As such a system would be placed on top of different positioning approaches, the according projects were not yet mentioned in the literature part.

Not all approaches presented in this section fulfill the specifications for a fine-grained indoor positioning system, but each of them includes some ideas that can be incorporated into a fine-grained indoor localization system. One outcome of this research is the fact that multilateration is an essential part of each high precision positioning system. Most of the range based multi-hop algorithms try to solve the problem with the help of multilateration. If such algorithms have a distributed calculation method, the workload of each node rises a lot but remains constant for the network size.

The importance of infrastructure is still not negligible especially for fine-grained positions. Almost every ad-hoc based system is strongly dependent from the density of reference or even anchor nodes in the network. For outdoor systems this is mostly realized with GPS equipped nodes. These nodes need more energy and cost more than the other nodes. Therefore, it could be possible to introduce another platform for these nodes, either with fixed power supply or higher storage capacity. The benefit of indoor environments is the easy access to a fixed power grid. This makes it possible to reduce the additional costs for a fixed infrastructure. Yet the deployment costs remain and they should be minimized to provide higher flexibility for expanding the operational area.

Literature provides also completely infrastructure less positioning systems, for example the anchor free localization (AFL) system [Priyantha03]. These systems need high computational effort but provide only coarse-grained positioning. Although the deployment costs decrease, the precision is only suitable for wide area outdoor applications. The problem of low precision also is valid for all the range free positioning approaches. An error of several meters is unacceptable for indoor environments. Some ideas can still be borrowed to assist a range based approach, for example the correction factor calculation from the DV-hop method or some ideas from area based methodologies like APIT. Another very promising technique is to introduce a refinement phase [Langendoen05]. This idea is a special case of a iterative positioning approach, which performs the same algorithm several times. A refinement phase by contrast is another algorithm that increases the positions precision, which was found by an initial positioning process, in an iterative way.

So a possible solution for indoor scenarios will utilize multilateration together with a fixed infrastructure. In most cases, there is no need of multi-hop position estimation. To increase the number of nodes that can be positioned correctly, a method like iterative or collaborative multilateration has to be considered. If an unknown node has contact to fewer reference nodes than needed, this strategy can be a backup algorithm. To ensure power awareness, there has to be found a threshold between higher precision and lower power consumption.

2.5 Power Measurement and Estimation

A very important task is to estimate the power consumption of a WSN program. The knowledge of this information can be used for algorithm improvements, because a low power dissipation is a main design goal for WSNs. On the other hand, this value can be used to calculate the estimated lifetime or a duty cycle, which has to be introduced to ensure a certain time of operation. To estimate the power consumption computationally, an energy model of the targeted hardware has to be built. Most power aware simulators have detailed information about the power states of each hardware block on a sensor node. For the estimation of a program, the simulator accumulates all power states with respect to the executed code.

So the first task is building a power model of the node. To gather all the needed information, all possible power states should be executed and measured precisely. A sample of a current profile for a Mica2 node utilizing the radio is depicted in Figure 2.13. By averaging the power consumption during each operational state, the according power value can be calculated and divided into the single components. This method was utilized by the PowerTOSSIM project to build the power model of the Mica family [Shnayder04]. The project builds upon the TinyOS simulator and additionally calculates energy requirements for each simulated node.

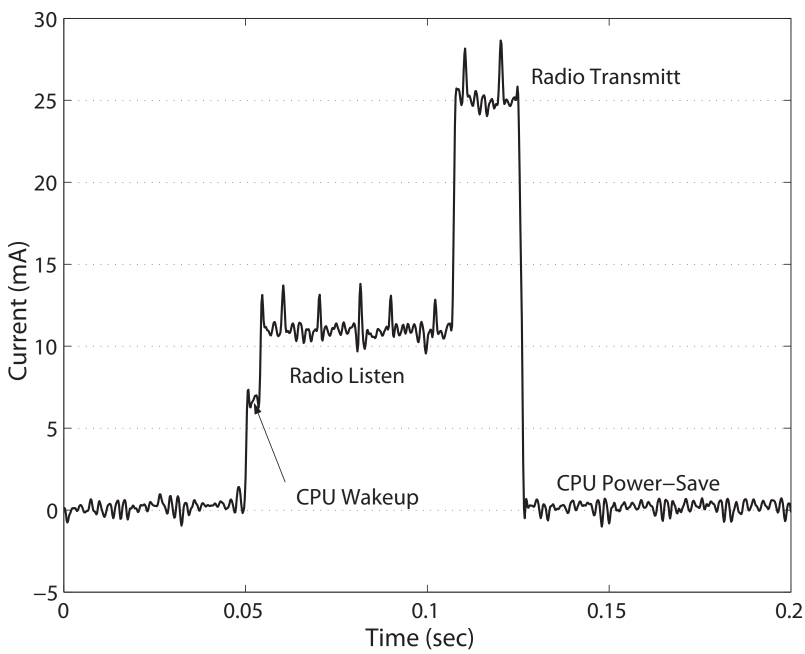


Figure 2.13: Profile showing the current consumption of a Mica2 sensor node when using the radio module with a battery voltage of 3 V [Shnayder04].

A more general approach which tries to bridge the gap between software simulation with power profiling and the field of energy harvesting systems is the energy harvesting modeling in wireless sensor network simulation (EMWiSS) framework [Glatz10b]. The field of energy harvesting modeling introduces adaptive power cycles for sensor nodes that depend on the available energy in a small storage device that is loaded by energy harvesting modules. The nodes application is written as state machine in a high-level language and then simulated based on discrete events to reduce computational effort. Between two events, a specified power consumption is calculated based on the power states of all participating components of the Mica2 hardware model. The need of a power model is essential to achieve a precise estimation of the power consumption.

As already discussed, the exact measurement of power consumption is an essential task for each simulator with power estimation. The direct measurement can further be used to test a program directly on the target hardware before a deployment or to evaluate more complex power profiles. For example, the energy requirements for a more complex algorithm can be measured directly. An example for a power measurement test bed is explained in [Glatz10a]. The device under test (DUT) can be either a constantly powered Mica2 node or a complete energy harvesting system (EHS) that activates the mote. By making different profiles a power model of the Mica2 was found and some values are listed in Table 2.3

For some applications it is also feasible to observe the power consumption at runtime [Trathnigg08]. These platforms observe the power consumption during the whole lifetime of the mote. The measured power is accumulated to monitor the energy consumed by the mote itself. The necessary circuit has to be very power efficient, because the power needed for monitoring is wasted and cannot be utilized by the mote. The benefit of such systems is the knowledge of the exact lifetime of the node. This information can for example be utilized within applications to control the packet flow of routing algorithms.

Component	**State**	**Power**
CPU	active	23 mW
	idle	10 mW
	standby	0.75 mW
EEPROM	write	55 mW
	read	19 mW
Radio	idle	30 mW
	transmit (-18 dBm)	26 mW
	transmit (0 dBm)	51 mW
	receive	29 mW
Peripherals	LEDs	7 mW
	sensor board	2 mW

Table 2.3: Primary Parameters of the Mica2 Power Model sending at the 900 MHz band which has higher power consumption. The power values differ in literature, these are taken from the AEON project [Landsiedel05].

Chapter 3

Concept

The following chapter defines the concept of this thesis. There are several design principles, which guide through the following chapters. A positioning system is rarely used as user application on a sensor node. It only assists these applications by providing helpful information, which is used to accomplish a certain task. Nevertheless, some classifications have to be made to the network to fulfill the requirements for this concept. These can be derived form design space exploration as described by [Römer04]. The deployment phase of the network is different, depending on the type of positioning algorithm. When only using multilateration the reference nodes have to be deployed manually, specifying the current nodes position. The main number of nodes can be deployed randomly. The type of mobility is another classification. The concept will cover networks consisting of partly, occasional and passive moving nodes. This means that most of the nodes can be mobile as long as they are static during ranging measurement. The type of communication modality must be radio and acoustic to work with this implementation. Theoretically, the network can be heterogeneous as long as the communication between nodes can be accomplished. However, this implementation will only cover nodes of the Mica family. There are no claims to the network topology because the ranging process is based on single-hop communication. The same applies to requirements regarding network connectivity, lifetime and network size.

As already mentioned in chapters 1 and 2 the system operates in indoor environments. This can be storehouses, production facilities or even office buildings. To reduce the needed infrastructure large rooms are beneficial because the acoustic media is absorbed and reflected by walls and ceilings. Theoretically, every mobile node must have a direct connection to four reference nodes to calculate a valid three-dimensional position. To decrease failure impact this number should be higher. So the relation of unknown nodes to the room size specifies if this system is appropriate. The targeted accuracy of the ranging process should be less than 10 cm to provide reasonable position information in indoor environments. With this error, it is feasible to get enough position information two distinguish two nodes separated by half a meter.

3.1 Basic Design Principles

The two main design principles are modularity and adaptive fidelity [Hightower01].

The **modularity** of the whole system is the first main aspect, primarily the differentiation of positioning from the ranging process. This makes it possible to develop both parts separately or exchange one of them. For further research the current multilateration, based positioning approach can be evolved to an ad-hoc method. Alternatively, the ranging process can be exchanged to improve the support of highly heterogeneous networks with different types of ranging strategies. Another advantage is introduced by a layered design of the software which makes the concept and implementation phase easier.

For this reason, the positioning is divided into several layers. The first handles the incoming hardware events and makes the signal detection. The next layer combines the times of arrival from both signals to calculate a TDoA value. The third layer handles successive measurements and filters the gathered data to a final estimation. The next layer implements a protocol to start and stop such a measurement between two ore more involved nodes. On top of that, the distance model calculates the final result. The fifth layer is the first of the positioning algorithm. It has to detect available reference nodes, stores their information and manages the ranging results. The next layer starts the ranging processes and implements the multilateration algorithm, which provides a position estimate as result. The upper most layer implements the positioning services, which are provided to the main application.

The second main goal of this thesis is **adaptive fidelity**. „A location system with this ability can adjust its precision in response to dynamic situations such as partial failures or directives to conserve battery power.“ [Hightower01] The first concern can be considered active or passive. The active approach would be to increase the effort in power and time for a ranging process in the case of too high error rates. This can be made by a different signal type, higher number of measurements or a more complex data-filtering algorithm. In case of a passive approach, the problem has only to be detected and signaled to the application. Another concern is to make a position estimation in consideration to the available power. This consideration could be done automatically or on demand. Manual declaration can be done from the application by specifying a level of precision when starting a positioning process. This level is considered when defining the amount of redundant ranging measurements to detect a possible NLoS ranging or when initializing the parameters of the ranging process. The parameters should not exploit all available energy because eliminating errors can need more power than expected. An automatic power adoption could be made when using a battery monitor or even utilizing an energy harvesting system [Meyer07] to get knowledge of the available energy. Some state of the art platforms have integrated such a feature already.

The main considerations for developing WSN applications still apply for this thesis. This is mainly power awareness and simple design. As modularity leads to a simple design in most cases, a third design principle is power efficient programming. This can be done by optimizing calculations and algorithms. Floating-point operations and 32 bit multiplications should be minimized because many microprocessors in use only have an 8 bit architecture.

3.2 Network Topology for Positioning

As already mentioned the implementation deals with an infrastructure based positioning approach. This makes it necessary to divide the nodes of the network into two basic classes. A small amount of fixed reference nodes, which know their own position, represents the infrastructure. The higher number of mobile nodes tries to find a position with the help of the infrastructure. After successfully finding its position the mobile node can itself assist others to find their position. Despite these different starting modes, all nodes of a network should run the same positioning service. Therefore, the goal is to integrate all three following roles into one application:

Reference node is a node of the network with manually configured position information. This information can be given by a local reference database defined in an offline phase before deployment. Another way is to transmit this data during operation from a base station. A reference node does not need to make a positioning approach at any time but can initiate a ranging to other reference nodes to check current parameters of the distance model or calibrate dynamic parameters.

Unknown node is the term for every mobile node before initiating a positioning process. If this is done the node switches to the assistant state. To make this possible an application has to initiate a positioning. The result contains an additional trustability value which is important for the future operation.

Assistant node is an already positioned node in the network. At this stage, it can assist other unknown nodes by acting like a reference node. So they accept ranging requests from these. The already mentioned trustability value prevents too high error propagation through the network.

The use of assistant nodes can solve some problems especially for indoor environments. Due to obstacles in a room it is possible that many NLoS rangings occur. This would make it impossible to get a position in some areas of the room. In this case, the node can utilize additional assistant nodes to calculate a position. The resulting precision decreases, but can be made predictable by merging the trustability of each assistant node.

3.3 Ranging Method

The following section describes the concept of the ranging process. There are always a minimum of two nodes involved into a ranging. One node which sends the radio and acoustic signals, here referred to as sender. On the other side, one or more nodes, called receivers, listen to the signals and try to calculate the distance. The most know how and computational effort resides on the receiver side but this approach makes the process of ranging scalable with the network size. There are typically few anchor nodes and many more unknown nodes which have to find their position. Figure 3.1 shows an overview of the ranging concept. The boxes represent the main working parts and the ellipsoids represent the hardware involved. The graphic includes the parts for sender and receiver because it should be able to run both behaviors on the same mote as well.

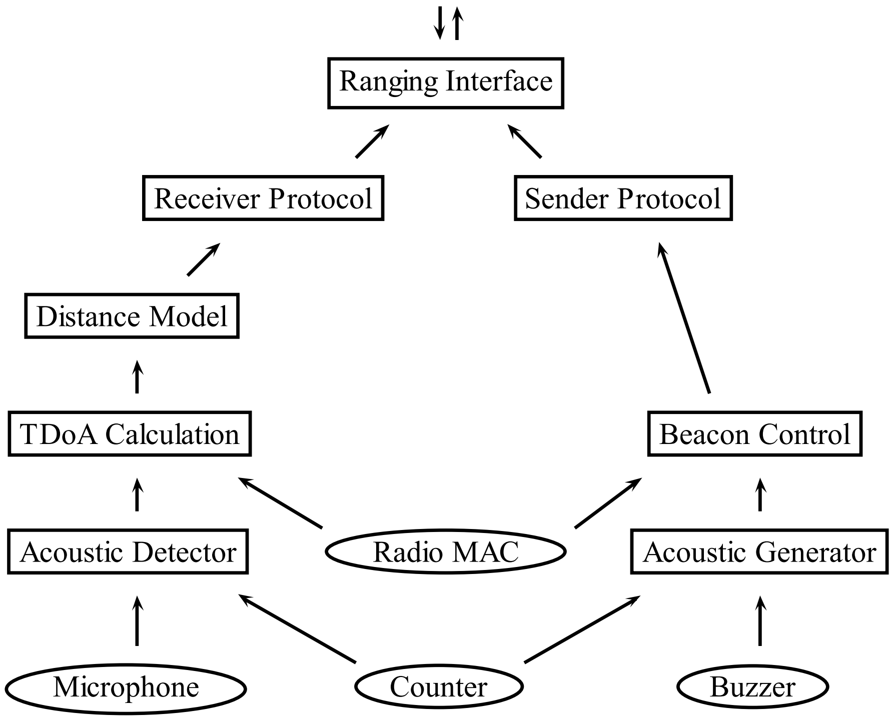


Figure 3.1: Component diagram of the ranging process showing the different layers and the asynchronous hardware access.

3.3.1 Radio MAC Time Stamping

The main goal is to measure the time interval between receiving a radio message and an according acoustic beacon. Due to the very large radio stack there is a huge delay between giving a message to the radio stack and the time the radio transceiver really sends the payload of the message over the air interface (Table 2.1). The time difference between trying to send and the actual send process is not deterministic. When using radio time stamping, an additional software interrupt fires when the start of frame (SOF) delimiter was sent successfully.

On the receiver side, the same problem arises when measuring the exact time of arrival of the radio message. Therefore, an asynchronous event fires after receiving the SOF delimiter. This timestamp is the base offset for calculating the TDoA between the two signals on the receiver side. However, the reception of a start of frame delimiter tells nothing about the content or type of message. This information can only be retrieved after decoding the message payload. The incoming messages in TinyOS are not buffered in any way, so that the latest receiving timestamp before getting the packet payload itself must be the according start of frame delimiter to this message. This fact makes the mapping of the exact timestamp easy and accurate on the receiver side.

Yet, there is some latency from the hardware. The propagation of the signal itself has the smallest impact so it can easily be neglected for the calculation of the TDoA. Since the payload length of the synchronization messages is constant, the latencies for encoding, byte alignment and decoding are the same for each transmission.

3.3.2 Acoustic Pulse Generation

The control of the acoustic actuator on Mica2 sensor boards is simply to turn it on and off with a GPIO port of the micro controller. So the only parameters to vary are the duration of an acoustic pulse and the time of silence between two consecutive pulses when sending a sequence. The duration of a single pulse should be as short as possible to reduce the power consumption. If the pulses are too short, the detection can be hardly accomplished. The signal of a pulse as depicted in Figure 3.2 will later be analyzed with a test program to find the minimum length empirically.

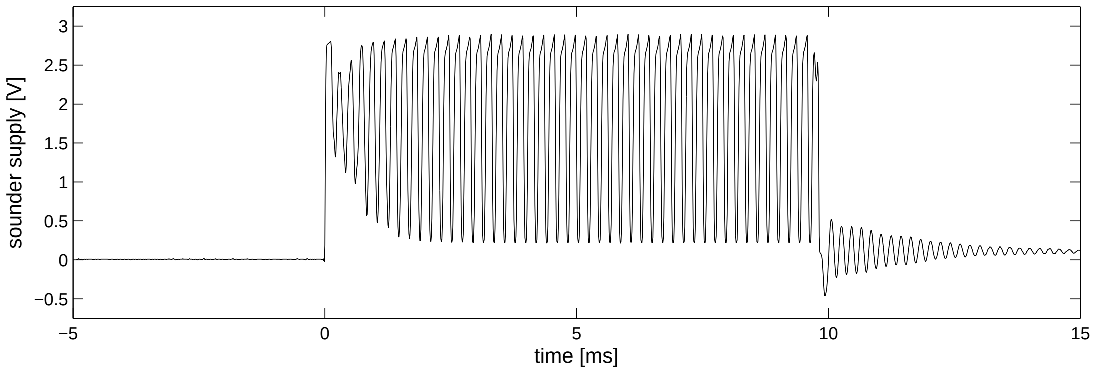


Figure 3.2: Supply voltage of the sounder, measured between pin M and G, showing the transient effects after turning the power on and off. Especially the tune-in effect makes if difficult to detect the start of the pulse exactly.

There are mainly two approaches in literature for sending acoustic pulses with fixed frequency for ranging purposes in WSNs (Section 2.3.3):

beacon pulse is the first and very simple method by sending a single acoustic pulse with a predefined length. The computational effort is marginal and the needed power is very low. Nevertheless, this signal is very vulnerable to environmental noise and reflections.

pulse sequence is the approach to send multiple acoustic pulses with predefined length and variable time in between. This means that the times of silence between two pulses of a sequence are different and additionally change with every sequence that is transmitted. The according timing information can be sent within the radio message. However, the generation of such signals needs more power. First, more pulses have to be transmitted and second the ranging process needs more time to complete. During this time, the micro controller has to be active and therefore has higher power consumption.

The project will integrate both of the mentioned signal generation mechanisms. The beacon pulse method will be good for ranging small distances or when low environmental noise is expected. This system will target office environments as well and for this reason, the pulse sequence will be necessary to make long distance ranging with interferences possible. Reference projects that use pulse sequences send up to 16 acoustic pulses per sequence. The power consumption rises nearly linear with the number of pulses sent, but the benefit in signal detection is only logarithmic as postulated in Formula (2.6). To enable modularity, the number of pulses is adjustable for every single ranging process.

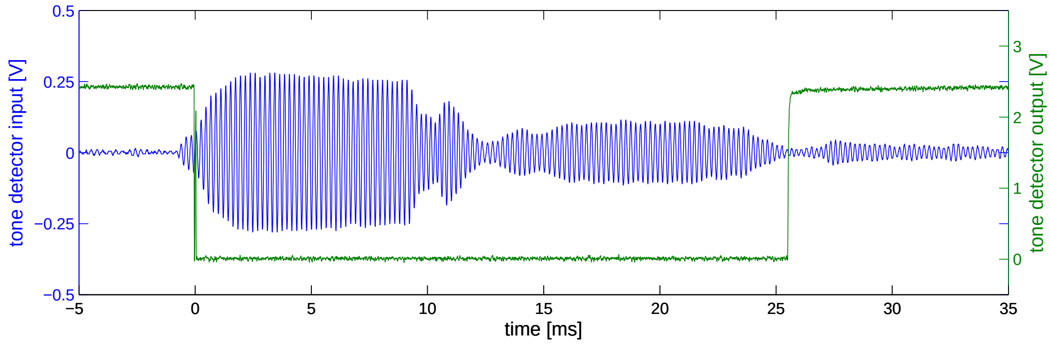
The timing for the signal generation should be done by a hardware counter or timer. Typical micro controllers have multiple hardware timer/counter components with specific output compare functionality. With this feature, it is possible to continuously run a counter and fire interrupts at exactly defined times. So the length of the pulses and the interlacing silence for pulse sequences is controlled by interrupts in asynchronous context. Another timing constraint has to be met. If radio and acoustic signals are sent at exactly the same time, the receiver has not enough time to prepare for receiving the acoustic signal after getting the information from the radio message. To counteract this problem a fixed delay is kept after sending the radio message, also timed asynchronously.

