ISSN 0280-5316
ISRN LUTFD2/TFRT--5880--SE

Using Sensor Equipped Smartphones to Localize WiFi Access Points

Anton Hansson
Linus Tufvesson

Department of Automatic Control
Lund University
May 2011

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
May 2011

Document Number
ISRN LUTFD2/TFRT--5880--SE

Author(s)
Anton Hansson and Linus Tufvesson

Supervisor
Magnus Persson Sony Ericsson Mobile Lund, Sweden. Fredrik Tufvesson Lund University Electrical and Information Techn. Sweden
Bo Bernhardsson Automatic Control Lund. Sweden (Examiner)

Sponsoring organization

Title and subtitle
Using Sensor Equipped Smartphones to Localize WiFi Access Points (Lokalisering av WiFi-accesspunkter m.h.a. sensorutrustade smartphones)

Abstract
Determining the position of mobile devices is an important business and research area due to the many applications it enables. Outdoors, positioning is often solved by the Global Positioning System (GPS). Indoors however, the signals received from GPS satellites are too weak to provide meaningful position estimates. Therefore, indoor positioning must be done using other techniques, such as trilaterion based on WiFi signal strengths. The trilaterion technique requires that the positions are known for the WiFi Access Points (APs) used. While determining AP positions can be done manually, this is often time consuming, error prone and fragile to changes of AP positions. A better way is to use an automatic system for determining the AP positions. There are two contributions presented in this thesis. The first contribution is a navigation system using accelerometers and gyroscopes in a phone to track the movement of a walking person. A requirement of the navigation system is that the start position and heading of the user is known, and that the orientation of the phone does not change in respect to the user while navigating. The navigation system typically gives position estimates accurate to within a few meters for up to two minutes after navigation started.The second contribution is an algorithm that given a set of Received Signal Strength Indications (RSSIs) at different locations provides an estimate of where the transmitter is located. The accuracy of the algorithm varies depending on the coverage of measurements as well as the radio environment around the AP. The position of APs can typically be estimated within five meters from their true location when provided with measurements from several angles from the AP.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
94

Recipient's notes

Security classification

Acknowledgements

Several persons have contributed with ideas, technical solutions and practical help when we needed it the most. We would like to thank all the people in the Technology group at SEMC for their welcoming attitude and great company. In particular, we would like to thank our advisor Magnus Persson since this thesis would never have been created without him, and for all the ideas and feedback provided on our work throughout the thesis. Bo Bernhardsson and Fredrik Tufvesson have been our advisors at Lund University and have provided great help of more technical nature. Finally, Magnus Midholt, Erik Hellman and Masar Sadik deserve mentioning for helping us with more practical problems arising when implementing and testing our developed system.

Thank you!

Contents

Chapter 1

Introduction

This thesis describes a system developed at Sony Ericsson Mobile Communication (SEMC) in Lund, Sweden, as part of the authors' pursuit of a Master's degree. The work is a cooperation between SEMC, the Department of Automatic Control, Lund University and the Department of Electrical and Information Technology, Lund University.

1.1 Background

Positioning of mobile devices is an important area of research, and many applications requires the current position of the device to function. For example, the position of a mobile device can be combined with a map to help the user navigate. Another example is configuring the mobile device differently depending on its current location, e.g. lowering the volume of a cell phone when entering a meeting room.

Outdoors, the Global Navigation System (GPS) is widely used to provide a position accurate within a few meters. However, GPS requires signal contact with satellites to work and when indoors, the signal becomes too weak to provide meaningful position data.

Indoor positioning is a popular research topic and several methods have been proposed. One method is to use sensors such as accelerometers and gyroscopes to determine how the device is moved. The problem with this approach is that tracking the sensors can only give locations that are relative to previous locations and can never give an absolute position. Additionally, since the accuracy of the sensors are limited, location can only be tracked for a relatively short time before the position estimate becomes useless.

Another method is to use WiFi signals, since nowadays most buildings are equipped with wireless access points (APs). The idea is that if the locations of a number of APs are known, and the distance to each AP is also known, one's

own position can be determined using a technique called *trilateration*. The distance to APs can be estimated using the received signal strength indication (RSSI). This approach gives a location that is not affected by time or previous locations like the sensor approach.

Of course, the location of the APs must be determined accurately for the trilaterated position to be accurate. This causes a problem since it typically involves manual labor, is error-prone and does not respond well to changes in AP positions. A way to overcome this obstacle is to use a system that automatically determines the location of access points.

Because the usage of smart phones is widespread and growing, constructing the system for smart phones is a way to reach a large amount potential contributors.

1.2 Goal

The thesis describes a system that uses sensor equipped Android smart phones to localize the position of WiFi access points. The goal is to develop a system that makes it seamless to establish and maintain the positions of APs.

To achieve this, a navigation system is created using the sensors in the phone (accelerometer and gyroscope) to track the location of the user. While the location estimate is accurate, measuring the RSSI from nearby APs enables the creation of a database of positions and distances to APs. With such a database, it is possible to conclude where the APs are most likely to be positioned.