3.3.3 Acoustic Pulse Detection

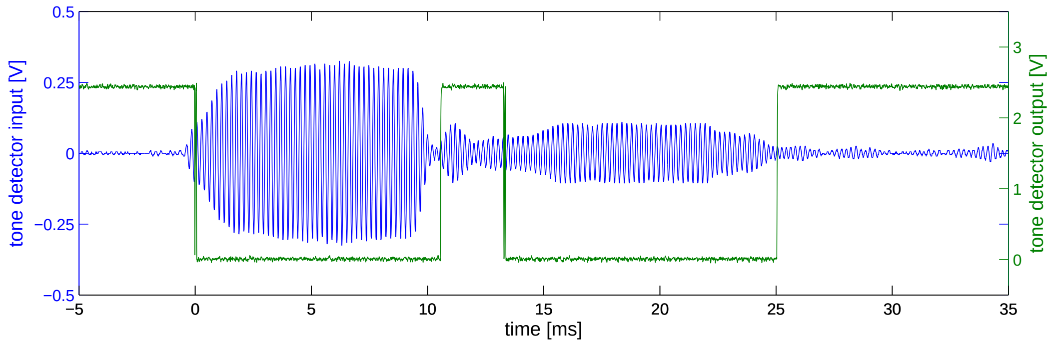
The analog tone detector of the Mica2 sensor board is used to detect an incoming acoustic signal. The microphone signal is amplified and filtered by a band pass before it is fed to the detector. The output of this device controls an external interrupt of the micro controller. The only adjustable parameter of this circuitry is the gain of the amplifier. There is no information of the signal amplitude, which is detected. This fact makes false detections unavoidable and the main effort of correct signal detection lies in detection, filtering and prevention of false detections. When sender and receiver have direct line of sight connectivity, the earliest detected signal is the correct one. This fact is only true if there is no environmental noise. The multi path components of the signal have always a longer way to travel and therefore are received later than the original one.

The output of the tone detector is at low level when there is a signal present. So there is the possibility to detect the rising and falling edge of the signal to improve the precision of the detection. Since the beacon length is known, the time difference between both edges can be used to identify false detections. This is only possible if there are no echo signals. As seen in Figure 3.3, an echo component partly overlaps the original signal and the tone detector may interpret this phenomena sometimes as one long pulse. A counter measurement would be to reduce the amplifier's gain. Then the echo signal would have a too low amplitude to be detected, which reduces the maximal ranging distance. This problem makes the rising edge for measuring the end of a pulse not reliable for TDoA a measurement because more failure and computational effort would be introduced than what is gained in terms of SNR.

Both signal forms mentioned in Section 3.3.2 have their own problems with different sources of error. When looking at the beacon pulse method this is mostly the environmental noise. The detection algorithm activates the tone detector after receiving the radio



(a) echo signal corrupts the detected pulse



(b) correct detection of the pulse length

Figure 3.3: Tone detector input and output signals when an acoustic pulse is received. The echo signal interferes with the correct end of signal detection and makes it impossible to use the rising edge of the interrupt to detect the pulse length.

message. From now on the time when the first interrupt fires, is considered to be the time of signal reception and the interrupt is disabled. If no interrupt is triggered within a specified time, the measurement is ignored. This method perfectly eliminates echo signals if there is a line of sight connection, but completely fails if not. When measuring long distances the possibility of a false detection between the activation of the microphone and the arrival of the signal grows. So this approach is inappropriate for noisy environments or large distances.

For receiving pulse sequences, a much more complicated technique is necessary. It divides into gathering the external interrupts and a detection algorithm, which tries to cross-correlate the received signal with the expected signal pattern. After receiving the radio message, the microphone is activated again. If an interrupt occurs and the signal does not bounce within some time interval, the current time difference is stored into memory. After a specified time, an amount of possible detections resides in the memory containing correct ones, echo signals and false detections from noise. Together with the timing information of the sequence, received with the radio message from the sender, a pulse sequence detection (PSD) tries to find the actual signal starting point. Typical cross correlation functions work with time discrete input data. In this case, the input data is event driven. Regarding to the constraints of a sensor node this algorithm has to be optimized for low memory footprint and short calculation time.

The principal task of the algorithm is as follows. The start of the known pulse sequence has to be positioned at this point of the input array that all following pulses fit best with specific detections of the array. As indicator, an accumulated absolute deviation is used. The use of the squared error is also possible but increases the computational effort. The result of the algorithm is a timestamp of the input array. To include the improvement in SNR, gained by using the pulse sequence, the standard deviation of the detected pulses is calculated and considered in the result.

This detection mode perfectly eliminates normally distributed false detections from background noise but is still vulnerable to short but high amplitude noise, which results in many detections within a short period of time. The available memory to store the detection times is limited and fixed at compile time and the calculation time of the PSD algorithm would increase. As counter measurement, such cumulated events can be detected and ignored. Another problem arises when a single pulse of the sequence is overheard. So the correct sequence is not likely to be found, because the error would rise very high. For this reason, the algorithm should be adopted to be fault tolerant to some missing pulse of the sequence. The main problem comes from the multi path components. If all pulses of such an echo were detected, the chance to be elected is the same as that of the correct signal. This comes from the fact that the error of the detection time is mainly Gaussian distributed around the true value [Fogh04]. A possible solution is to consider the already mentioned fact that the true signal is the earliest if a line of sight connection exists. Another possibility is to accept such false measurements and filter them out with statistical methods.

Since the gain adjustment is the only parameter to influence the hardware, it can be used to counteract false detections or reduce the error. Reducing the analog gain would result in less sensitivity of the detector and generally less false detections from environmental noise. Finding the trade off between correctly detecting all pulses and avoiding as many failures as possible is not trivial. For single beacon pulses, this can be accomplished by trail and error. If earlier detections yielded high errors, the gain has to be reduced and if some pulses were overheard, the gain has to be increased. When receiving pulse sequences the same considerations can be applied. There is also a possibility to increase the detection precision at the cost of more false detections. As stated by [Whitehouse02] the tone detector's setup time depends on the amplitude of the incoming signal. Referring to Figure 3.3 this edge is not sharp but could be made better by overdriving the amplifier. Now the increase of the signal's amplitude is higher and therefore the setup time of the detector. Under normal circumstances this deviation can be up to 1 ms or around 30 cm in distance.

3.3.4 TDoA Measurement and Calculation

This section covers the concept for measuring the exact time difference between radio message and acoustic pulse and the techniques for filtering the gathered data. As already mentioned in Section 3.3.1 a hardware timer is used to clock the TDoA. To reduce setup times, this counter runs all the time. When the radio message is received, the current value of the counter is the basic time offset. The pulse detection mechanism subtracts this time from the counter state, when an acoustic pulse is received. This technique nearly eliminates the setup time of the counter.

$$f_{CLK} = 7.3728MHz, \quad k_{prescale} = 8, \quad 1\, tick = \frac{k_{prescale}}{f_{CLK}} = 1.085\mu s \tag{3.1}$$

Hardware counters are controlled by an external frequency provided by crystals which suffer an unit to unit variation in frequency, and are rarely an exact multiply of 1 Hz. This fact makes it necessary to transform the measured timer-ticks into real microseconds. This procedure is stated in Formula (3.1) for a Mica2 node. So the measured timer ticks have to be multiplied by 1.085 to retrieve a result in microseconds. The unit-to-unit variation of the quartz oscillator is negligible in most cases. The transformations need not be made for every time measurement. It is still sufficient to transform the result of the whole ranging process before calculating the actual distance as described in the next subsection.

Under certain circumstances a TDoA measurement can fail. This is the case when the radio message is not received. This can be detected by estimating the latest arrival time. It is not possible to acknowledge these messages because of timing constraints. When detecting such a problem, it could be an incorrect CSMA approach or more likely a bad link quality. This information can be used by upper layers. For example, high packet loss could lead to ignoring the beacon for future use. Another detected error is the absence of the acoustic signal. This problem occurs by either a false gain setting (Section 3.3.3) or if the distance between sender and receiver is too high. Referenced projects have maximal ratings of 20 meters, as can be seen in Table 2.2, partially with special hardware and in silent environments.

A ranging process normally consists of many single TDoA measurements. To provide flexibility, there are three basic parameters to control the measurements. First, it should be possible to mix the two pulse types on the fly for example as reaction to failures. Another parameter specifies the timing between two measurements. It can either be a constant time between sending the acoustic pulses or an interval between the end of a measurement and the start of the next one. This dynamic approach is especially suitable for pulse sequence mode because depending on the pattern, the signal duration can change. This makes it possible to make as many measurements as possible. The third parameter targets the stop mechanism of the ranging process. The first possibility is that the sender starts a fixed amount of measurements and automatically finishes afterwards. The other method is to send acoustic pulses until the receiver actively stops the sender by sending a radio message. All these flags are sent with the radio message every time. When using pulse sequences, information about the sent pattern must also be transmitted, because it changes every time.

When the ranging process is finished, the gathered data has to be analyzed and filtered. A coarse filtration is already made by the pulse detection algorithms by ignoring measurements with timeouts and many false detections. A possible approach is using the arithmetic mean together with the standard deviation. The calculation is relatively simple but the result is heavily influenced by outliers. A better solution is to work with the values from the five-number summary of a dataset; minimum, lower quartile, median, upper quartile and maximum. When a sorted dataset is split into four equal peaces, the five values represent the limits of these four peaces. For example, the first quartile cuts off the least 25% of the dataset. Which value to take depends also on the pulse detection method. When detecting sequences some echo signals will increase the inner quartile distance. This fact can be used to identify this problem and use the lower quartile for the resulting TDoA. If this is not the case, the median will be the best estimation for the result. The number of outliers can also give information to the quality of the analyzed data. When detecting beacon pulses these conclusions cannot be used identically. Since this form of detection normally underestimates the real distance.

To work with these statistical methods the measurement data has to be sorted. The used algorithm should operate in place for a small memory footprint and should operate online if possible. Insertion-Sort is a very simple but yet effective algorithm for this case. Although it has a worst case run time of $O(n^2)$, the constant factor is very low due to its simple mechanism. It is also possible to use other comparison based sorting algorithms. A third possibility to analyze the ranging data is density estimation. This method would identify all accumulation points of the dataset, if there are echoes present. The lowest maxima would be the real pulse sequence and taken as the result. The high complexity however is a major drawback and for this reason the quartile-based approach is used for data filtering.

3.3.5 Distance Model

After finding a good result for the TDoA it has to be transformed into a distance value. To reduce complexity the timer ticks to microseconds (Formula (3.1)) conversion can be made in one step with this transformation. As discussed already, environmental conditions affect the speed of sound. Most projects correct, if at all, only the temperature dependency because it introduces the highest divergence. The Mica2 sensor board ships with a built in temperature sensor that can be measured by an ADC input. The targeted operating range is from around 0°C up to 30°C to use common approximations. Inputs for the transformation are the temperature value of the ADC and the measured time interval which is not yet converted into real time in seconds. The first task is to calculate the thermal resistance with the parameters of the ADC and the value of the reference resistor. This value can be converted to a temperature value using a logarithmic approximation. At the final stage this temperature can be used to get the current speed of sound following Formula (2.7) and the multiplication with the real TDoA gives the distance as result.

Especially in indoor environments, the temperature is not always constant over the whole area. This makes it necessary to cope with differences between the temperature of sender and receiver. The exact distribution along the signal path is not known, so it must be approximated linearly by taking the arithmetic mean of the two temperatures. A problem is direct solar radiation onto one of the nodes. This heats up the complete mote and significantly increases the measured temperature although the normal air temperature is lower. So if the two values differ too much, the use of only the lower temperature would reduce the failure.

With the considerations mentioned so far, the transformation cannot be accomplished. There is still a constant offset in time which would be present if a distance of 0 m would be measured. This offset value has several reasons. A constant and user defined part comes from the interval between sending the radio message and the acoustic pulse. Other deterministic delays come from hardware and software but should have little variations. The value of this offset will be measured in an offline phase. If there is a significant unit-to-unit variation the value must be calibrated for every single node, otherwise it is globally valid.

3.3.6 Ranging Protocol

This section describes the desired overall operation of a ranging process between s single sender and one or more receivers depending on the chosen mode. The ranging process begins by configuring the mode and starting the process. The result is the distance itself with some value containing information about the detection error rates (Section 3.3.3), link quality and possible noise, which could have corrupted the process. Such a value could be called confidence where low values could lead to restart the measurement or ignore it completely.

The process of ranging has to be framed into a communication protocol between sender and receiver. Every ranging has to be initiated by a receiver because the main goal of this thesis is to provide self-localization in a fixed infrastructure consisting of beacon nodes. One main goal of the initialization phase is that all participants agree on the parameters of the ranging process already described in this section. At the end of the ranging process a finishing phase should stop both sides operation and can be used to exchange some relevant data, for example about link quality or the current temperature on the sender side. It is very essential that radio messages of the protocol really arrive at the receiver. Therefore, some sort of acknowledgment mechanism has to be integrated. The messages should also be resent if the receiver did not reply. This is essential because a state machine residing behind such a protocol could get stuck if the reception of a message is necessary to switch between different states.

The protocol consists of two basic operating modes. The first and simple one is the 1-to-1 ranging which provides a fast ranging process between two designated nodes. The more complex second mode is the 1-to-n ranging. In this case many receivers can register to a single sender and receive the TDoA measurements at the same time. Both modes are now described in more detail:

1-to-1 ranging is a simpler but fully customizable ranging mode. A receiving node asks an already known sender to start a ranging. Additionally the receiver specifies the pulse generation mode and the type and duration of the interval between single measurements. If the sender is busy with another measurement, it answers with an approximate finish time as information for a retry. Otherwise, it gives a positive reply and immediately starts with the ranging process. When the receiver gathered enough valid results, it stops the process by sending a message. The sender stops the process and replies with the number of sent pulses to get information about the link quality.

1-to-n ranging intends to reduce power by coordinating different ranging requests. This means that the sender collects requests and periodically answers these in a ranging broadcast. The first step for a receiver is to ask a sender for a ranging. Additionally the parameters for the pulse generation type, the type of interval and the amount of pulses that are sent are specified. These values represent desired values and the sender decides whether to fulfill these, because all different request parameters have to be integrated. As reply, the sender gives information if the desired values were accepted and the time until the ranging is broadcasted. If the receiver wants to cancel the process, it can be done now. If the specified time is over, the receiver gets another message from the sender, which announces the amount of pulses that will be sent. After that, the sender begins with the ranging process and finishes after a defined amount of pulses. This amount is the maximum value of the received requests.

The decision which method is taken comes mainly from timing constraints. If the receiver wants to have the result as fast as possible, the 1-to-1 method should be used. When the power consumption of the sending nodes should be minimized, the 1-to-n method is generally the better mode. A higher density of unknown nodes makes the 1-to-n method more power efficient. So a trade off has to be found between the duration of the positioning process, which depends strongly on the ranging duration, and the overall power consumption.

The protocol itself does not maintain a list of available sender nodes. This should be done in an upper layer, because it is not required that every sender is also an anchor node with a specified position. Additionally it should be possible to run a sender and a receiver protocol on the same node. This makes it possible to get bidirectional ranging information as postulated in Section 2.3.5. The strategy for a ranging between two nodes moves completely to the next layer, which already implements a certain positioning algorithm.

3.4 Positioning Algorithm

As already pointed out in Section 3.1 the calculation of the position is assumed independent from the method of ranging. This modular design approach eases the possibility to design and implement the positioning part separately. There is only one interface definition to separate the two parts. Figure 3.4 gives an overview of the main components involved into the positioning design. The „Ranging Interface“ is the connection to the underlying ranging process as depicted in Figure 3.1. The following parts of the concept describe the major parts of a multilateration based positioning approach as described in Section 2.4.3.

3.4.1 Beacon Detection

The beacon table holds all known position data including the distances measured in between. Further the environment temperature is logged which is considered in the distance

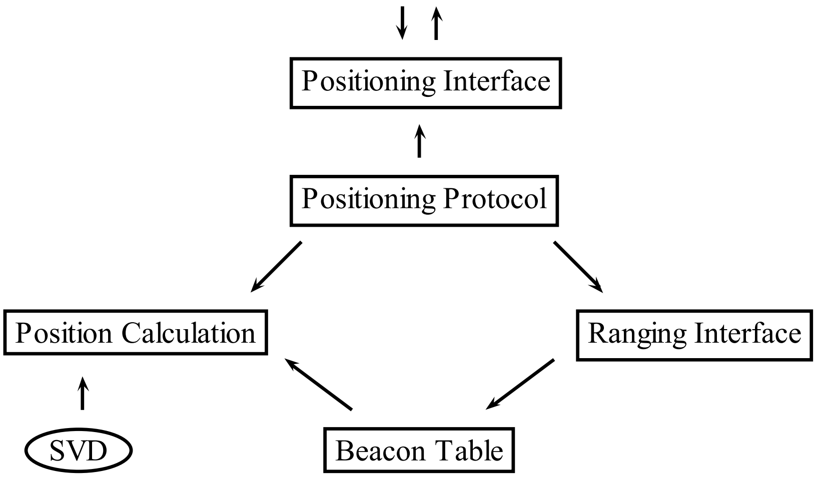


Figure 3.4: Component diagram of the positioning process showing the interactions of the main components

model as described in Section 3.3.5. The main goal of this program part is the exchange of reference and metadata between the nodes. To start a three dimensional position calculation with multilateration, the positions and distances to at least four different non-collinear reference nodes have to be available. Therefore, all nodes in the networks have to propagate their own positioning data if available. The technique for data exchange is based on broadcast messages and such a data table is found in many routing protocols.

Every node collects a limited amount of reference data in a table structure. Additionally every reference and assistant node (see Section 3.2) propagates its own position and current temperature measurement. If one of the parameters changes, the table entry is updated. Additionally there is a maximum lifetime for the information, after which the entry is deleted. The lifetime is reset after every successful reception of the broadcast message and there are two possibilities why the time can expire. First is the deactivation of the node either by the application or due to a power down. Another possibility is a bad radio link, which indicates a too large distance between the nodes for an acoustic ranging process.

Due to the limited table size, which comes from memory constraints, an appropriate replacement strategy has to be considered. As mentioned earlier, a typical topology for the application involves a limited amount of reference nodes and many unknown nodes, which transform to assistant nodes after positioning. So the basic strategy is to prefer reference nodes because of their higher trustability. Only if there is enough space in the table, additional assistant nodes are considered for adding redundancy. The node type can be verified against the trustiness, because no assistant node can have as high values as the reference nodes.

The second dataset stored in this table regards the distance measurement. After a successful ranging process, the results are inserted into the according table entry. Once the distance to a specific reference node is set, the dataset remains unchanged as long as the coordinates do not change. This can happen if an assistant node recalculates the own position or the node manually changes its position and propagates this information.

3.4.2 Position Calculation

After enough position and distance data was gathered, the calculation process can begin. The project targets a normal multilateration process. After the transformation in an over determined linear equation system, this system can be solved with a pseudo inverse based on the SVD of the input matrix. This section will first cover details about this process and afterwards some consistency checks that have to be made to discover and prevent errors in the calculation.

As mentioned in Section 2.4.3 there are several methods to solve a linear equation system and the chosen method is the SVD, because of the high success rate. To speed up the implementation phase, the actual calculation including many tasks from linear algebra is not implementation from scratch.

Before the calculation starts, the beacon table has to be filtered to find the best-suited entries. The first choice would be all entries with the highest possible trust, therefore the reference nodes. Additionally the confidence of the ranging process should be as high as possible. The trustiness of an assistant node includes the single confidence values of the included rangings. If the first constraints resulted in too few beacons, it is better to reduce the necessary confidence first.

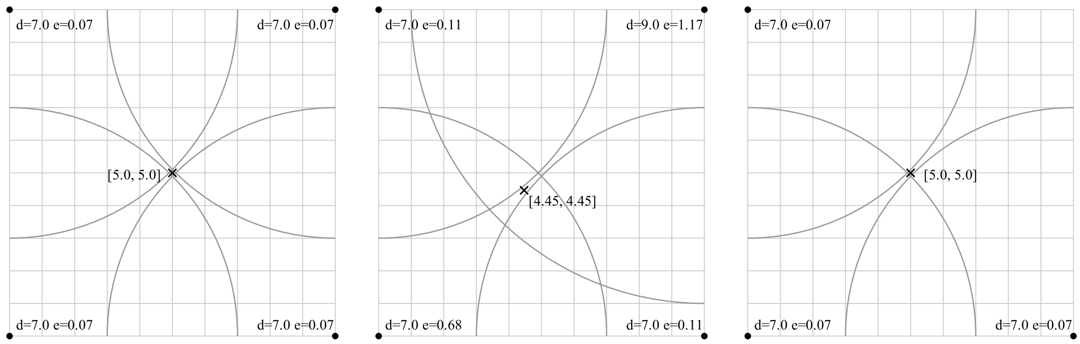
The advantage of the SVD is that the inverse can be found although the input matrix is singular. A special case of singularity leads to completely false results. If one coordinate of all considered beacons has nearly the same value, the result for this axis is unpredictable. This is the case if only beacons on the ceiling are used. The z-axis of the result would be the same if all inputs are equal. If the differences are minimal, this results in an unreasonable high value. So this special case, which can be easily detected, has to be avoided.

After a successful calculation, a test can give information about the result's precision. The error according to Equation (3.2) sums up the differences between calculated distances and the ranging results d_i for all n beacon nodes:

$$e_{SVD} = \sum_{i=1}^{n} \left| \sqrt{(u_x - x_i)^2 + (u_y - y_i)^2 + (u_z - z_i)^2} - d_i \right| \tag{3.2}$$

Too high values of this error indicate imprecise ranging data. Even if only one ranging has an high error this yields to an high value of e_{SVD}. To optimize the calculation process, this single beacon should be detected and ignored for a new recalculation. There is no trivial correlation between the outlying beacon and the highest deviation of the calculated distance from the measurement. If the correct beacon was ignored the error of the new calculation will be lower because the found residuum will shrink. An example for this strategy is presented in Figure 3.5.

Nevertheless, ignoring one beacon can lead to different problems. First is to make the resulting matrix singular. This can occur for example when the ranging to a single node on the floor is imprecise and all other nodes are mounted on the ceiling. The other



(a) accurate distance values result in a perfect result

(b) if a single outlier disturbs, result has a high error

(c) after eliminating the correct beacon the result is perfect again

Figure 3.5: The depicted graphs show a 2D testbed for position calculation using the SVD to solve the over determined linear system with linear LS.

problem is that removing a beacon can make the equation system under determined. In this case, the constraints for filtering the beacon data should be relaxed if this is possible.

3.4.3 Positioning Interface

The positioning interface gives complete access to the implementation of this thesis. With regard to the design principles postulated in Section 3.1 the interface gives limited control for an adaptive behavior of the positioning system. This mainly includes the targeted effort for the process parameters that can be translated into required energy for the task. This energy constraint is seen as minimal available value and can rise due to bad environment conditions or hardware faults.

Due to the layered design principle of the software, there is no more detailed control of the sub layers from the top. A small drawback of this approach is that the user application cannot directly react to unpredictable failures on low layers. This task is done by every single layer to decrease the complexity of the overall system. So the interface for the complete positioning system is reduced to four commands and two events:

activate starts the overall system, making it ready to receive start commands in the future and initializes all sub layers.

deactivate immediately powers down the system and all dedicated hardware and interrupts a running positioning calculation.

start initiates a positioning approach and sets parameters for the targeted quality for the ranging process.

invalidate resets the currently set position. In most cases, the user application can observe position changes of the mote for example from an acceleration sensor.

finished is signaled when the positioning task is done without failures. In addition to the coordinates, a trustiness is given to evaluate the calculation

error is thrown when failures occurred that corrupted the process or if the result would be too imprecise.

3.4.4 Positioning Protocol

The coordination of the positioning process is made by the protocol. This component fills the gap between user application, beacon table, ranging process and the calculation algorithm as shown in Figure 3.4. The process flow in short form after getting a start from the user application is as followed. For each unmeasured entry in the beacon table, a separate ranging process is initiated. If the gathered data has a high enough confidence, the most valuable beacons are taken into account for the position calculation. If the resulting error is low enough, the process is finished. Otherwise, a new calculation is initiated with different beacons. If no solution can be found an error is raised.

The protocol receives all commands from the user application that are necessary to make a complete positioning. Therefore, the component provides the main interface described in Section 3.4.3. With the activation of the positioning system, the main task is to activate the beacon table. This enables the search and storage for beacons in the neighborhood. Additionally all sub layers are initialized. This includes the sender mode of the ranging process, which means that the mote can participate in a ranging as a beacon.

After the start command, the actual positioning task is executed. If there are less than four beacons available, an error is thrown because there is no solution to the positioning problem. The first thing to do is measuring the distances to the different beacons. So a ranging process is started to every beacon that could be essential to fulfill the calculations. If some distance parameters already exist from earlier measurements, these beacons are omitted.

The next step is to filter out the appropriate beacons to increase the chance of a accurate position calculation as described in Section 3.4.2. Afterwards the linearization tool is started which converts the beacon table data into matrices representing the linear equation system. On this system the SVD algorithm is started, the result is transformed to the pseudo inverse and the resulting coordinates are calculated. Depending on the error of Equation (3.2), the positioning task is finished or recalculated with other beacons.

After the positioning process, the node becomes an assistant referring to the terminology from Section 3.2. That means the broadcasting of the own position information begins and this node can act as beacon to help other unknown nodes to find their position.

3.5 Power Profiling and Middleware Integration

Low power consumption is a very important design goal in applications for WSNs. The literature study showed that many projects keep low power consumption in mind during concept and design phase. There are no real energy values available to compare a new solution to already existing implementations for node positioning. One approach to estimate the needed energy for a positioning approach is to sum up the power consumption during the execution of the algorithm. In the case of an acoustic ranging system the main power is consumed by the micro controller itself, the acoustic actuator, the microphone amplifier circuit and the radio module.

During the implementation phase single components of the software can be measured with methods described in Section 2.5. In a layered software design, every layer can be measured step by step to get a complete power profile of the program. The result is a overview of the power consumption for the main components. Afterwards an optimization for power can be made by altering algorithms or communication protocols. During the concept phase the minimization of the sent radio packets and acoustic beacons generally reduces the overall power consumption of the system.

The single parts of the implementation are profiled with the power measurement test bed as referred to in Section 2.5. A fundamental measurement is the needed energy for the transmission of a single acoustic beacon and the calculation of the according TDoA with the sent radio message. The power consumed during the execution of a complete ranging operation will be another important fact to achieve adaptive fidelity in the overall application. The last measurement considers the power consumed during the calculation of the SVD during the position calculation. As this calculation only utilizes the processor, a good energy model of the mote could also be used with a simulator to achieve the same result.

Providing the current nodes position is a feature that can be abstracted by a middleware solution. Such an enhancement can increase the significance of gathered data in traditional database oriented middleware systems [Hadim06] as the data can now be bound to a specific position. This makes it possible to reduce the complexity of the deployment phase and allows adding mobile nodes into the sensor network.

The positioning system can also benefit from a middleware system in terms of reduced program complexity or memory consumption. Most middleware systems provide a sophisticated radio stack that is more advanced to the standard TinyOS variant by providing message queuing, automatic retransmission or handshake mechanisms. This can remove code complexity from some components of this system (Sections 3.3.6, 3.4.1, 3.4.4). The temporary memory needed by the SVD algorithm could also be administrated by a middleware system making it possible to reuse these memory-blocks in other components of the user program. The functionality of the system itself can be integrated with the same interface as described in Section 3.4.3.

3.6 Tool Chain

The targeted hardware platform for the implementation is the Mica2 from Crossbow [Crossbow1]. This wireless sensor node is basically equipped with a Atmel AtMega 128L micro processor [Atmel] and a Chipcon CC 1000 radio transceiver [Chipcon]. To incorporate sensing capability different extension boards exist. The acoustic localization system presented here build upon the MTS 3xx [Crossbow2] board which provides a fixed frequency piezo-electric buzzer, a microphone and a phased locked loop tone detector in hardware.

The most common approach to develop software for this sensor nodes is the TinyOS operating framework developed by U.C. Berkeley [TinyOS]. The framework provides event based core components that abstract the hardware completely and gives methods to implement the software independent from used hardware. The programming language for TinyOS is network embedded system C (nesC) [Gay05] and the used version is 2.1. The development platform is a Cygwin environment running on a Microsoft Windows system. The compiler, hardware programming and make tools for TinyOS are installed within Cygwin. Additionally there is a simulator available which can virtualize a network of nodes based on a specially compiled version of the nesC program [Levis03]. This TinyOS simulator (TOSSIM) can be controlled and debugged by Python scripts.

MATLAB is used at first instance for the development of the core algorithms as for example the PSD described in Section 3.3.3. These scripts are later used to verify the results calculated by the hardware. The code fragments afterwards are transformed to normal C, whose syntax is almost compatible with nesC. This intermediate step is done due to limited and more complicated debugging capabilities of the final TinyOS modules.

Especially the lower level implementation of the ranging method has to be tested directly on hardware, because of the limited capabilities provided by TOSSIM. The simulator gives no possibility to implement an acoustic channel between nodes in the network. So the bottom two layers (see Section 3.1) of the software cannot be simulated correctly. Nevertheless, the positioning algorithm should definitely be simulated before deployment. Therefore, it is necessary to provide a dummy implementation of the ranging process which simulates the distance measurement.

To give a short overview here is a list of the used software tools:

- Windows XP Professional with installed Cygwin version 1.5.24-cr
- NesC compiler version 1.2.8 and AVR GCC compiler version 4.1.2
- Java Development Kit version 1.5.0.11 and Python version 2.5.1
- TinyOS Source version 2.1.0.2 and Tools version 1.2.4

3.7 Evaluation Metrics

There are many performance metrics used in literature to categorize a ranging or positioning system. Most of them cope either with the needed energy to make execute or a distance based error to specify the precision and accuracy of the system [Jordt06]. Another necessary value to determine the robustness of a positioning system is the success-rate. A perfect solution would incorporate high accuracy, precision and success-rate while having low power consumption.

The increase of either quality factor always increases the needed energy too. More measurements or complicated calculations must be done or a backup solution has to be executed after failing with the initial variant. Finding a trade off between quality and power consumption is the goal for an optimization problem. If a positioning system can dynamically adapt the behavior by giving constraints for maximum power ratings and minimal achieved result quality, this would be an optimal solution. For that reason, this positioning system has different parameters and methods to improve the quality for the cost of higher power consumption. As described in Section 3.3.2 the type of acoustic beacon generation can be switched from error-prone beacon pulses to more noise resistant pulse sequences with a variable amount of single pulses. The amount of TDoA measurements can also be configured during runtime. As the ranging process returns a precision value, the upper layers can repeat the ranging until a dynamic constraint of quality is reached. The positioning system itself has also methods to improve quality. If more beacons were considered for the calculation, the quality would rise, but also the needed energy.

To evaluate and compare this solution with other projects from literature, a meaningful comparison value has to be found which copes with the quality factors as well as the needed energy. The so-called accuracy-power-product (Equation (3.3)) should be as low as possible and the success-rate and relative error are given in percent. High success means that most of the nodes could be positioned, and the relative error means the distance ratio between calculated position and the actual node position.

$$accuracy\,power\,product = \frac{1}{success\,rate} \cdot relative\,error \cdot energy \tag{3.3}$$

Chapter 4

The Acoustic Indoor Positioning System

This chapter illustrates the implementation of the concept presented in Chapter 3. The software basically splits into two main components, ranging and positioning, according to the design principles postulated in Section 3.1. Side effects of this rule are also two programs that can be separately compiled and used for detailed functional tests, test data generation or debugging. Additional test and debug programs were made for several reasons, for example to keep track of the correct functionality during the implementation. Others were made to define constants and constraints empirically such as minimal acoustic pulse length for example. Special debugging versions are needed because preparation and transmission of this data disturbs tight timings and complicates the control flow. The software modules are described bottom up, same as the implementation was realized.

The TinyOS source code consists of Components, Modules and Interfaces. As nesC is based on ANSI C there is the possibility to declare structures and enumerations in header files as well as including source files. The single Modules represent the functionality of a TinyOS program. Implemented functions are *provided* to others by using interfaces. If a Module needs some functions from other modules, it has to *use* their interfaces. Components configure the *wiring* of interfaces between modules providing functionality and those who need their features. If modules have to be instantiated, this is done by the components. Components also have interfaces to other outlying components. The TinyOS application is a special form of component that does not provide nor use any interface.

Data exchange with interfaces can be done by commands or events. Commands are equal to function-calls in C and can have parameters. The events are part of a observer pattern like mechanism provided by TinyOS. The subscription to the provider has to be done during compile time by wiring in the components. If the event is *signaled* by the provider, every observing module is notified. This paradigm comes close to software interrupts normally used in micro controller programming. Another feature of TinyOS are tasks which can be *posted* everywhere in the code. This task is executed when the current control flow ends and the controller would normally go to an idle state. Further details about TinyOS can be found in the documentation and the according TinyOS extension proposals (TEPs) [TinyOS].

4.1 Ranging Software

The whole implementation of the ranging process resides in the folder `ranging` of the contributed source tree. The following sections cover the description of the source code for sender and receiver in parallel to keep track of the inter node communication at each layer of the design. The figures 4.1 and 4.2 show a simplified component diagram of the low-level components implementing the pulse generator for the sending node and the pulse detection, TDoA calculation and data filtering for the receiver side. Rounded rectangles represent modules, rectangles are used components, dashed rectangles represent instances of generic components and the gray ellipses are the interfaces provided by the component. The arrows represent one or more interface connections between the modules. In the following chapter, all modules are written in typewriter style and can be found in the component diagrams.

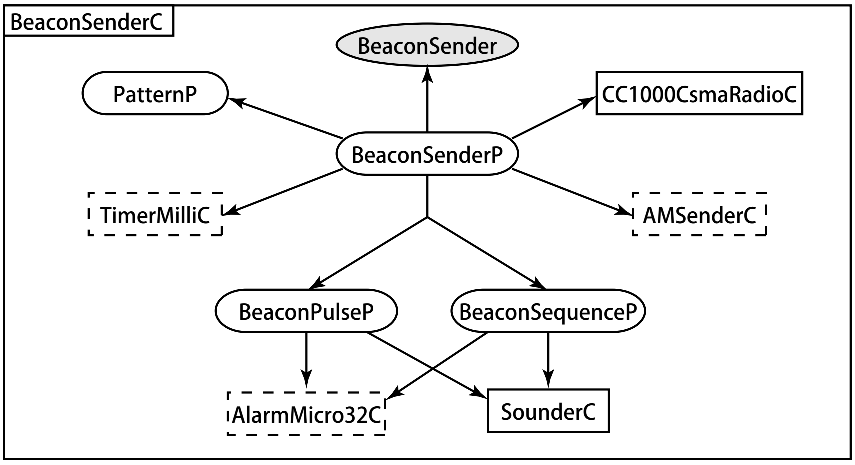


Figure 4.1: TinyOS Component Graph showing the main parts of the pulse generator. The single modules are described in the following sections.

4.1.1 Acoustic Pulse Generation and Detection

As mentioned in Section 3.3.2 two different approaches for the beacon signal generation exist, a single beacon pulse or a pulse sequence. Both methods are implemented as part of this thesis. To form perfectly timed acoustic beacons, two different hardware components of the Mica2 have to be used. These are the integrated 32-bit asynchronous alarm timer provided by TinyOS and the power control of the buzzer integrated on the sensor board. The alarm timer is built exclusive on top of the micro controller's 16-bit timer with output compare functionality. All access commands are asynchronous to provide real time control. This gives the possibility to fire an event exactly at a specified time in the future. The

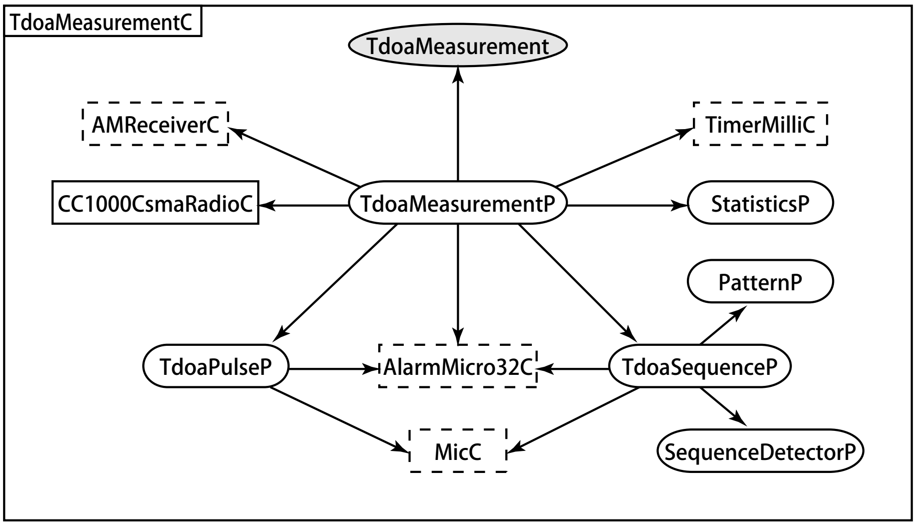


Figure 4.2: TinyOS Component Graph showing the main parts of the pulse detector and TDoA calculation. The single modules are described in the following sections.

timer is always running, making it possible to get the current value at any time with an overflow occurring every 75 minutes.

Beacon Pulse Generation and Detection

A single pulse with a specified length has to be formed a predefined time after the synchronization message was sent via the radio. The receiver side has to capture the exact timer value at the start of the pulse. Together with the time of arrival of the radio message a TDoA value can be calculated. The modules `BeaconPulseP` and `TdoaPulseP` implement the mentioned behavior. The constraints for pulse length and acoustic delay are all defined in the header file `tdoa.h` as all the other constants needed by the ranging program. The current implementation uses pulses with a length of 5 ms that are delayed for 10 ms after sending the radio message. The generation of the pulse is implemented straight forward by a simple state machine switching the buzzer power depending on the alarm event.

The receiver side uses the `MicC` component from TinyOS to gain access to the microphone amplifier and tone detector output. The receiver module activates the external interrupt of the tone detector in response to the start command. When receiving an interrupt the TDoA has to be calculated. Therefore, the already known time of arrival of the radio message is subtracted from the current timer value. The result has to be validated for consistency, timer overflow or timeout. A timeout occurs if no interrupt was received within a specified time after the sync message reception. This time is defined upon the maximum ranging distance, currently 30 ms which equals ten meters of distance.

To improve detection reliability the output of the tone detectors bouncing is filtered in software. So the interrupt is only valid if the signal level after the interrupt stays constant for some time. During first tests a bad behavior of the `MicC` component was observed.

When activating the interrupt pin, an interrupt event is fired after a constant time. To counter this problem the first occurring interrupt event is ignored.

Pulse Sequence Generation and Detection

The pulse sequence is defined as a train of fixed length acoustic pulses with a variable time of silence in between them. The so-called pattern represents the shape of such a sequence. The implementation supports up to 9 pulses per sequence with the same length as normal beacons. The times in between can be varied for every sequence between 1 and 15 ms. A pattern is generated dynamically by the acoustic sender with the `PatternP` module. For best results, a pseudo random generator initialized with the node id is used to generate the sequence delays. The pattern itself is stored as a 32 bit integer consisting of eight nibbles representing each interval of silence. If not all nine pulses are used the remaining intervals are set to zero. The pattern has to be transmitted together with the synchronization message to the receiver to identify the pulse sequence correctly.

After the pattern generation, the pulse sequence is generated very similar as mentioned before by the `BeaconSequenceP` module. After sending the first pulse, the alarm timer is set to the first value of the generated pattern. The relation between generated pattern value and signal generation can be seen in Figure 4.3. After the timer event, the next pulse is formed and so on. It is very crucial that the time difference between the rising edges of two pulses is exactly the same as the stored pattern value. However, the setup phase of the alarm timer needs some processing cycles and the timing is distorted. To fix this problem a constant correction value is introduced to equalize this delay.

The receiver module `TdoaSequenceP` completely differs from the beacon pulse approach. The tone detector interrupt is activated after receiving the synchronization message containing the current pattern. As the receiver has to detect many pulses, the module records all timer values when a tone is detected. After this first step the data is analyzed based on the pattern with the PSD algorithm described in the next paragraph. After getting an interrupt, the according timer value is read and stored into a temporary array of fixed length. It is very crucial to detect bouncing of the tone detector output to reduce redundant detections and therefore the needed memory to store the timer values. To save memory the time values are already stored as time differences to the reception of the radio message and additionally are divided by eight. This makes it possible to store a time of up to 500 ms and a resolution of 8.6 μs within a 16 bit data type. The interrupts are captured until a specified timeout, which can be calculated with the pattern information. After this time the detection algorithm is started and the according results are checked for consistency.

PSD Algorithm

The PSD algorithm is a core element to improve signal detection based on pulse sequences as already mentioned in Section 3.3.3. The basic idea is to fit the known pulse pattern into the array of captured timer values. The module `SequenceDetectorP` is used by the receiver to fulfill this task. There are two inputs for the algorithm, the captured time

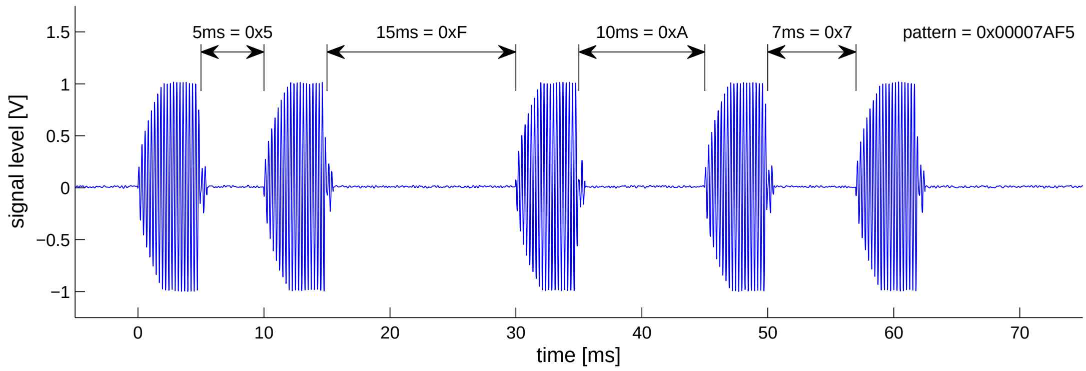


Figure 4.3: This graphic shows the connection between the pattern value and the generated signal form. The least significant nibble of the pattern represents the delay between the first pair of pulses.

array and the pattern value that is also converted into a time array containing all times of rising signal edges with the first pulse set to zero.

The algorithm iteratively goes through the data array as shown in algorithm 1. The time array (M) is overlaid with the pulse array (P) generated from the pattern value matching the first pattern pulse exactly with the current index of the data array (off). The next step is to find those captured events that fit best into the remaining pulses of the pattern. The sum of the deviations (acc_diff) is used as a minimization criteria. Now the algorithm continues by shifting the overlaid pattern to the next data point. The position with the lowest deviation ($index$) is the best guess for the actual signal position. Yet there is the challenge of echo cancellation which is trivial in the beacon pulse approach but not for pulse sequences. To improve the precision of the calculation several slight modifications are made to the algorithm mentioned. For example, during iteration the best two hits are saved and analyzed at the end, taking the earlier starting time to prevent echo detection. Further discussion follows in the evaluation chapter.

Minimal Pulse Duration and Delay Specification

The duration of the pulse should be as short as possible to reduce the power consumption. There is a lower bound which is defined by the following two factors. The first is the transient effect of the piezo-electric sounder. In the case of the MTS-310 sensor board [Crossbow2] this is typically 2 ms before the maximum output power is achieved (Figure 3.2). An additional boundary comes from the tone detector on the receiver side. The circuitry needs some time to lock the internal PLL to the measured signal in order to detect if a tone is present. So it has to be accomplished that during this time the signal is still present at full amplitude and measurements have shown that this setup time can be as long as 1 ms which can be seen in the graphs of Figure 4.4. The resulting minimal length of a single acoustic pulse is therefore given with 3 ms.

Another negative behavior of the sounder is the reduction of the maximal output power when two pulses are made within a too short time. To avoid this effect two consecutive

Algorithm 1: Simplified pulse sequence detection without error minimization

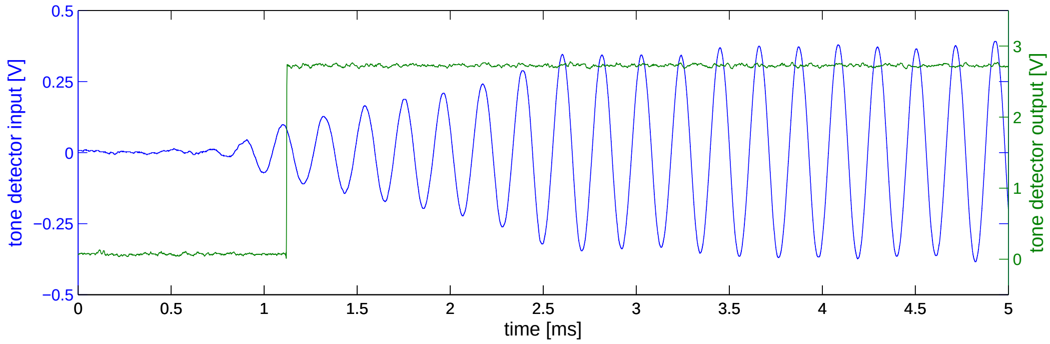
```
error ← ∞
for start_idx = 1 to Data.size − Pulses.size do
    data_idx ← start_idx + 1
    acc_diff ← 0
    for pulse_num = 2 to Pulses.size do
        min_diff ← ∞
        for search_idx = data_idx to Data.size do
            dev = Data[search_idx] − Pulses[pulse_num] − Data[start_idx]
            if |dev| ≤ min_diff then
                min_diff ← |dev|
            else
                data_idx ← search_idx
                break
            end if
        end for
        acc_diff ← acc_diff + min_diff
    end for
    if acc_diff < error then
        error ← acc_diff
        index ← start_idx
    end if
end for
return Data[index]
```

pulses have to be separated by a minimal time of silence to stabilize the power supply. From different measurements this time should at least be 5 ms. When breaking this rule, the following pulse reaches the maximal amplitude with a higher latency.

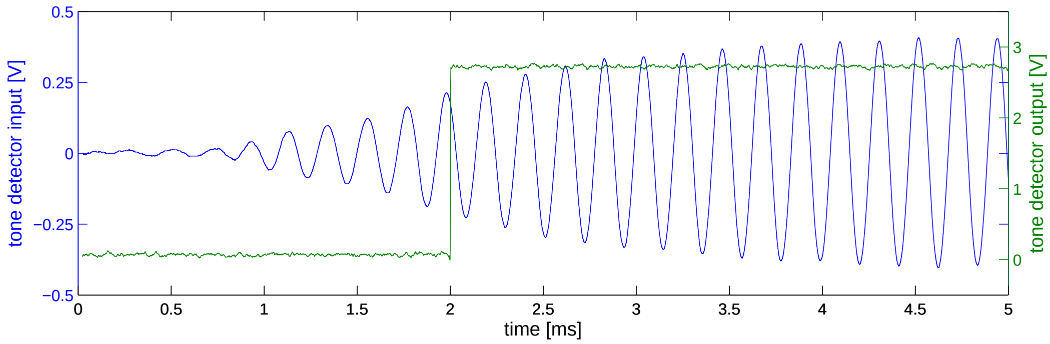
To ease access for the upper layer the modules for both types of beacons have the same interfaces to communicate. Additionally both implementations could even be used in parallel. The control of the hardware has to be done in the upper layer. This includes TinyOS's resource requesting, component initialization and parameter configuration as the microphone gain.

4.1.2 TDoA Measurement and Data Filtering

This section covers the next layer in the design, measuring successive TDoA values and minimizing the error with statistics. The result of the receiver side is a guess for the measured TDoA value. The inputs for the sender side are different operation parameters. The functionality of both sides is separately encapsulated into the components `BeaconSenderC` and `TdoaMeasurementC`. Both components can run in parallel, but the upper layers should



(a) the incoming pulse is detected very early



(b) the incoming pulse is detected too late

Figure 4.4: Tone detector input and output signals when an acoustic pulse is received. The huge variance of detection sensibility and setup time is shown.

prevent this because the signal detection would be disturbed by sending acoustic pulses while listening. This layer does not initiate a ranging process nor communicates with beacon nodes to start one. It only executes a ranging with predefined parameters. The next paragraphs explain the possible measurement modes, the content of the radio message and the generation of the MAC timestamp, the overall behavior of the system and the concluding statistical analysis.

Operational Modes

To provide flexible operation and future extension, different parameters are introduced as mentioned in Section 3.3.4. The first is the used type of beacon, either a pulse or a sequence. Typically, the type remains the same through out the ranging process but the implementation supports also the dynamic change of type for each separate beacon. This feature can be used to switch the type if too much false detections were made in the past.

The next parameter specifies the timing between single measurements. Either all beacons are sent in a specified time interval or the start of the next beacon is delayed constantly to the end of the former beacon. This parameter must be the same during the complete ranging process. If a fixed interval is taken, the time value has to be communicated between the two nodes by the upper layers. The dynamic interval is especially for the pulse sequence because due to the random intervals of silence between the single pulses the overall duration of the sequence is dynamic. So the time between single measurements can still be minimized. The minimum latency is defined by the maximum transmission duration and the processing time of the receiver node.

The third parameter specifies the termination of the ranging process. Either the number of beacons is predefined by the sender node or the beacons are sent infinitely until the receiver requests the termination of the process. In the case of an infinite amount the `TdoaMeasurementP` module gives an event when a defined number of measurements were successfully executed. The upper layer now must inform the sender to stop the ranging.

Ranging Synchronization Message

The synchronization message fulfills two tasks, first the time of arrival at the receiver as a reference to calculate the TDoA. As the message must be sent once every beacon, it can be used to exchange status information between the nodes. So all parameters mentioned in the last paragraph are transmitted. The current sequence pattern value is also added to the message. This value defines the number of pulses and the waveform for the detection algorithm. Finally, the message contains a counter. It counts down to zero if a fixed amount of beacons is sent. Otherwise, the counter goes up, letting the receiver detect lost packets. Figure 4.5 shows the payload of the synchronization message which is sent through the active messaging radio stack of TinyOS. The pulse sequence pattern needs the most space of the payload, but there is no easy possibility to send variable length messages via the TinyOS radio stack. So the whole payload of 6 Bytes is sent every time.

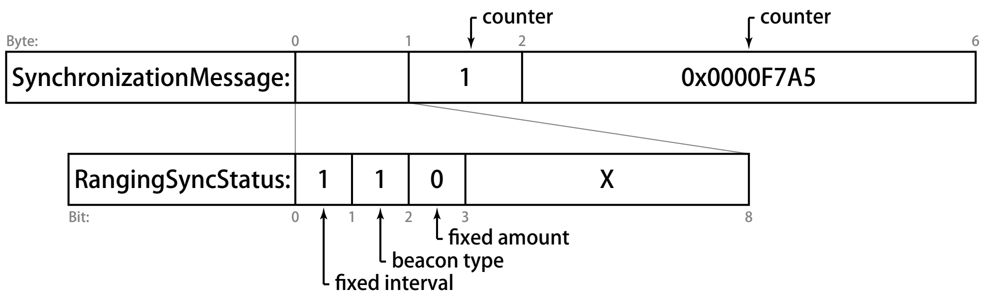


Figure 4.5: Payload definition of the synchronization message that is sent from the sender with every acoustic beacon. The sample specifies an acoustic signal that has a fixed interval, is a pulse sequence and is sent infinitely. The beacon is the first that was sent, and the pattern specifies 5 pulses with the according times of silence in between them.

Ranging Behavior

The two low level components (`BeaconSenderC`, `TdoaMeasurementC`) each provide a single interface to the upper layer. Every interaction is a split phase operation that is started by a command and ended by firing a event. The needed hardware components are initialized and started on demand, so a power control is necessary to the upper layer. The behavior of the ranging process, regardless of what parameters are set, basically stays the same. The `BeaconSenderP` module starts by sending a synchronization message and activates one of the two acoustic beacon generators. If the current beacon is not the last one to send, the node waits a specified time until sending the next beacon. The `TdoaMeasurementP` module waits for an incoming synchronization message. If one is received, the MAC timestamp is taken. Further, the according detector is activated to measure the time difference, which is stored in a result array. From now on, the node is ready to receive the next beacon. After the last measurement, the `StatisticsP` module analyzes the data as described in the next paragraph.

The interfaces distinguish two start commands, one for each beacon amount option. Additionally the split phase control is different between the two modes. Sending infinite beacons needs an additional command to finish the sending process. Therefore, the receiver side fires an event when enough measurements were made. When using a fixed amount of beacons, the sender side automatically returns with an event when all beacons have been sent. Finally, receiver side must be informed to end the measurement and start the data analysis. So for both modes, some information has to be transmitted by the upper layer to execute a complete ranging process.

The timing between the single beacons is made with the virtual timer component from TinyOS, which has more jitter than a hardware timer has but still meets the requirements. The receiver side is very prone to failures. Some of these are impossible acoustic signal detection, the packet loss of the synchronization message, the reception of the next beacon while still calculating the current one or the detection if the sender node was powered down due to hardware problems. A raised error does not abort a current TDoA measurement, but the information gained is integrated into the confidence of the measurement. A perfect measurement has a confidence of 100 and less if errors occurred. The value is calculated according to Formula (4.1), where *success* is the amount of correctly computed TDoA values, *invalid* is the amount of measurements corrupted by noise or too low signal level and *timeout* is the number of occurred timing problems.

$$confidence = \frac{success}{success + invalid + timeout} \tag{4.1}$$

Statistical Data Analysis

The `StatisticsP` module implements the filters to eliminate possible outliers as well as a method to find a mean value from the remaining measurements. The outlier detection is based on the five-value summary mentioned in Section 3.3.4. To detect outliers, a guess for the result must be made to identify them. As discussed the median and the first quartile are suitable for this. Several test series gave as result that the first quartile gives results that are more precise. The quartile is neither prone to false detections nor to echo measurements. As the tone detectors output has a high variance, taking the first quartile as result for the measurement is not acceptable. Therefore, an average value is calculated based on all measurements that are close to the quartile. The inter quartile distance is also used to adjust the earlier mentioned confidence value. If the distance is too high, it can be estimated that the process detected many echoes, which reduces the significance of the measurement.

4.1.3 Ranging Protocol

The last section explained the lower two layers of the ranging program stated in Section 3.1. The third layer is the ranging protocol, which is capable of initiating a ranging process between two nodes by exchanging process parameters and handling start and stop of sending beacons. Therefore, two different protocols implement a state machine for the sender (`ProtocolSenderP`) and receiver (`ProtocolReceiverP`) side. The specification of the radio messages and the operation mechanism is described in the following paragraphs. Figure 4.6 shows the top level of the ranging application as component graph. The receiver side additionally implements the distance model discussed in Section 3.3.5 within the `RangingP` module. The `RangingC` component also combines the functions of the sender and receiver side of the ranging. This makes it possible for a node to act as beacon as well as initiating a ranging process.

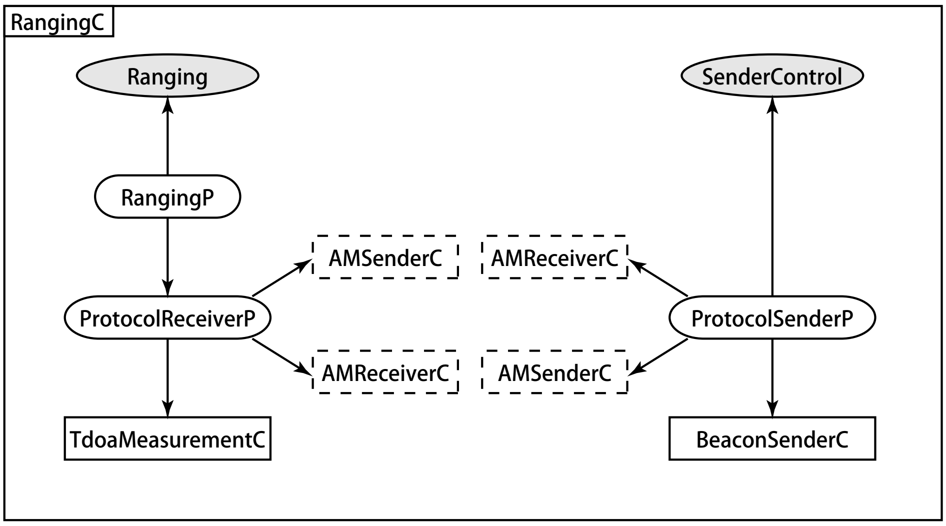


Figure 4.6: TinyOS component graph showing the main parts of the ranging program, using the components for pulse generation and TDoA measurement mentioned before.

Protocol Message Definition

Before the actual TDoA measurement begins the receiver side has to communicate with the beacon requesting the ranging and exchanging the process parameters. A handshake mechanism based on a TinyOS message is introduced. The detailed payload information of this ranging protocol message is shown in Figure 4.7. The first three bits specify the parameters, whilst the ack and stop bits are handling the handshake communication between the two nodes. The mode bit specifies the basic operation mode for the ranging process as described in Section 3.3.6. If mode is zero the direct 1-to-1 ranging mode is executed, implementing the infinite beacon send method. This bit must be the same throughout the whole process because depending on this bit, the according state machine is taken.

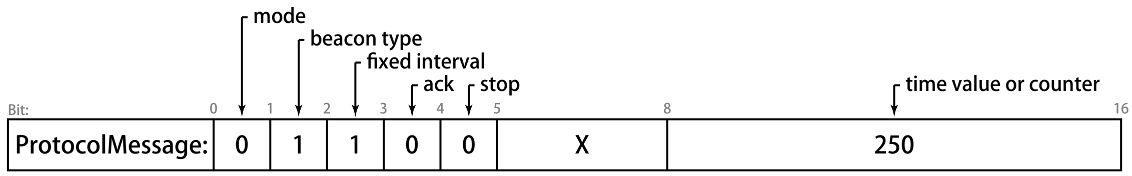


Figure 4.7: Payload definition of the ranging protocol message that is sent between sender and receiver. The displayed message is a start request from the receiver, asking for a direct 1-to-1 ranging of pulse sequences with a fixed interval of 250 ms.

The ack bit is set by the sender node if the request was accepted. If the sender is busy at the moment the ack bit is zero and the time value field specifies the time till the sender is ready to accept a new request. The stop bit is set by the receiver in the 1-to-1 mode to terminate the sending of beacons. If the beacon has stopped, a message with set stop and ack bit is returned to the receiver. A more detailed view on this procedure is shown in the next paragraph.

Direct 1-to-1 Ranging

To initiate a ranging process, the protocol receives the sender node id, beacon type, timing parameters and the amount of beacons to measure. After initializing the microphone, which has a setup time of one second, the request message is sent to the beacon node with cleared mode, ack and stop bit. Afterwards the receiver waits for the acknowledgment message. If no message is received within 500 ms, the message is resent up to three times before aborting the ranging process with an error.

The sender stays idle until a request is received. As an answer to the request the beacon sends a protocol message with cleared mode and stop bit and a set ack bit. If the beacon is busy making a ranging to another node, the request is answered with a cleared ack bit and additional information of how long the process will last in seconds. When accepting the ranging, the pulse generator is started with parameters set in the request message. The generator stays idle for 250 ms before sending the first pulse, allowing the receiver to prepare for the measurement.

After receiving the acknowledgment message, the pulse detector component is activated for receiving the specified amount of beacons. From this moment on the protocol modules of both nodes stays idle until the pulse detector component finishes. The next step is to send a stop request message to the beacon, which is done by setting the stop bit. If no answer is received within 500 ms, the message is resent up to three times.

After receiving the stop request, the pulse generator is stopped and an according acknowledgment message with set ack and stop bits is returned. Additionally the sender includes the amount of sent beacons in the time/counter field of the message payload. From this moment onwards, the sender goes idle being ready to receive a new ranging request.

Until the receiver gets the acknowledgment, the measurement result should be calculated and the protocol passes the information to the distance model. Finally, the microphone is deactivated and the receiver protocol transitions into idle mode.

Distance Model

The implementation of the distance model follows the considerations made in Section 3.3.5. As the temperature for the conversation not only depends on the local value, the ranging component receives this parameter together with the start command. The positioning program is responsible for calculating a value by merging the temperature information of the sender and receiver node.

As postulated in Section 2.3.4, the correction term for the speed of sound is depending linearly from the temperature within the typical operation conditions. So the distance model maps the measured time difference to the distance with a linear equation. The exact model is specified based on an experiment discussed in Section 5.1.2.

Ranging Interface Definition

The ranging program provides two interfaces. The `SenderControl` interface gives the possibility to start and stop the sender protocol and if started, the node accepts incoming ranging requests if possible. The `Ranging` interface is printed in listing 4.1. The command starts a ranging process with only three parameters, the node id of the beacon, the effort in terms of execution time and used power to spend for the measurement process and the temperature used by the distance model. The function returns a failure if the hardware could not be activated or the component already handles a measurement.

After the ranging was completed the component returns with an event giving the beacon node ID, the distance and a confidence value as parameters. Additionally the real effort for the process is stated which can differ from the input parameter because of bad audio or radio link quality. The confidence value also includes an error indicator. If the confidence is zero, an error occurred that could not be ignored or corrected. This can be a wrong or not existent id or a hardware failure of the beacon as well as a very high error rate that makes is impossible to determine a reasonable result.

```
interface Ranging
{
  command error_t startRanging(am_addr_t beacon_id, uint8_t
      effort, float temperature);
  event void rangingDone(am_addr_t beacon_id, float distance,
      uint8_t confidence, uint8_t real_effort);
}
```

Listing 4.1: The basic interface which is provided by the Ranging Application.

4.2 Positioning Software

The whole source code of the positioning software can be found in the folder `positioning` of the source tree. This includes also the adopted source of the liner algebra library Meschach mentioned later [Meschach]. The positioning method is implemented using direct multilateration with a LS solution method to calculate the unknown nodes position. Every node uses only self-gathered data and makes the calculation on its own. Figure 4.8 gives an overview of the involved components of the positioning system. There are three main modules providing the most functionality of the system. These three modules are explained in detail in the upcoming sections.

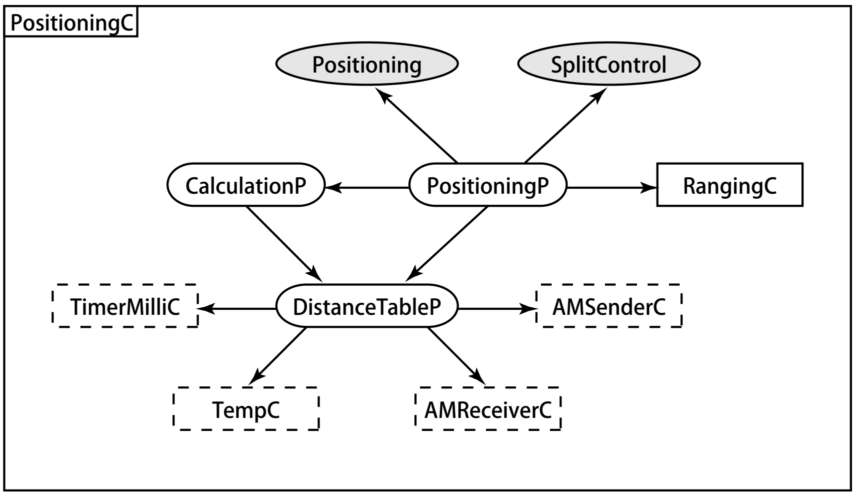


Figure 4.8: TinyOS Component Graph showing the main parts of the positioning program which also utilizes also the ranging component.

4.2.1 Beacon Detection and Data Storage

The `BeaconTableP` module is the central storage facility of the positioning system. This includes the gathered information of beacon nodes, results of the ranging processes and the own positioning information. For that reason, a table containing the beacon and ranging data is maintained. The modules itself fulfills two tasks; the first is the maintenance of the table by analyzing the beacons position broadcasts. Additionally it grants command-based access to all the stored data.

Beacon Position Message

The position message is used by reference nodes to broadcast their own position together with some extra information. The payload of the message is shown in Figure 4.9. The coordinate and temperature values are stored in 16 bit integer variables representing a fixed-point representation of the real value with a factor of 1000. This means the coordinates are stored in millimeters, whilst the temperature is stored in millidegrees Celsius. The maximal coordinate value is around 32 meters, which is enough for this implementation and can be varied for future use by changing a factor in the configuration file `position.h`. The trust value has a similar meaning as the confidence value for the ranging result and gives information about the quality of the position. The value is calculated from the confidence values of the used ranging results and quality parameters of the LS solution. So the trust is more important for the positioning process because is prevents high error propagation.

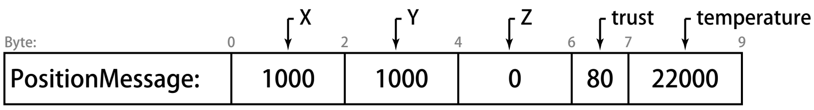


Figure 4.9: Payload definition of the position message that is broadcasted by beacon nodes. The payload specifies an x- and y-coordinate of one meter, a z-coordinate of zero, a trust of 100 percent and a local temperature of 22 degrees.

The beacon table periodically sends out a position message broadcast containing the own position information. This behavior is activated when the node has already a valid position. The messages are sent every 10 seconds as specified in the configuration file. At the same time the table updates the current temperature of the node using the `TempC` component provided by TinyOS. The returned ADC value has to be converted into degrees Celsius. The sensors potential is measured with a voltage divider and can be converted to the current thermal resistance as in Formula (4.2). With the given thermal characteristics [Panasonic] an approximation for the operating range can be made as stated in Formula (4.3).

$$R_{th}\,[\Omega] = R_1 \frac{ADC_{FS} - ADC}{ADC}, \quad R_1 = 10k\Omega, \;\; ADC_{FS} = 1023 \tag{4.2}$$

$$\theta\,[^\circ C] = \frac{10^6}{1307.05 + 214.381 \cdot \ln(R_{th}) + 0,093 \cdot \ln^3(R_{th})} - 273.15 \tag{4.3}$$

Additionally the table has to cope with incoming position messages. The information has to be analyzed and compared with data already stored for this beacon node. So every change of the five values contained by the message has an impact on the future behavior of the positioning system which is explained in the next paragraph.

Distance Table Data

The `DistanceTableP` module holds an array of distance table entries to store all information that is needed to calculate the own position based on ranging and beacon position information. Figure 4.10 shows an example of such an entry together with the field names. Most values were already explained, except the flags byte. The vitality value is a form of time to live indicator for the entry. It is set on every successful reception of an according position message and decremented every broadcast interval. With this mechanism, beacons with hardware failure or bad link quality can be sorted out. The current attempt of a ranging process is saved in the retry field, which is incremented every time the ranging reports an error or has a too low confidence. The selected flag is used by the SVD beacon election mechanism described later.

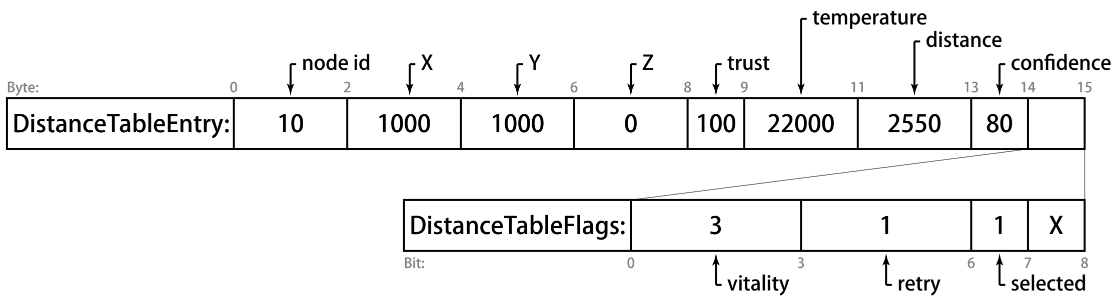


Figure 4.10: Field definitions of a single entry in the distance table with sample values. This entry is for a beacon with the node id of 10 at a position of [1,1,0] meters. The trust of 100 means the node is a reference with a local temperature of 22 degrees. An already made ranging returned a distance of 2.55 meters with a confidence of 80 at the second attempt. The beacon is selected as input for the current SVD calculation.

Incoming information from position messages is inserted into the table. If no entry exists for the beacon id, a new is created. If no entries are free, the entry with the lowest trust value is replaced. A new ranging result can only be inserted, if the entry already exists. When a position message contains differing coordinate values, these are updated and an already made distance calculation is deleted by setting the confidence to zero.

Finally, the table is capable of receiving radio messages setting the coordinates of the node. This method is used by quasi-stationary reference nodes to program or update the current position. The information is only stored in volatile memory and has to be received in every power cycle. The reception of a coordinate message sets the trust to 100, which makes the node a reference station too.

The `DistanceTableP` module provides two interfaces. The first provides access to the data stored in the table and also to the currently set position of the node itself. Additionally the interface has some library functions for the calculation process that make heavy use of the table data. The other interface contains events that are fired on table entry changes, for example on entries that are created or deleted. As for memory constraints the table size is predefined to 6 entries in `position.h`.

4.2.2 Position Calculation

The position calculation is implemented in the `CalculateP` module and uses the data provided by the distance table as shown in the component graph of Figure 4.8. When a calculation is started, the selected beacon data is converted by the linearizing tool, then the SVD is started and finally the LS solution is calculated (Section 2.4.3). If high errors can be detected in the solution, the process is repeated with other parameters. To execute the mentioned algebraic features the next paragraph explains the adaption of a library for the target processor architecture.

Linear Algebra Library

There are many libraries available for matrix computations but to ease the adoption, only C implementations were considered. The most common CLAPACK framework runs pretty well on PC class processors but cannot be adapted to the AVR architecture. SVDLIBC is another library especially for solving a SVD but heavily relies on dynamic memory allocation which is not possible for micro controllers. Finally the Meschach framework suited best, although some rewriting has to be done [Meschach]. The framework is only adopted to solve either a matrix inversion or a SVD which also includes most of the basic vector and matrix operations. All the source files including the altered library header reside in the `meschach` sub folder of the source tree.

The library also uses some temporary objects that originally were created by dynamic memory allocation. These objects are given as additional parameters in the adopted version of Meschach. A slightly negative impact of using an algebra framework instead of a specific SVD solution method is the code size, which is increased by 20 kB. Nevertheless, the library is significantly smaller than the other choices available.

Linear Least Squares Solution and Error Detection

The calculation process is initiated by the `PositioningP` module. The first step is to elect the beacons that are used to calculate the position by defining a threshold for trust and confidence. The distance table now marks all adequate entries by setting the selected flag.

The calculation process of the pseudo inverse needs some temporary matrices, which have to be defined statically in TinyOS and therefore permanently allocate memory. When using an input matrix with a rank of maximum 6, the over all needed memory is more than 500 Bytes. There has to be found a trade off between consumed memory, computational effort and the improvement in precision for the calculation. Due to the linearization tool mentioned in Section 2.4.3, the input matrix rank for the SVD is one less than the amount of considered reference nodes. Therefore, the theoretical minimum of four beacons results in a matrix with a rank of 3. To achieve modularity the maximum rank and the resulting memory consumption can be easily configured during compile time.

If less than the needed four beacons are found an error is reported, the thresholds are reduced and the process is restarted.

The next step is to prepare the input for the SVD with the data gathered from the distance table. Next, the generated matrix is tested for special cases of singularity. If a single coordinate is equal for all selected beacons the SVD still gives a result which is only two-dimensional. If the test is passed, the decomposition itself is executed.

Afterwards the pseudo inverse of the decomposed matrix is calculated and the equation system is solved according to Formula (4.4). The key feature is the easy calculation of the pseudo inverse of Σ, which is a diagonal matrix of the positive singular values of A. The inversion is done by replacing every value by its reciprocal. The resulting position is then checked for precision according to Formula (3.2). If the accumulated error is higher than one meter, the beacon is ignored and the calculation is redone following the strategy of Section 3.4.2.

$$A \xrightarrow{SVD} U \cdot \Sigma \cdot V^T \qquad A^+ = V \cdot \Sigma^+ \cdot U^T \qquad x = A^+ \cdot b \tag{4.4}$$

4.2.3 Positioning Strategy and Protocol

The `PositioningP` module is responsible for coordinating the positioning process. This is done by providing the interfaces for the user application, initiating the ranging processes, controlling the distance table and activating the position calculation. After starting the positioning system, the table is activated to receive data of the surrounding beacons. Once a positioning request is received from the application, the table is checked for enough beacons. If there are at least four beacons present, the ranging starts by executing a distance measurement to every node of the table. If some results did not have high enough confidence values, these rangings are repeated up to two times as defined in the configuration file. The retry flag of the table entry (Figure 4.10) is used for coordination. The temperature value for the distance model is calculated by averaging the values of local and remote temperature.

If no more rangings have to be made, the calculation is started with a minimum trust of 100, which means only reference nodes are considered. The minimum confidence is set to 80 at first and is only reduced if the calculation resulted in a singular matrix or the error was too high. The application informed before the calculation is started. This is because the SVD algorithm is not interruptible and executed in one task which could affect other software components. After completion, the positioning result is returned. As the node now knows the own position, the beacon table is set to broadcast this information turning the former unknown node into an assistant node as stated in Section 3.2.

Positioning Interface Definition

The top level `PositioningC` component provides two interfaces, one of them is a default `SplitControl` interface already defined by TinyOS. The start and stop commands are redirected to the distance table and the ranging receiver. If the table already contains a valid position, the broadcast is activated. Finally, the `Positioning` interface, displayed in listing 4.2, gives access to the positioning functionality. A previous activation of the component is necessary before every call of this interface.

The `getPosition` method returns the currently set position including trust and temperature or null if no position was calculated. The second command starts a positioning process. If the force ranging parameter is true, every existing ranging is redone except for those with high confidence. Otherwise all available ranging information is reused. The effort value stands for the same as already mentioned in the ranging interface, the effort taken in time and power consumption to retrieve a result. There is also the possibility to invalidate a currently set position, if an application recognizes a displacement of the node. The `interruptPositioning` method tries to end a running process as early as possible. Even this method cannot stop the SVD task during runtime, because this is not possible in TinyOS. Finally, there is also a possibility to set a new position with the `SetupPosition` command. This automatically turns the node to an assistant, activating the broadcast mechanism.

After the start of a positioning, two events are always fired in distinct order. Before the LS solution is made, the `startingCalculation` event is fired with an estimated execution time in milliseconds as parameter. At the end of the process, the new calculated position including trust and temperature is returned as parameter by an event. If an error event is raised, the whole component transitions to idle state. The different error codes are listed as enumeration in the `position.h` configuration file.

```
interface Positioning
{
  command PositionMsg* getPosition();
  command error_t startPositioning(bool force_ranging, uint8_t
       effort);
  command error_t invalidatePosition();
  command error_t interruptPositioning();
  command void setupPosition(uint16_t x, uint16_t y, uint16_t
     z, uint16_t trust);
  event void startingCalculation(uint16_t estimated_time);
  event void positionCalculationReady(PositionMsg *position);
  event void error(uint8_t code);
}
```

Listing 4.2: The basic interface which is provided by the Positioning Application.

Chapter 5

Evaluation and Experiments

This chapter presents the tests of the implementation that are made to evaluate the performance of the localization system. As the ranging and positioning implementations can be taken as separate systems, the chapter splits into two sections. First, the performance of the acoustic ranging is evaluated. Second, the results of the positioning system simulations and experiments are presented.

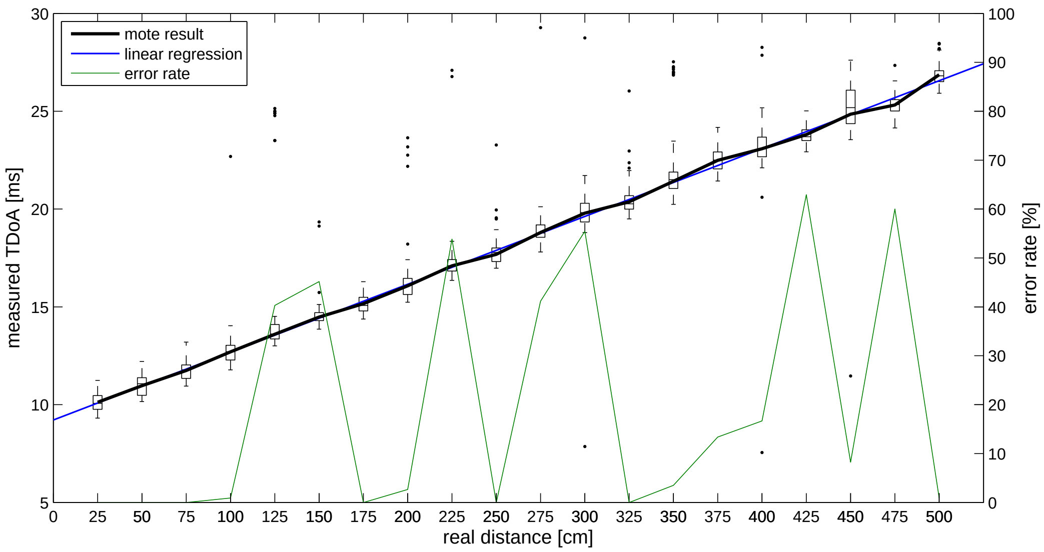
5.1 Acoustic Ranging

The acoustic ranging process is analyzed in different stages. The first section covers all experiments concerning the TDoA measurements. This involves the overall performance, as well as the analysis of the PSD algorithm. Further the characteristics of the acoustic channels' analog signals are presented, followed by an evaluation of the unit-to-unit variation. The next section describes the process to define a distance model that transforms the measured time into an actual distance in meters. Based on the last tests, the next section presents the overall performance of the acoustic ranging system, followed by the results of an experiment under real operation conditions. Finally, a power analysis of the ranging component is presented.

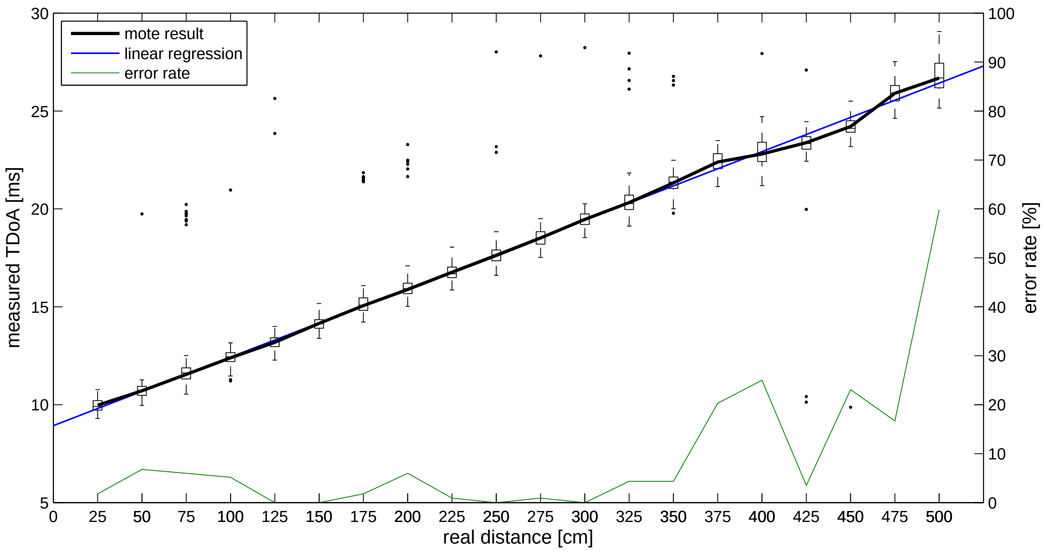
5.1.1 TDoA Measurement Analysis

The first experiment tries to analyze the performance of the TDoA measurement process. The goal is to evaluate the measurement error and distribution for both beacon modes at different distances. The test setup involves two nodes, one fixed node acting as beacon sender and the receiver node is positioned at different distances from 25 centimeters up to 5 meters. The nodes are deployed on the floor facing toward. The real distance between the two nodes was checked by a measuring tape and introduces an error of less than one centimeter. A single ranging process consists of 100 TDoA measurements to gain enough ranging data for a statistical analysis.

To perform the test, the default ranging program was extended to provide additional debugging information. After each distance measurement, a radio message is sent to the base station consisting of the result and a possible error code. After the whole process, the

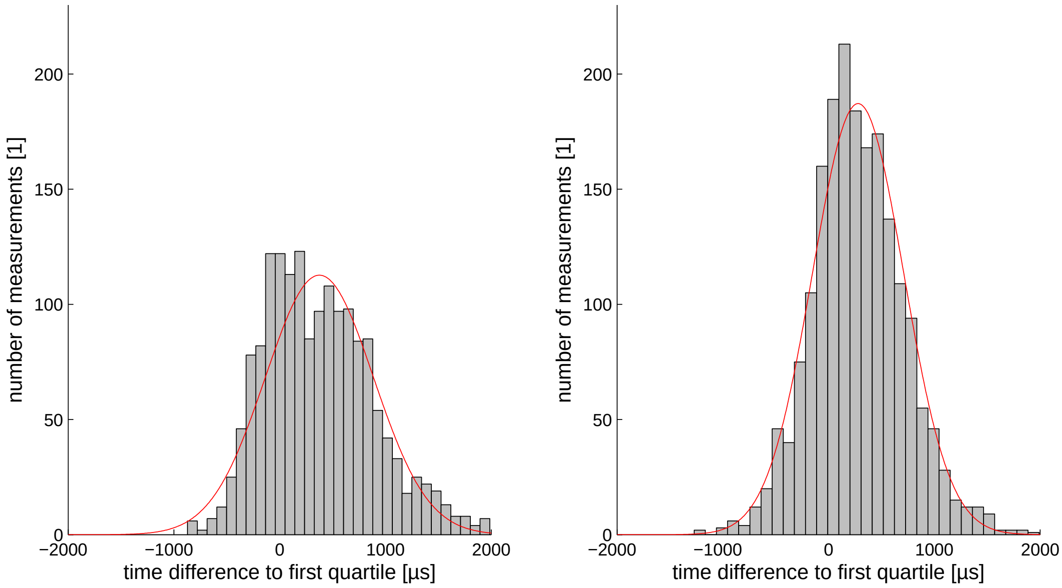


(a) Measurements with single beacon pulses



(b) Measurements with pulse sequences (5 chirps)

Figure 5.1: TDoA measurement results over the real distance. The single measurement deviations are displayed as box-plots for every experiment. The black dots represent outliers mainly caused by echo detections. The black line represents the result of the statistical data analysis and the blue plot represents a linear regression of these results. The second y-axis shows the error rate of the measurement. Errors occur when no signal detection can be made and are less frequent with sequence sequences.



(a) Measurements with single beacon pulses (b) Measurements with pulse sequences (5 chirps)

Figure 5.2: Shows the TDoA measurement result deviation from the corresponding first quartile of the measurements also presented in figure 5.1. As expected the error is nearly Gaussian distributed and is lower for the sequence approach.

result is transmitted together with the confidence value. Figure 5.1 shows the results of the test for both beacon types. The linear characteristic of the results can be seen as postulated by theory. The graph also shows a box-plot for every measurement to give information about the precision. A high precision is represented by a small inter-quartile-distance. The difference between the linear regression and the measurement results is the accuracy of the measurement system. The accuracy slightly drops at high distances because of the lower acoustic signal level whilst the precision nearly stays the same. This proves the presumption that the hardware latencies have the highest impact on the precision and remain constant over the measured distance. The measurement error rate is also included in the graph. It is lower with pulse sequences and rises with increasing distance because the signal amplitude at the receiver is lower.

Figure 5.2 shows the error distribution of the distance measurements and proves the concept of the implemented statistical data analysis described in Section 4.1.2. The data for the histograms is filtered for outliers and is relative to the first quartile of the single distance measurements. Especially the sequence data is Gaussian distributed which makes it possible to gain higher precision by averaging the results of successively taken measurements. The mean value is only slightly above the first quartile. This fact makes it possible to detect the outliers based on the first quartile value reducing the impact of echo measurements as postulated in Section 3.3.4.

The increase of SNR when using pulse sequences (Section 2.3.3), can also be seen in the two histograms. The standard deviation decreases from 511 μs in the beacon mode to 431 μs when using 5 pulses per sequence. The air temperature was observed during the whole experiment and was constant at 23 degrees. This eliminates the influence of the speed of sound to the measurement error.

PSD Algorithm Performance

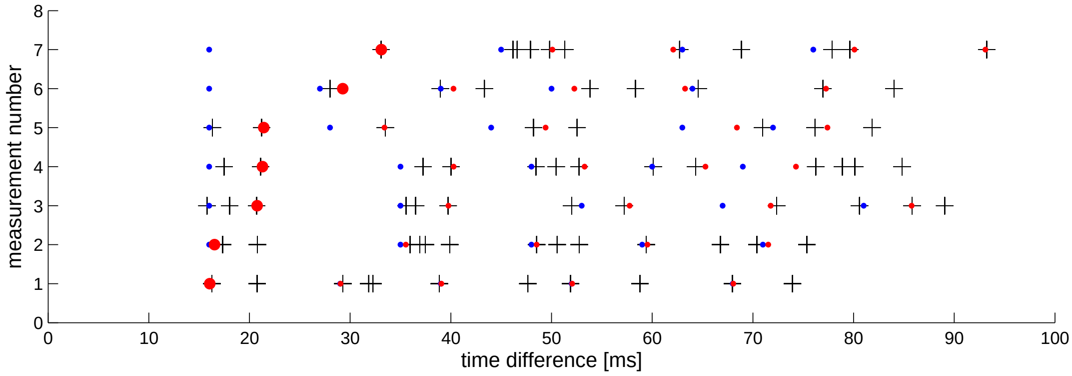
The PSD algorithm is the key feature to improve the measurement results in case of environmental noise or false detections. The design of the algorithm itself is coupled tightly with recorded experimental test data and the basic functionality is already described in Section 4.1.1. To gain the SNR benefit for the sequence as formulated in Equation (2.6), the standard deviation of the selected pulse detections to the known sequence has to be calculated and considered in the result.

Figure 5.3(a) shows an analysis plot of the algorithm performance. The x-axis represents the time measured since the reception of the radio synchronization message, whilst the y-axis represents the number of the measurement. Every line of points stands for exactly one PSD, where the black crosses mark all the detected pulses. The small blue points represent the pulse sequence searched for at the correct TDoA. The first pulse of the detected pulse sequence is marked with the bigger red point whilst the other pulses are represented by the small red points. A perfect solution would be an overlay of the first blue point with the big red one. Measurement number one of Figure 5.3(a) shows a nearly exact pattern matching. The second one shows the correct impact of the standard deviation to correct the too lately detected first pulse. The next two calculations are false because in each case one pulse of the signal was not detected. However, the echo signal was detected better and thus taken as result. Measurement 5 is a more drastic sample because even three of the five pulses ware not detected correctly. The overall result would still be correct because the first quartile is taken to eliminate outliers but the algorithm is still very vulnerable especially for long distance rangings with low acoustic signal levels.

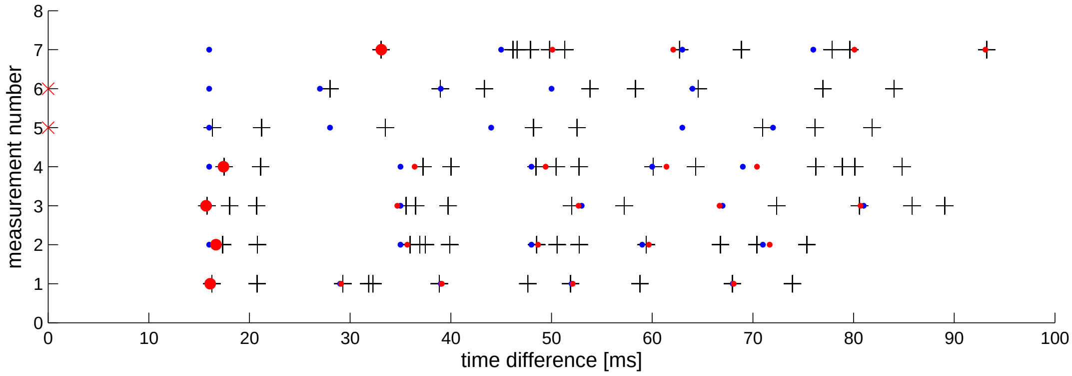
So the major problem for the detector are received echo signals, which often have a much higher signal amplitude than the original signal. So the algorithm has to be optimized to ignore these false detections. This is primarily done by preferring earlier correlation results with nearly the same error. Later detections must have a significantly lower failure to be accepted. In the implementation, the algorithm stores the two best hits. At the end, the earlier value is taken regardless of the introduced failure. The introduction of a maximum error is also necessary to ignore the detections with very high variance of the single pulses. The accumulated error must not succeed a defined threshold.

Finally, the algorithm is made resistant to one undetected pulse of the sequence. The first pulse for which no corresponding detection could be found is being ignored. The current deviation is increased to still prefer correlations where all pulses are detected. Theoretically, the number of maximum ignored pulses can be increased but this mechanism also makes the whole algorithm prone to false detections from environmental noise.

The optimized PSD algorithm succeeds better results as seen in Figure 5.3(b). Measurements 3 and 4 are now detected correctly. The next two pulse sequences were detected badly so the algorithm aborted the operation. This behavior is more useful because invalid



(a) Algorithm with low constraints and standard deviation correction only.



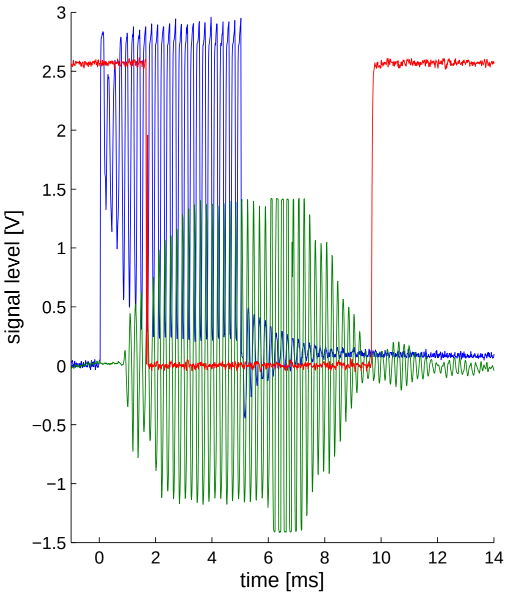
(b) Advanced Algorithm with high error constraints and tolerance to not detected pulses.

Figure 5.3: This two figures give a visual representation of the PSD algorithm results. The interrupt times, represented with black crosses, are aligned horizontally for each distinct measurement. The pulse pattern that is searched is represented by the blue points, whilst the red ones mark the results of the algorithm. A red cross indicates that the algorithm was unable to detect the sequence.

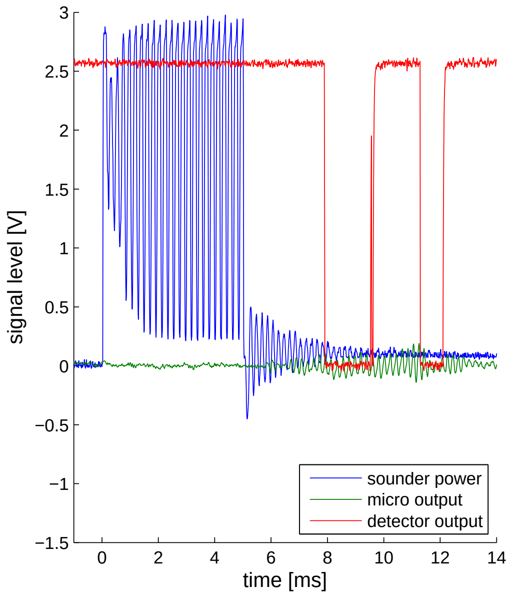
detections cannot corrupt the quartile calculation. At measurement seven, the algorithm still detects the echo signal. A drawback of the improved code are false positive decisions, but it is still better to ignore a measurement than to risk echo detection. Advanced test data resulted in an abort rate of 24 percent on average. The remaining sequence detections had a success rate of 50 percent. The basic algorithm failed in more then half of the measurements with a mean success rate of 29 percent and hence getting an incorrect first quartile for outlier detection.

Acoustic Signal Detection

A further experiment is to get knowledge of the analog signal flow concerning the buzzer and microphone circuitry including the tone detector. The most significant signals are the power supply pin of the buzzer, the output of the microphone amplifier and the tone



(a) Measurement at a distance of 10 cm.



(b) Measurement at a distance of 2 m.

Figure 5.4: The scopes show three signals of the acoustic channel at sender and receiver side. The tone detector correctly finds the beacon signals in both cases although the signal amplitude at 2 m distance is very low.

detector output. Figure 5.4(a) shows such a measurement at an inter node distance of 10 centimeters. The acoustic delay can obviously be seen in the scopes. This delay between sounder and amplifier consists of the propagation delay and the hardware latencies introduced by the mechanical inertia of the sounder and microphone membranes.

The measurement is done with a digital oscilloscope with two involved nodes. The receiving node is equipped with an adapter to get access to all pins of the expansion connector. The detector output is measured at the INT3-pin, whereas the microphone output is taken from the ADC2-pin of the connector. The two nodes' ground is combined and the sounder output is taken from the M-pin of the device.

Additionally the sounder needs some startup time before reaching the maximum output power, which can also be seen in the waveform of the amplified signal. The delay of the PLL circuitry inside the tone detector can be seen as delay between received signal and the external interrupt signal of the micro processor, and is approximately 0.75 ms. Finally an echo effect can be seen as overlain signal starting at approximately 5 ms in the time line. This signal summation is constructive and therefore increases the overall amplitude.

Figure 5.4(b) shows the same measurement at a distance of 2 meters. The received acoustic signal has a remarkably low amplitude. The scope would include less noise if captured after the band pass filter but the pin cannot be measured without hardware modifications. After all this manifests the reason why the pulse detection success rate decreases at distance of more then 3 meters.

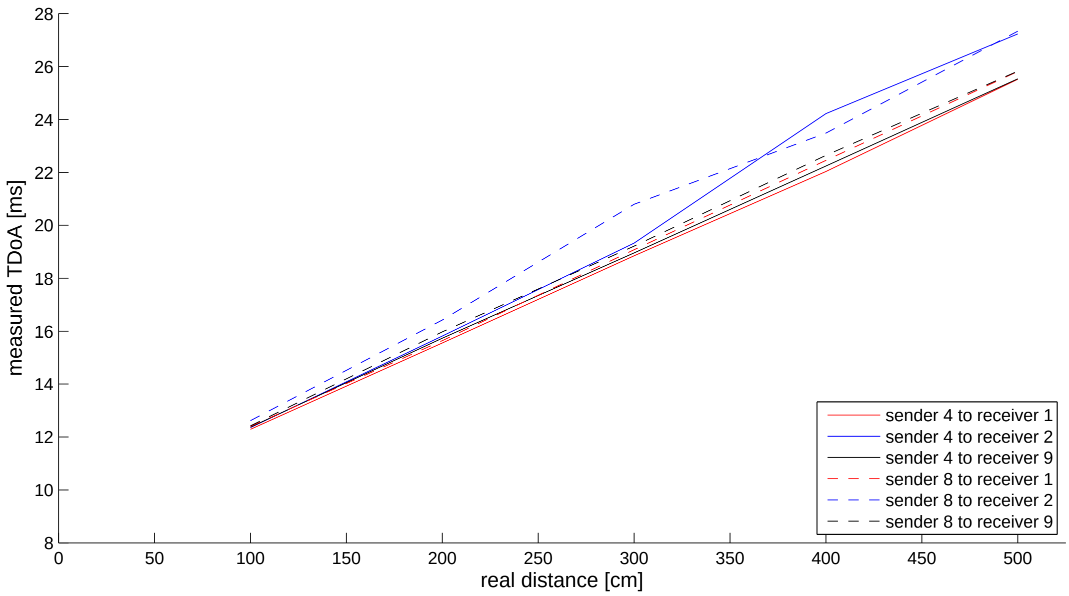


Figure 5.5: Compares the results of ranging measurements from different nodes acting as sender and receiver. Receiver 2's sensor board introduces a high diverge to the other measurements.

Unit to Unit Variation

The presence of hardware tolerances between different nodes cannot be avoided easily. It is necessary to obtain test data to specify the possible impact on the results of the ranging process. To keep the effort in limits two sender and three receiver nodes were taken into account to evaluate unit-to-unit variation. The test setup is similar to Section 5.1.1 but only a few distinct distance measurements are made. The TDoA over distance plot for every sender-receiver combination is show in Figure 5.5. Obviously, two combinations have a high discrepancy to the other curves. Both involve the receiver with ID2. As the according sender nodes have highly correlating results to the other two receivers, node 2 can be marked to deliver corrupt results.

For the complete testing period, each node is connected to a distinct sensor board. These combinations are never changed as long as not specified otherwise. To narrow down the source of error from node 2, the sensor board is exchanged temporarily with the board of node 1. Additional measurements make clear that the source of error is purely caused by the sensor board. After exchanging the boards, node 1 also delivers inaccurate results but node 2 measures the distance correctly now. So the error must emerge from hardware components of the acoustic receiver circuit. As this thesis cannot cope with these additional parameters, the appropriate sensor board is left away for future experiments.

To introduce the error caused by unit-to-unit variation in the final distance model building process, the test data has to be averaged. The difference between the mean unit variation and the reference combination has to be considered later by building the distance

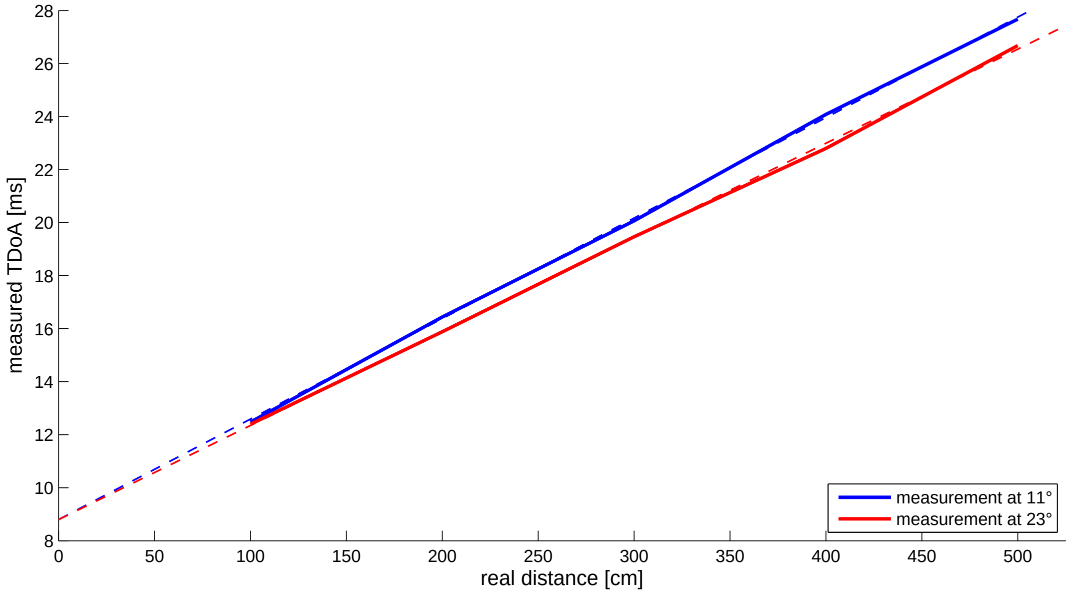


Figure 5.6: Two TDoA measurement series at different environmental temperatures to gain additional information for forming a distance model to convert measured time difference into a distance in meters.

model. The reference combination is node 8 as sender and node 9 as receiver, which are used in all other experiments of this section. The deviation also has high impact on the precision of the whole acoustic ranging process.

5.1.2 Distance Model Specification

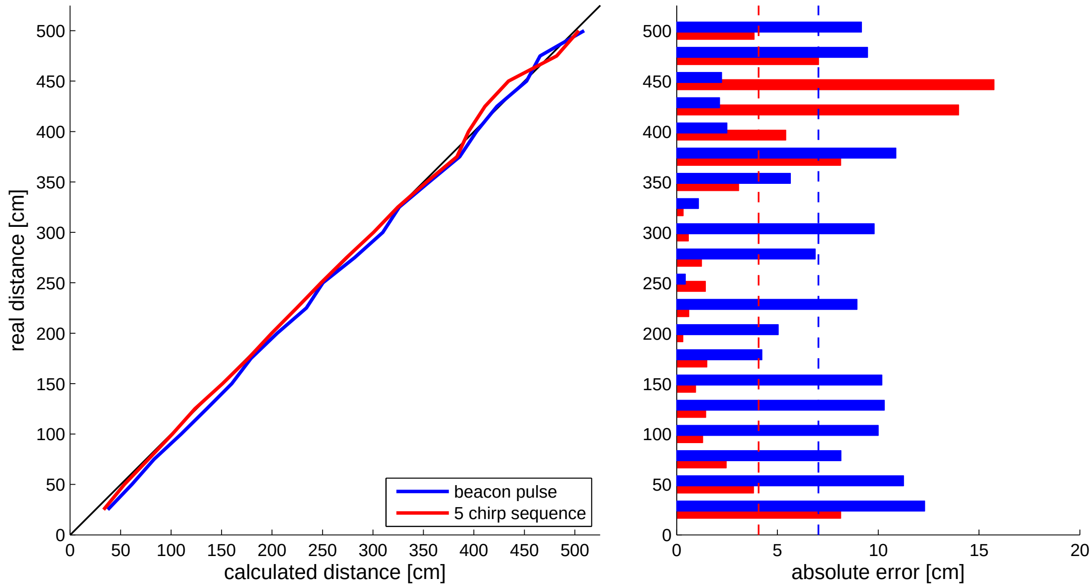
This experiment mainly should validate the speed of sound given by means of physics. As the results will not match perfectly, the model calculated on top of this test will be used in the ranging software to convert the measured time difference to a distance value. As the environmental temperature has most impact on the speed of sound, the experiment consists of two measurement series at different temperatures. The linear regression of both series should have an equal offset in time with different gradients. Figure 5.6 shows both measurements as TDoA over distance plots where every data point is averaged from 100 measurements. The graph shows both mentioned behaviors and can therefore be used to gain the model parameters.

$$regression\ for\ 23^{\circ}C : d_{23} = 8798\,\mu s\ ,\ k_{23} = 3549\,\mu s/m \tag{5.1}$$

$$regression\ for\ 11^{\circ}C : d_{11} = 8738\,\mu s\ ,\ k_{11} = 3803\,\mu s/m \tag{5.2}$$

$$model\ parameters : d = 8798\,\mu s\ ,\ k_0 = 4013.6\,\mu s/m\ ,\ k_T = -20.20\,\mu s/m/^{\circ}C \tag{5.3}$$

The three parameters are the constant offset in time at the distance of zero meters (d), the gradient at a temperature of zero degrees Celsius (k_0) and the linear speed dependency



(a) Final results of the ranging process. (b) Absolute error of the calculated results.

Figure 5.7: Two plots showing the accuracy of the ranging process. The precision information can be estimated from the result's deviation.

from temperature as gradient variation per degree (k_T). Equations (5.1) to (5.3) show the results of the linear regressions, and the value of the parameters.

5.1.3 Ranging Accuracy

The goal of this evaluation is to check the distance model and specify the final accuracy and precision of the acoustic ranging process. The test data from Section 5.1.1 and the specified model from the last section are taken into account. The transformation is calculated in MATLAB, but the results are the same because the nodes are provided with exactly the same model parameters. The results of the transformation are presented in Figure 5.7(a). The black line displays the ideal transformation, so the calculated distance is overestimated if the plots are below this line.

The accuracy of the ranging system is shown in Figure 5.7(b) as absolute error between calculated and real distance. As postulated, the accuracy of the pulse sequence is lower than the beacon pulse ones. This fact was already observed in the result distribution of the TDoA measurements in Section 5.1.1. The average error shown in the figure is an optimal value because parts of the data were taken into account for the model generation. So the overall accuracy will be lower because the unit-to-unit variation and unknown parameter variances introduce slightly more error than what is shown in the figure.

During the last evaluations, several measurements series ware captured. Together with some additional series with different topologies, nodes and parameter settings a final error rating can be given as an average of all series. This final accuracy is $8.75cm$ absolute or

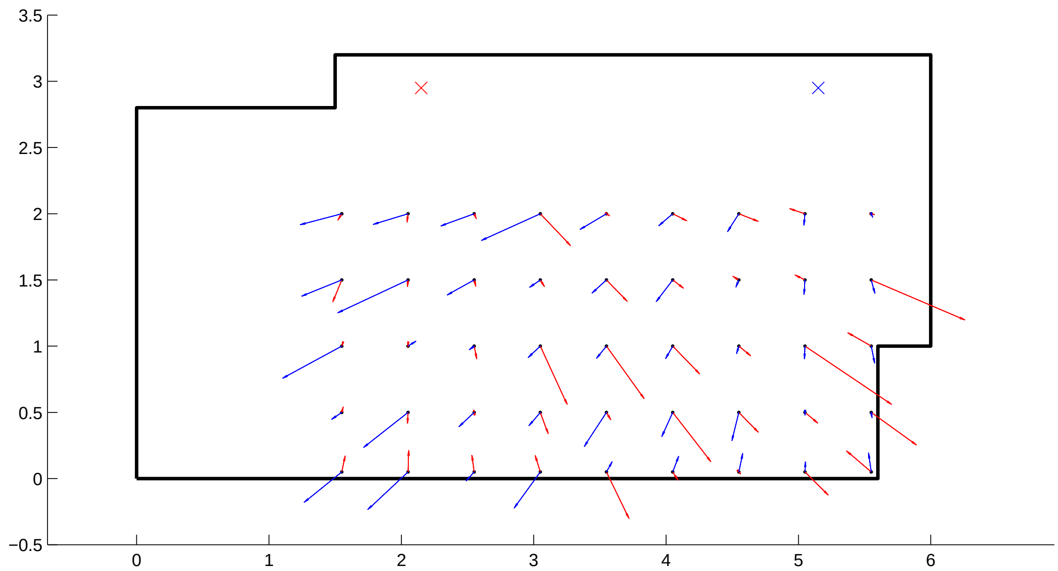


Figure 5.8: Ranging test along a grid in a vertical plain of the laboratory room with two differently positioned reference nodes marked with a colored cross. The ranging error is shown as force vector toward's the direction of the according beacon.

2.75% relative error to the measured distance.

5.1.4 Test in a Laboratory Environment

A final experiment to gain knowledge about the error for the ranging process is described in this section. The laboratory room was intersected by a vertical plain and divided into a grid of approximately 50 centimeters. Two reference nodes are mounted on the ceiling with the sounder directed toward the floor, while the receiving node was mounted upwards on the head of a tripod.

The ranging parameters for the experiment were set to values as used in a real world scenario. So every ranging consists of only 20 TDoA measurements with a 5 chirp pulse sequence in a fixed interval of 250 ms. The ranging process is aborted if the PSD can not succeed in more than ten measurements. The receiving node was positioned with perpendicular and measurement tape providing less than 5 cm of displacement to the targeted point of the grid. The results are presented in Figure 5.8 by drawing the measurement errors as force vector from the real position toward the beacon node. Approximately 15 percent of the measurements resulted in echo detections and were repeated. The error prone points mostly are in the area outside the signal cone of the speaker where the received acoustic signal strength is much lower.

As seen in the graph, most of the rangings are over estimated. This is especially true for measurement points outside the earlier mentioned direct speaker cone. This behavior comes from the orientation of the sounder to the receiver and is therefore a variation of the received signal strength. Lower signal amplitudes result in a higher hardware latency of the tone detector as discussed in Section 2.3.4. The reduction of signal strength with higher distance is included in the distance model, but the orientation dependency cannot be considered because it is unknown. In the last tests, the measurements were taken on the floor facing toward, which results in higher signal amplitudes at the receiver. In the current experiment the acoustic energy is emitted at a wider angle, which gives a lower signal amplitude at the receiver. This introduces an overestimation because of the longer tone detector setup time. The evaluation of the measurements did result in a standard deviation of 20 centimeters with an overestimation of 14 centimeters.

5.1.5 Ranging Power Measurements

A final and very important task for WSN development is the power characterization of the software. To gain exact power dissipation data, a test setup especially for Mica 2 nodes is present at the institute. The error of the resulting power values is bounded by a maximum of 2.1 percent [Glatz10a]. Table 5.1 lists the most important power states measured with the setup that are necessary to characterize the sender and receiver behavior. Most power is consumed by the radio even if the module is in idle listening state. The values for sending and receiving pulses is a mean value over the whole process, which is around 50 ms for a pulse and 125 ms for a 5 chirp sequence.

State	**Power**
idle with activated radio	53.83 mW
during radio send process	65.07 mW
during radio receive process	53.98 mW
sending a beacon pulse	57.06 mW
sending a 5 chirp pulse sequence	55.88 mW
receiving periodic beacon pulses	56.50 mW
receiving periodic pulse sequences	56.47 mW
activated buzzer	58.63 mW
activated microphone circuit	56.56 mW

Table 5.1: Different power states necessary for the ranging program. The power dissipation is a mean value during the execution and does not depend on the states' duration.

The power measurements for the receiver side are not that meaningful because the values do not differ to the power value of only the microphone turned on. Another remarkable point is the power consumed during message reception. This value is only slightly higher than the idle listening value of the module. So the fundamental power states for the ranging process are radio on or off, microphone amplifier circuit activated or not and the usage of the buzzer. The power values of sending a pulse or sequence are different, because the activity rate of radio and buzzer are higher during sending a pulse.

By analyzing the power profiles, the needed energy for certain tasks can be evaluated and the results are summarized in Table 5.2. With these values, the approximate energy

Task	Avg. Energy	Avg. Duration
sending a 5 chirp sequence	6.98 mWs	120 ms
sending an additional chirp	0.83 mWs	15 ms
sending a beacon pulse	2.85 mWs	50 ms
sender protocol overhead	9.83 mWs	150 ms
activating microphone circuit	63.5 mWs	1100 ms
receiving a 5 chirp sequence	7.06 mWs	120 ms
receiving an additional chirp	0.85 mWs	15 ms
receiving a beacon pulse	2.82 mWs	50 ms
receiver protocol overhead	26.7 mWs	450 ms

Table 5.2: Energy needed for certain tasks of the ranging process. The values are absolute ratings representing the energy needed for the whole node. The protocol overhead is the energy needed for inter node communication and measurement preparation.

needed during a ranging process can be obtained. For example a ranging with 5 chirp pulse sequences, 16 valid measurements and a fixed interval of 175 ms needs the following energy. The sender program runs between 3 and $6.2s$ and needs 160 to $340mWs$ of energy. The receiver side has an execution time between 4.4 and $8.2s$ with a energy requirement of 250 to $460mWs$. The high variance of the values mostly comes from the radio link quality. If the connection is bad, the protocol messages have to be resent until a valid acknowledgment is received. The receiver generally needs 1.1 seconds more time to execute, because in this time, the microphone circuitry is activated. The time is fixed by TinyOS, but could possibly be lowered for energy optimization.

To give visual feedback, Figure 5.9 shows the power dissipation of a sender during the generation of a 5 chirp pulse sequence. The plot differs to the one of Section 2.5 because the data was filtered with a low pass at $f_g = 2000Hz$ to average the power peaks of the radio module. The filter also equalizes the power dissipation of the sounder that is oscillating at $4.1kHz$.

Summary

In the following paragraph, the most important findings of the last evaluation section are summarized:

- TDoA measurement succeeds perfectly for distances up to three meters, above the success rate drops dependent on the environment to under 50 percent.
- The measured time difference is nearly linear dependent from the real distance.
- The resulting standard deviation of the measurements is lower for the pulse sequence approach as postulated in literature and concept.
- The detection algorithm for the pulse sequence is the key element for high success rate and accuracy. Precision is mainly increased by multiple measurements.

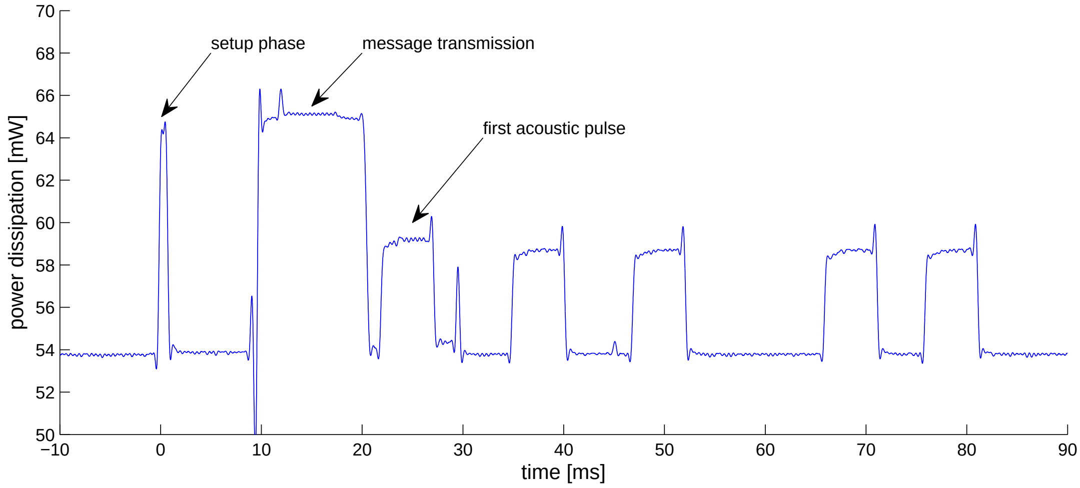


Figure 5.9: Shows the low pass filtered power profile from the used measurement platform during a TDoA ranging process using a 5 chirp pulse sequence. Most significant are the random pattern generation and setup, the transmission of the radio message and the five activations of the buzzer.

- For the PSD algorithm, echo signals are much worse than random false detections from environmental noise.
- The output power of the sounder has to be increased especially for indoor environment to increase the maximum possible range or success rate for long rangings.
- Unit-to-unit variation is less than expected, but erroneous sensor boards produce completely false results and should be sorted out.
- The distance model can be specified effectively from test data measured at two different environmental temperatures.
- Acoustic ranging has an accuracy of 8.75 centimeters or 2.75 percent at distances up to 5 meters.
- A typical ranging takes approximately 5 seconds with a required energy of 200 mWs for the sender and 300 mWs for the receiver respectively.

5.2 Position Calculation

The second evaluation section covers all experiments for the positioning software. In a first step, the implemented multilateration process is simulated. MATLAB is used as simulation environment, which starts the calculation module that is compiled in Ansi C. The section lists the simulation results of two and tree-dimensional coordinate systems, followed by a discussion over the impact of ranging outliers in the position calculation. A final laboratory experiment evaluates the performance of the whole localization implementation. The final section presents results of the positioning power analysis.

5.2.1 2D Algorithm Precision with Generated Distance Data

To observe the performance of the multilateration algorithm, the first step is a simulation based on the distance data gathered in the last sections. The compiled program is a branch of the original version in NesC altered to support 2D coordinates and parses the ranging information from a file. The simulator generates distance values according to the error distribution formulated in Section 5.1.1 and 5.1.3.

Figure 5.10(a) shows a map of the results. Black dots specify the beacons position, the red crosses are the unknown node positions and every blue dot stands for the result of a multilateration process using a LS solver. The results cumulate around the real position and the precision is nearly the same for all node positions. The deviations of the results' coordinates is normally distributed and already lower than the standard deviation of the distance data.

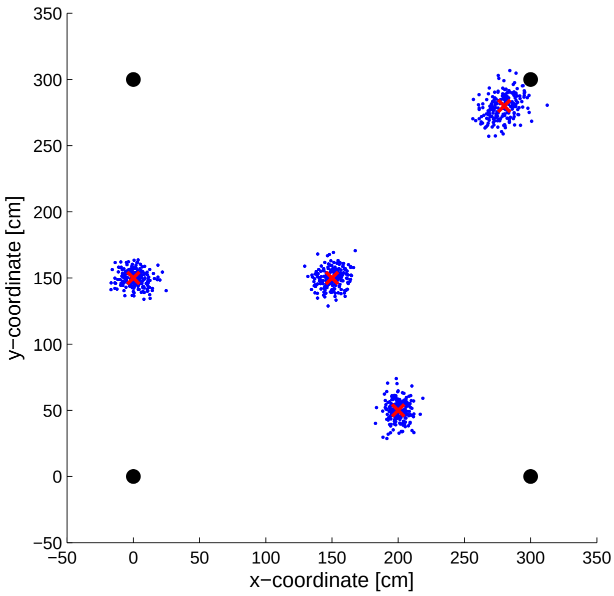
The positioning error is defined as Euclidean distance between real and calculated position. The error distribution of all 800 simulation runs is shown in Figure 5.10(b). The data is no longer normal but instead Chi distributed [Saxena05] with two degrees of freedom, making it identical to the Rayleigh distribution. The four node positions are taken by random to gain a result that is not directly dependent to the beacon topology. With a distance error deviation of 8.75 centimeters, the two resulting coordinates are distributed with a standard deviation of approximately 7.8 centimeters each. As the mean of the Rayleigh distribution is defined as $1.256 \cdot \sigma$ the resulting positioning error is 9.8 centimeters on average. The cumulative distribution function reaches 95 percent for an error of twice the mean value. In other words the error of the multilateration is less than 20 centimeters in 95% of the calculations.

5.2.2 3D Algorithm Precision with Generated Distance Data

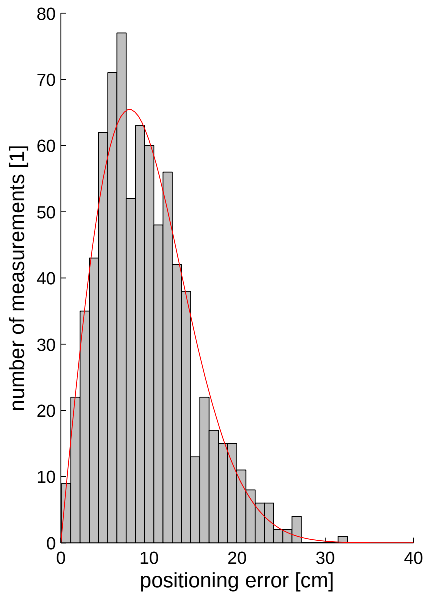
The same simulation runs as described in the last section are now extended to the three dimensional space. The random distance values are again normal distributed with a standard deviation of 8.75 centimeters. As in the latter simulation the results' coordinates are also normal distributed. This leads to the assumption that the positioning error will be Chi distributed this time with three degrees of freedom.

Several simulation runs were made with different numbers of beacon at varying positions. In all cases, the positioning error was higher than in the 2D simulation. As example the error with eight beacons arranged as cube is with 11 centimeters average still higher. On the other hand the error for only four beacons, which is the least amount of reference nodes to perform a multilateration process, is with 14 centimeters no that much higher. Depending on the position of the beacon nodes, the variance of the three coordinates can be very different. The variances are almost equal for regular beacon placement, as for example in a cube. If one coordinate is poorly specified by only one beacon, for example all beacons share the same z-coordinate and only one reference is placed at a different height, the according z-coordinate variance increases. Though this effect is not that visible in the positioning error, because the other two coordinates are specified more precisely instead.

Figure 5.11(a) shows this effect. The five beacons are positioned as in Section 5.2.1



(a) Map showing the results of the positioning algorithm giving cumulating around the ideal positions.



(b) Absolute position error of the calculated results, following a Rayleigh distribution

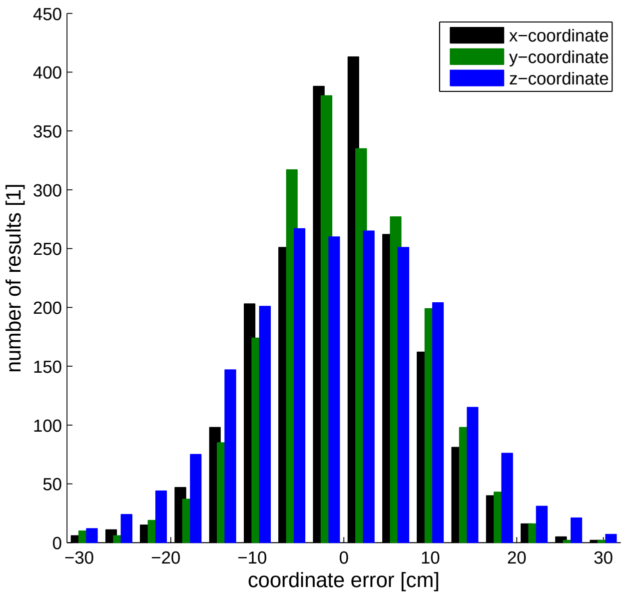
Figure 5.10: Simulation results of the positioning algorithm for two dimensions.

with a fifth reference at the origin in 3 meters height. The histogram of the coordinate errors shows the discussed effect that the z-coordinate introduces a 20 percent higher variance. The resulting position error is shown in Figure 5.11(b). The error is indeed very similar to the Chi distribution.

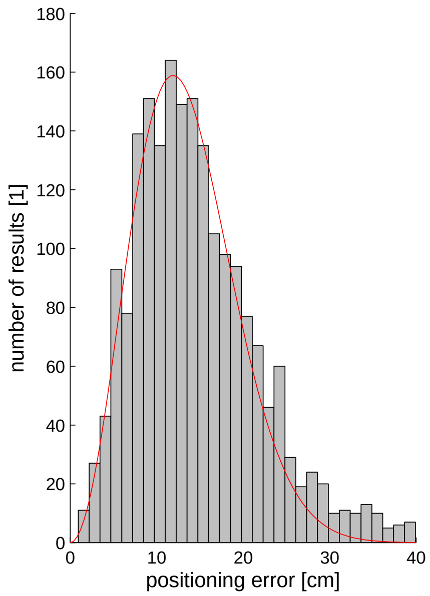
5.2.3 Positioning Algorithm Robustness against Distance Failures

The goal of this simulation is to verify the error correction approach discussed in Section 3.4.2. Outliers in the distance values mostly come from echo detections and can have much higher values than for direct line of sight. To prove the concept, the simulator significantly changes the mean of the distance to one reference node. The calculated SVD error should be used for two matters, first to detect the a distance value outlier exists and secondly detect the according beacon and remove it for another position calculation.

The results were not that good as expected. The detection if an outlier is present has a good success rate. To minimize the false positive rate the maximal allowed error sum, as proposed in Equation (3.2) and specified in the configuration header file, has to be adjusted. By contrast, the outlier correction mechanism fails in most cases of echo measurements. If one distance value of the calculation is overestimated, the linear LS solution of the multilateration process gives no clear evidence, which beacon delivered a wrong distance. Out of different simulation runs with regular beacon topologies, different node positions and varying distance outliers a success rate for detecting the faulty measurement



(a) Deviance of the resulting position for each coordinate, all normally distributed.



(b) Absolute position error of the calculated results, following a Chi distribution

Figure 5.11: Simulation results of the positioning algorithm for three dimensions.

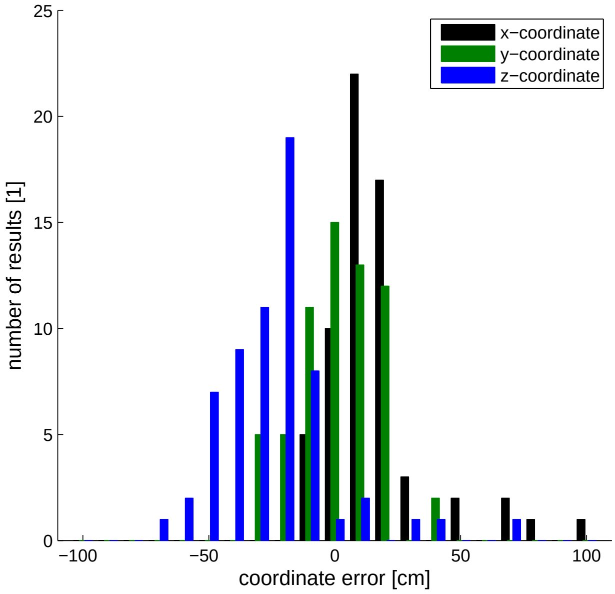
and thus decreasing the positioning error was only 20 percent. Additionally the positioning error increases almost every time a wrong beacon is banned from the recalculation.

This drawback of the positioning software makes the results very prone to faulty distance measurements. To get a really feasible positioning algorithm the stated sensibility has to be reduced by detecting the outlier more efficiently. This can be done by solving the over determined equation system with some a constrained multilateration approach or another type of equation linearization [Fogh04].

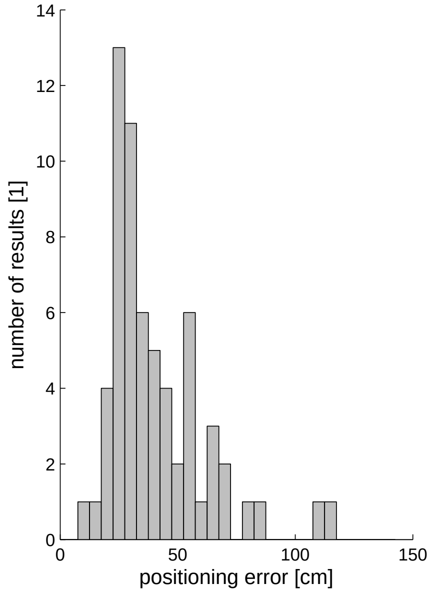
5.2.4 Test in a Laboratory Environment

A final test of the positioning system is made to prove the functionality and performance of the overall system. The experiment was performed in the same room as the test in Section 5.1.4. Five nodes act as infrastructure by placing them at beacons in the room spanning a cubic playground of approximately 3 meters edge length. The beacon positions are taken by random with regard to the local conditions and furniture. The positions between senders and receiver are at distances of up to 3.5 meters with a angular misalignment of up to 80 degrees from the center of the acoustic beam. As the test procedure is very complicated, only three different receiver positions are taken into account. As the rangings are all in free room, similar performance as in Section 5.1.4 is expected. One ranging consists of 16 single TDoA measurements with 5 chirp pulse sequences.

Figure 5.12 shows the combined results of all measurement points that were made at



(a) Deviance of the resulting position for each coordinate.



(b) Absolute error of the positioning results, nearly Chi distributed

Figure 5.12: Results of the positioning experiment in a laboratory environment. Five beacons are placed in the room at distances between 1 and 3.5 meters to the receiving node.

the three different receiver positions. To permit a comparison, the deviation to the real position is depicted. The main problems during the measurement are once more the echo detections. As the test setup enforces rangings in constellations with bad ranging behavior, the results are accordingly not as good as the simulation results. Additionally rangings to two of the beacon nodes were constantly overestimated. This was most likely caused by frequency mismatch based on the unit-to-unit variation of the sounder and therefore lower signal amplitude for the tone detector after the band pass. Figure 5.13 shows the topology of the positioning experiment. For a better overview, the aid lines project the nodes position to the three axis plains.

The high negative shift of the x-coordinate is mostly caused by the overestimation of the two beacons. In contrast the variance of all coordinates is almost as expected with values of 20 centimeters. This is the same as the experiment made in Section 5.1.4. Another factor for the bad ranging behavior is the high angular misalignment between some of the nodes. The signal amplitude at the receiver is much lower for these topologies. As result, the ranging process often has a low confidence because the receiver circuitry cannot detect all pulses of the sequences because of a too low SNR.

The overall accuracy of the experiment is a mean error of 40 centimeters. In other words 50 percent of the calculation results have an error of less than 33 centimeters as well as 90 percent are within an error of 66 centimeters.

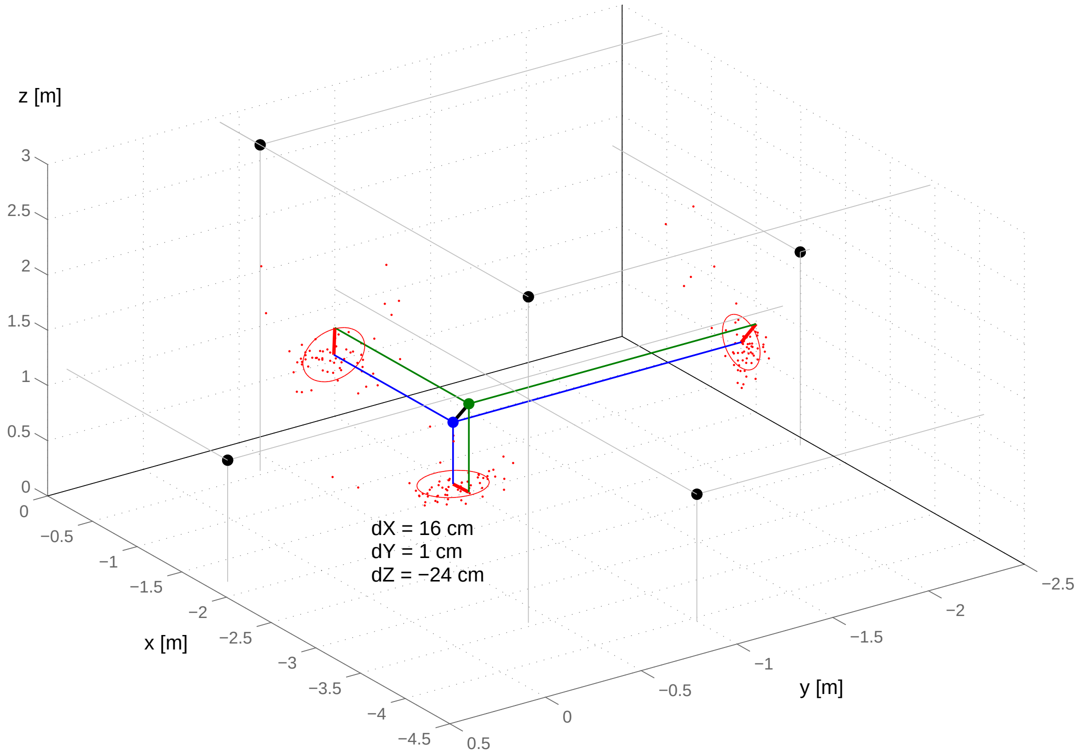


Figure 5.13: Shows the topology of the positioning experiment for one receiver position (green point). The mean calculated position of the 64 measurements is drawn as blue point, whereas the black points represent the beacons. The error vector and variance is displayed in red for each coordinate with additional points for each single position result.

5.2.5 Power Measurements

In a final section, the energy needed for a position calculation is analyzed. Before that the power dissipation of the positioning calculation is measured with the same test bed as the ranging power measurements (Section 5.1.5). The only new and interesting power profile is during the calculation of the position result. This profile is depicted in Figure 5.14. The profile splits in the real computation phase and the announcement of the node as assistant with the new coordinates.

The major part of the computation time is needed by the SVD as annotated in the graph. Within the current implementation the node cannot execute other synchronous tasks during this period of approximately 37 ms. It can happen that radio messages are overheard because the working tasks posted by the TinyOS radio stack are executed too late. During the first 8 ms of the calculation the beacons are elected and the linearization is done. At the end, the pseudo inverse is computed, followed by solving the equation system. The overall calculation time is around 58 ms for six beacons and will vary between 40 and 70 milliseconds depending on the amount of beacons.

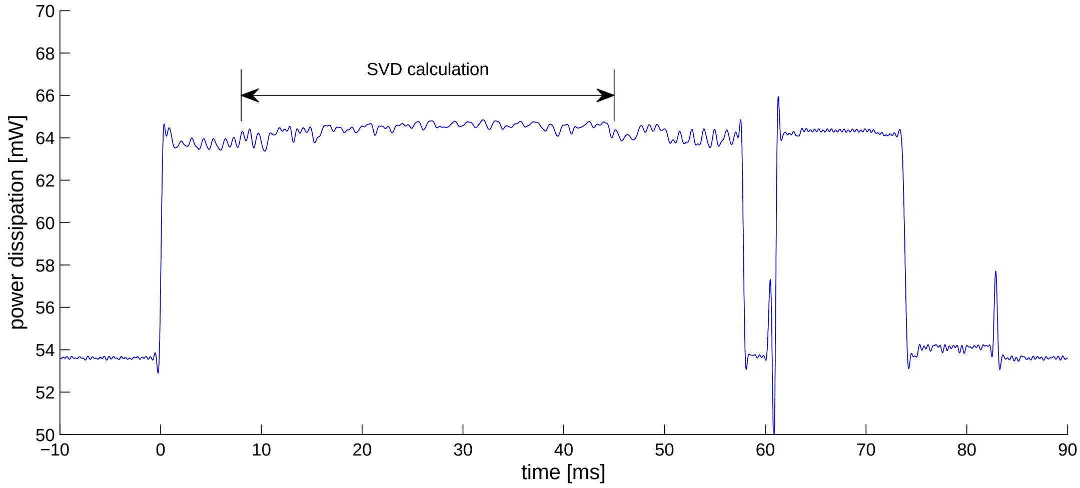


Figure 5.14: Low pass filtered power profile of a receiving node during the final position calculation. The profile splits into the calculation itself with annotated time for the TDoA, and the coordinate propagation with a radio message transmission.

The power consumption during the position calculation is 64.28 mW on average. The energy needed for the complete position calculation process including the following coordinate propagation is 5.85 mWs which is less than receiving a single pulse sequence. This fact makes it arguable to invest in more complicate positioning approaches than the current implementation.

Finally the overall needed energy for a typical positioning approach is stated. This value has a high variance since bad radio or acoustic links increase the execution time and therefore the energy needed. In a sample topology consisting of 5 reference nodes with a trustiness of 100 percent, the minimum required energy for the unknown node is 1.3 Ws. This value can increase to 6.9 Ws if there is a bad link quality to all beacon nodes. Each ranging is based on 16 measurements of 5 chirp pulse sequences with a fixed interval of 175 ms. The worst case scenario should not occur, because with a regular infrastructure density there are always some reference nodes with good link quality.

Memory Consumption

The used microprocessor has a program memory of 128 kB and the implemented localization system needs 42 kB. In comparison to WSN programs delivered with the TinyOS source, this is a high value but still leaves plenty of space for additional functionality. For example a program using almost only the radio stack needs 9.5 kB of program memory. As the ranging system needs only 24 kB, the difference is needed only for the positioning algorithm and that are 19 kB. From the memory consumed by the ranging system, a part is used by TinyOS for the radio stack, I2C stack and the other hardware abstractions that are provided. So the real code size for the ranging implementation alone is between 8 and 10 kB.

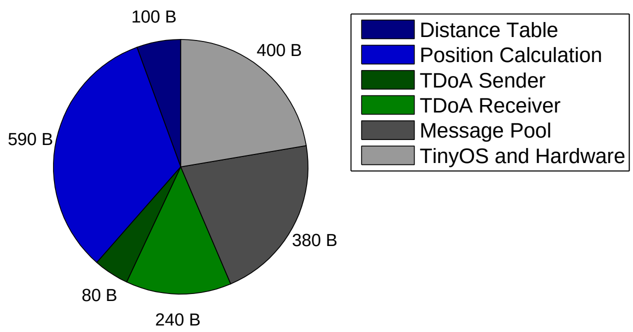


Figure 5.15: The pie chart shows the memory usage of the main positioning components including the part of TinyOS. The whole program needs 1790 bytes of memory, 690 bytes are dedicated to the positioning system, 320 bytes to the ranging system.

A more detailed analysis is made for the data memory holding all global variables and arrays and the complete stack during runtime. The processor provides 4 kB of memory, which should never be completely used by variables. The localization system currently needs 1790 bytes of memory with the following configuration: the radio message buffer is set to 8 and the positioning table size is set to 6, which has influence on the array size needed by the calculation module. Further the interrupt buffer for the PSD as well as the TDoA results array is set to 32 entries. Figure 5.15 shows a breakdown of the memory needed by the main components. The most memory is needed for the matrices of the multilateration process. If a small memory footprint is needed, the configuration can be adjusted to a message buffer size of 2, table size of 4 and an array length of 16 or even less. This settings give a total memory consumption of only 1090 bytes.

Summary

A final enumeration lists the most important outcomes of the positioning evaluation:

- The positioning error in this theses is characterized by the mean Euclidean distance of a result set.
- A higher number of beacons does not necessarily minimize the positioning error.
- The beacon arrangement has a high impact on the precision of the results.
- Echo detection based on the multilateration error is not as promising as expected, because the outlier cannot be detected correctly.
- The positioning error is 40 centimeters on average with a ranging precision of approximately 20 centimeters.
- The energy needed for a positioning process with 5 beacons varies between 1.3 and 6.9 Ws.

Chapter 6

Conclusion

This thesis presented a new localization approach in indoor scenarios for WSNs. The localization is based on distances gathered from a TDoA measurement between a radio and an acoustic message. The Mica2 sensor node is taken as target platform, since the additional expansion board provides the needed speaker and microphone circuitry. The resulting position is a least squares solution of the multilateration process.

For distance measurement the different propagation speed of electromagnetic and sound waves is capitalized. Both signals are emitted within a tightly measured time interval from the beacon node. The receiver measures the time difference between these two signals and can convert this time into a distance value based on the knowledge of the exact speed of sound and a constant offset based on hardware latencies. The radio message is detected by MAC time stamping whereas the acoustic signal is first filtered and detected in hardware and afterwards received as external interrupt by the micro processor. To gain robustness to environmental noise and multipath propagation a pulse sequence is used which consists of several acoustic impulses that are sent with a known pattern. As the detection mechanism introduces a high standard deviation, successive measurements are statistically analyzed to gain better results. Up to a distance of five meters the ranging has an error of 8.75 cm on average, some measurements have only 4 cm error. To gain enough robustness only 16 measurements are sufficient, needing 200-300 mWs of energy.

The positioning system relies on an infrastructure consisting of reference points. These nodes know their own position and act as beacons for the rest of the network. A unknown node first has to detect these beacons and store their meta data. Afterwards a ranging process to several of these reference nodes is executed. Together with the beacons coordinates the multilateration problem can be solved to a linear LS solution. Since the equation system is typically overestimated and not solvable to a single solution, the needed pseudo inverse of the input matrix is done by a singular value decomposition. Simulations show that the position error is around twice the ranging error. Experiments with tricky node topologies resulted in a error of 40 cm based on a ranging error of 20 cm, confirming the simulations.

The accuracy of the system is sufficient to speak from a fine-grained localization system, and can be used by a WSN to add a location context. The major downside of the current implementation is the short operating range of 5 meters as compared with outdoor systems. This low range is mainly caused by multipath propagation, which is outdoors not a serious problem.

Future Work

The presented solution of an indoor positioning system for wireless sensor networks is already suited to get fine grained results, but different improvements can be made. The following listing mentions these as possible parts for a future work:

1-to-n ranging: This facet of the ranging protocol already discussed in the concept of the thesis is currently not implemented. The related parts of the program are prepared for this option and only the state machine has to be added in the two protocol modules. The benefit would be a lower power consumption of all beacons. On the other hand, this mechanism increases the execution time for the receiver nodes and therefore increases the time needed for a positioning approach.

Bidirectional ranging is also not supported by the current implementation. This mechanism can be used to detect non line of sight measurements or even to filter out echo measurements because the impulse response of the room depends on the direction of measurement. A slight drawback is the incompatibility with 1-to-n ranging because a bidirectional ranging can only be made between two distinct nodes.

Collaborative multilateration: This approach already presented in the literature part improves the success rate of a positioning process in areas with spare infrastructure. The computational effort increases because twice as much unknown variables have to be solved by the linear LS algorithm.

Constrained multilateration: The current implementation of the multilateration process needs reference nodes whose coordinates differ in every direction. With additional constraints a three dimensional solution can be found even when all beacons share the same z-coordinate. This would make it possible to rely only on ceiling mounted beacons and therefore reduce the typical angular misplacement between sender and receiver.

Low Power Listening should be introduced which would reduce the energy needed for a positioning process to less than 50%, because the power consumed by the radio in idle receive mode is a major part of the system power dissipation. The low-level TDoA mechanism is also resistant to this radio mode.

Acoustic actuator: The current fixed frequency piezo-electric sounder has a very low power profile of approximately 10 mW. This value could be increased because higher output power results in a higher received signal amplitude and therefore a more precise detection further resulting in a reduced number of measurements and power consumption for both sides. A buzzer with better power-on dynamics would also increase the measurement precision.

Receiver circuitry: The used amplifier of the analog microphone signal had a too low gain to support higher distance rangings. A higher gain in combination with a improved band pass would results in a much better input signal of the tone detector.

Bibliography

[Agha03] G. Agha, W. Kim, Y. Kwon, K. Mechitov, and S. Sundresh. Evaluation of Localization Services (Preliminary Report). In DARPA NEST, 2003.

[Akyildiz02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, volume 38, pages 393–422, 2002.

[Ansari07] J. Ansari, J. Riihijarvi, and P. Mahonen. Combining Particle Filtering with Cricket System for Indoor Localization and Tracking Services. In PIMRC 2007, pages 1–5, Sept. 2007.

[Atmel] Datasheet: Atmel AtMega128L, 2004. `http://www.atmel.com/atmel/acrobat/doc2467.pdf`, visited: 14.01.2010.

[Bachrach04] J. Bachrach, R. Nagpal, M. Salib, and H. Shrobe. Experimental Results for and Theoretical Analysis of a Self-Organizing Global Coordinate System for Ad Hoc Sensor Networks. *Telecommunication Systems*, volume 26, number 2, pages 213–233, June 2004.

[Bahl00] P. Bahl and V. N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In INFOCOM 2000, volume 2, pages 775–784, Mar. 2000.

[Battelli07] M. Battelli and S. Basagni. Localization for Wireless Sensor Networks: Protocols and Perspectives. In CCECE 2007, pages 1074–1077, Apr. 2007.

[Boukerche07] A. Boukerche, H. A. B. Oliveira, E. F. Nakamura, and A. A. F. Loureiro. Localization systems for wireless sensor networks. *IEEE Wireless Communications Magazine*, volume 14, number 6, pages 6–12, 2007.

[Bulusu00] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, volume 7, number 5, pages 28–34, 2000.

[Capkun01] S. Capkun, M. Hamdi, and J. P. Hubaux. GPS-free positioning in mobile ad-hoc networks. In HICSS 34th, Jan. 2001.

[Chipcon] Datasheet: Chipcon CC1000, 2009. http://focus.ti.com/lit/ds/symlink/cc1000.pdf, visited: 14.01.2010.

[Cramer93] O. Cramer. The variation of the specific heat ratio and the speed of sound in air with temperature, pressure, humidity, and CO2 concentration. *The Journal of the Acoustical Society of America*, volume 93, number 5, pages 2510–2516, 1993.

[Crossbow1] Datasheet: MPR/MIB Users Manual, 2007. http://www.xbow.com/support/support_pdf_files/mpr-mib_series_users_manual.pdf, visited: 14.01.2010.

[Crossbow2] Datasheet: MTS/MDA Sensor Board Users Manual, 2007. http://www.xbow.com/support/Support_pdf_files/MTS-MDA_Series_Users_Manual.pdf, visited: 14.01.2010.

[Davids08] A. Davids. Positionsbestimmung innerhalb eines Gebäudes auf Basis eines Motes-Sensornetzwerks in einem Rescue-Szenario. Master's thesis, Fakultät für Technik und Informatik, Hochschule für Angewandte Wissenschaften Hamburg, 2008.

[Doherty01] L. Doherty, K. S. J. Pister, and L. El Ghaoui. Convex position estimation in wireless sensor networks. In INFOCOM 2001, volume 3, pages 1655–1663 vol.3, Apr. 2001.

[Elson02] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Oper. Syst. Rev.*, volume 36, pages 147–163, 2002.

[Farrokhi05] H. Farrokhi and R. J. Palmer. The designing of an indoor acoustic ranging system using the audible spread spectrum LFM (chirp) signal. In CCECE 2005, pages 2131–2134, May 2005.

[Fogh04] M. Fogh, T. Mikkelsen, E. Mustafic, A. Nielsen, R. Petersen, and J. Sommerset. Absolute position estimation calculated from acoustic time of flight. Department of Control Engineering, Aalborg University, 2004.

[Gay05] D. Gay, P. Levis, D. Culler, and E. Brewer. *nesC 1.2 Language Reference Manual*, Aug. 2005.

[Girod01] L. Girod and D. Estrin. Robust range estimation using acoustic and multimodal sensing. In IROS '01, volume 3, pages 1312–1320, Oct. 2001.

[Girod06] L. Girod, M. Lukac, V. Trifa, and D. Estrin. The design and implementation of a self-calibrating distributed acoustic sensing platform. In SenSys '06, pages 71–84, Nov. 2006.

[Glatz10a] P. M. Glatz, L. Hoermann, C. Steger, and R. Weiß. A System for Accurate Characterization of Wireless Sensor Networks with Power States and Energy Harvesting System Efficiency. In PerSeNS 2010, Mar. 2010.

[Glatz10b] P. M. Glatz, C. Steger, and R. Weiß. Poster Abstract: TOSPIE2: Tiny Operating System Plug-In for Energy Estimation. In IPSN 2010, Apr. 2010.

[Guha05] S. Guha, R. Murty, and E. G. Sirer. Sextant: a unified node and event localization framework using non-convex constraints. In MobiHoc '05, pages 205–216, Apr. 2005.

[Hadim06] S. Hadim and N. Mohamed. Middleware for Wireless Sensor Networks: A Survey. In Comsware 2006, pages 1–7, Jan. 2006.

[Harter02] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. *Wireless Networks Journal*, volume 8, number 2, pages 187–197, 2002.

[He03] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In MobiCom '03, pages 81–95, 2003.

[Hightower01] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer Journal*, volume 34, number 8, pages 57–66, 2001.

[Hongyang05] C. Hongyang, D. Ping, X. Yongjun, and L. Xiaowei. A robust location algorithm with biased extended Kalman filtering of TDOA data for wireless sensor networks. In WCNM 2005, volume 2, pages 883–886, Sept. 2005.

[Jeong01] Y. Jeong, H. You, and C. Lee. Calibration of NLOS error for positioning systems. In IEEE VTS 53rd, volume 4, pages 2605–2608, May 2001.

[Ji04] X. Ji. *Localization algorithms for wireless sensor network systems*. PhD thesis, Department of Computer Science and Engineering, Pennsylvania State University, 2004.

[Jordt06] G. J. Jordt. Evaluation of Energy Costs and Error Performance of Range-Aware, Anchor-Free Localization Algorithms for Wireless Sensor Networks. Master's thesis, Department of Electrical and Computer Engineering, Air University, Ohio, 2006.

[Julier97] S. J. Julier and J. K. Uhlmann. A New extension of the Kalman filter to nonlinear systems. In SPIE 1997, volume 3068, pages 182–193, July 1997.

[Kang07] J. Kang, D. Kim, and Y. Kim. RSS Self-calibration Protocol for WSN Localization. In ISWPC '07, May 2007.

[Kushwaha05] M. Kushwaha, K. Molnar, J. Sallai, P. Volgyesi, M. Maroti, and A. Ledeczi. Sensor node localization using mobile acoustic beacons. In MASS '05, Oct. 2005.

[Kwon05] Y. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha. Resilient Localization for Sensor Networks in Outdoor Environments. In ICDCS 2005, pages 643–652, June 2005.

[Landsiedel05] O. Landsiedel, K. Wehrle, and S. Gotz. Accurate prediction of power consumption in sensor networks. In EmNets 2005, pages 37–44, May 2005.

[Langendoen05] K. Langendoen and N. Reijers. *Embedded Systems Handbook*, chapter Distributed Localization Algorithms, pages 36.1–36.23. CRC press, 2005. ISBN: 978-1-420-07410-9.

[Levis03] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: accurate and scalable simulation of entire TinyOS applications. In SenSys '03, pages 126–137, Nov. 2003.

[Lorincz07] K. Lorincz and M. Welsh. MoteTrack: a robust, decentralized approach to RF-based location tracking. *Personal Ubiquitous Computing*, volume 11, number 6, pages 489–503, 2007.

[Maróti04] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In SenSys '04, pages 39–49, Nov. 2004.

[Meschach] Meschach: Matrix computations in C. http://www.cs.uiowa.edu/~dstewart/meschach/, visited: 12.01.2010.

[Meyer07] P. Meyer. Design and Implementation of an Energy Harvesting System for Wireless Sensor Nodes. Master's thesis, Institute for Technical Informatics, Graz University of Technology, 2007.

[Murphy07] W. S. Murphy. Determination of a position using approximate distances and trilateration. Master's thesis, Department of Mathematical and Computer Sciences, Colorado School of Mines, July 2007.

[Nasipuri02] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. In WSNA '02, pages 105–111, Sept. 2002.

[Niculescu03a] D. Niculescu and B. Nath. Ad hoc positioning system (APS) using AOA. In INFOCOM 2003, volume 3, pages 1734–1743, Apr. 2003.

[Niculescu03b] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *Journal of Telecommunication Systems*, volume 22, number 1-4, pages 267–280, 2003.

[Niculescu04] D. Niculescu and B. Nath. Error characteristics of ad hoc positioning systems (aps). In MobiHoc '04, pages 20–30, June 2004.

[Palmer02] R. Palmer. A spread spectrum acoustic ranging system - an overview. In CCECE 2002, volume 3, pages 1242–1245, May 2002.

[Panasonic] Datasheet: Multilayer Chip NTC Thermistors, 2002. `http://www.panasonic.com/industrial/components/pdf/arg0000ce1.pdf`, visited: 14.01.2010.

[Pletzer08] F. Pletzer. Location Based Services for Wireless Sensor Networks. Master's thesis, Institute for Technical Informatics, Graz University of Technology, 2008.

[Popa08] M. Popa, J. Ansari, J. Riihijarvi, and P. Mahonen. Combining Cricket System and Inertial Navigation for Indoor Human Tracking. In WCNC 2008, pages 3063–3068, Mar. 2008.

[Priyantha00] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket location-support system. In MobiCom '00, pages 32–43, Aug. 2000.

[Priyantha01] N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In MobiCom '01, pages 1–14, July 2001.

[Priyantha03] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-Free Distributed Localization In Sensor Networks. In SenSys '03, pages 340–341, Nov. 2003.

[Reichenbach06] F. Reichenbach, A. Born, D. Timmermann, and R. Bill. *Distributed Computing in Sensor Systems*, volume 4026 of *Lecture Notes in Computer Science*, chapter A Distributed Linear Least Squares Method for Precise Localization with Low Complexity in Wireless Sensor Networks, pages 514–528. Springer Berlin, 2006. ISBN: 978-3-540-35227-3.

[Römer01] K. Römer. Time synchronization in ad hoc networks. In MobiHoc '01, pages 173–182, Oct. 2001.

[Römer04] K. Römer and F. Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications Magazine*, volume 11, number 6, pages 54–61, 2004.

[Römer05] K. Römer, P. Blum, and L. Meier. *Handbook of Sensor Networks: Algorithms and Architectures*, volume 1 of *Parallel and Distributed Computing*, chapter Time Synchronization and Calibration in Wireless Sensor Networks. Wiley, 2005. ISBN: 978-0-471-68472-5.

[Rong06] P. Rong and M. L. Sichitiu. Angle of Arrival Localization for Wireless Sensor Networks. In SECON '06, volume 1, pages 374–382, Sept. 2006.

[Sallai04] J. Sallai, G. Balogh, M. Maróti, Ákos Lédeczi, and B. Kusy. Acoustic Ranging in Resource-Constrained Sensor Networks. In ICWN '04, June 2004.

[Savarese02] C. Savarese, J. M. Rabaey, and K. Langendoen. Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks. In ATEC '02, pages 317–327, June 2002.

[Savvides01] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in Ad-Hoc networks of sensors. In MobiCom '01, pages 166–179, July 2001.

[Savvides02] A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In WSNA '02, pages 112–121, Sept. 2002.

[Saxena05] S. Saxena and H. P. Singh. Some Estimators of the Dispersion Parameter of a Chi-distributed Radial Error. *Austrian Journal of Statistics*, volume 34, number 1, pages 51–63, 2005.

[Schmidt86] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, volume 34, number 3, pages 276–280, 1986.

[Shahzad08] K. Shahzad, A. Ali, and N. D. Gohar. ETSP: An Energy-Efficient Time Synchronization Protocol for Wireless Sensor Networks. In AINAW 2008, pages 971–976, Mar. 2008.

[Shnayder04] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh. Simulating the power consumption of large-scale sensor network applications. In SenSys '04, pages 188–200, Nov. 2004.

[Simic01] S. N. Simic and S. Sastry. Distributed Localization in Wireless Ad Hoc Networks. Technical report, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 2001.

[Stoleru04] R. Stoleru and J. A. Stankovic. Probability grid: a location estimation scheme for wireless sensor networks. In SECON '04, pages 430–438, Oct. 2004.

[Stoleru07] R. Stoleru, J. A. Stankovic, and S. H. Son. Robust node localization for wireless sensor networks. In EmNets 2007, pages 48–52, June 2007.

[Taylor06] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue. Simultaneous localization, calibration, and tracking in an ad hoc sensor network. In IPSN 2006, pages 27–33, Apr. 2006.

[TinyOS] Hompage: TinyOS. http://www.tinyos.net/, visited: 14.01.2010.

[Trathnigg08] R. Trathnigg, T. Weiss. A runtime energy monitoring system for wireless sensor networks. In ISWPC '08, pages 21–25, May 2008.

[Ward97] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, volume 4, number 5, pages 42–47, 1997.

[Welch95] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report TR 95-041, University of North Carolina, Department of Computer Science, 1995.

[Whitehouse02] C. D. Whitehouse. The Design of Calamari: an Ad-hoc Localization System for Sensor Networks. Master's thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 2002.

[Xiao06] J. Xiao, L. Ren, and J. Tan. Research of TDOA Based Self-localization Approach in Wireless Sensor Network. In IROS '06, pages 2035–2040, Oct. 2006.

[You06] Y. You and H. Cha. Scalable and Low-Cost Acoustic Source Localization for Wireless Sensor Networks. In UIC 2006, volume 4159, pages 517–526, 2006.

[Yu06] K. Yu, M. Hedley, I. Sharp, and Y. J. Guo. Node Positioning in Ad Hoc Wireless Sensor Networks. In INDIN '06, pages 641–646, Aug. 2006.

[Zhang07] J. Zhang, T. Yan, J. A. Stankovi, and S. H. Son. Thunder: towards practical, zero cost acoustic localization for outdoor wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, volume 11, number 1, pages 15–28, 2007.