The database is essentially made up of very noisy measurements. The navigation system can only provide meaningful positions for a short time, but even in a short time span the location estimates are very noisy. Additionally, the RSSI values at a given distance from an AP also fluctuates widely which makes the distance estimate noisy.

As such, a single measurement cannot be trusted. Instead, each measurement is only used to give a weak indication of where an AP might be, and it is only by combining a very large amount of measurements that the results become meaningful. In essence, the system described uses a large set of noisy measurements to estimate the location of APs.

1.3 Limitations

The system developed is to be regarded as a proof of concept of how access points can automatically be located. For a commercially useful application, some of the assumptions made in this thesis must be dealt with and solved.

It is assumed that the start position and orientation is known by the navigation

system. This is reasonable assumption, since there are several technologies that could be used to determine the start position. For example, the user might be outside and have an accurate GPS position before entering the building where navigation is to take place.

The system does not rely on map information to aid navigation or to predict the propagation behavior of WiFi signals. While it could potentially make the system better, there is no existing infrastructure for distributing maps of buildings known to the authors. It would therefore make the system more limited if it relied on map information to be available.

The typical use case of the system is a person walking around with the phone placed in a pocket, hand or handbag. It is assumed that the phone orientation is fixed in respect to the person navigating. This is needed if the navigation system relies on detecting the steps of the user, since an orientation change of the phone must reflect orientation changes of the user.

Four sensors are assumed to exist, and these are gyroscope, accelerometer, magnetometer and a WiFi chip for scanning signals.

1.4 Platform

The phone used in this thesis was a Sony Ericsson Xperia Neo. Future references to 'the phone' refer to this phone. Xperia Neo is a smart phone running Android[1] version 2.3 and is equipped with an accelerometer, magnetometer and a WiFi chip. Additionally, the phone used in the thesis was equipped with a gyroscope and a pressure sensor.

1.4.1 Android

Android is an operating system for mobile devices and is based on version 2.6 of the Linux kernel. The Android SDK[2] provides the tools and APIs necessary to develop applications for Android devices using the Java programming language. The API provides methods for accessing the sensors in the phone. All access to sensor data was done through Java applications written by the authors.

1.4.2 Event based sensor sampling

In Android, sensor data is made available through events. An event contains the data from the sensor along with a time stamp of when it was measured. The user can provide the system with a desired rate at which he would like

[1] http://www.android.com
[2] http://developer.android.com/sdk/index.html

to receive sensor values, but there is no guarantee that the device will deliver events at the desired rate. Even though the sensors run at a fixed frequency, not all the samples reach the application layer. This is due to the limited resources on the device, and the existence of other processes such as garbage collection which competes for these resources.

It is important to know how the sensors are sampled since most control theory on the subject of indoor navigation deals with sensors giving samples at constant time intervals.

1.5 Outline

The structure of this thesis is outlined as follows:

Chapter 2 contains an introduction to the available sensors and provides a brief overview of indoor navigation.

Chapter 3 discusses errors of the available sensors and ways to model and measure them.

Chapter 4 describes the theory of inertial navigation and evaluates the possibility of performing dead reckoning to track the position of the phone.

Chapter 5 defines the local frame used in the remainder of the thesis.

Chapter 6 presents a method for determining the distance traveled by using the accelerometer to detect steps.

Chapter 7 models how the orientation of the phone can be tracked by integrating of the gyroscopes. Additionally it explorers the possibility of aiding the orientation estimate by using the magnetometer sensor.

Chapter 8 merges the algorithms developed in Chapter 6 and Chapter 7 to an algorithm capable of estimating the phones position.

Chapter 9 covers the WiFi aspect of the thesis. First radio propagation is discussed and then an algorithm for trilaterating the position of WiFi APs is presented.

Chapter 10 is dedicated to the design of the final system, which combines the methods discussed in Chapter 8 and Chapter 9 to create a system that trilaterates the position of APs based of WiFi data collected during navigation.

Chapter 11 evaluates the result of the thesis and suggests future work on the subject.

There is a requirement from Lund University that it must be clear which contributions have been made by each author. In this thesis, the authors have been working together both when developing and implementing the system as well as writing the report.

Chapter 2

Background navigation theory

This chapter gives an introduction to the field of navigation and reviews some of the theory needed to construct the navigation system. It discusses the functionality of the sensors and how they can be used in a navigation system, followed by explanations of the different coordinate system arising from strapdown navigation.

2.1 Available sensors

The phone contains three sensors that could potentially be used in a navigation system. The three sensors are:

- 3-axis accelerometer
- 3-axis gyroscope
- 3-axis magnetometer

This section describes what the sensors measure and how they can be used.

2.1.1 Accelerometer

The accelerometer measures the acceleration of the device (in m/s^2), by measuring the forces affecting the sensor. The force of gravity is always affecting the sensor, which means the measurements will consist of the sum of true acceleration and the gravity. The acceleration is measured in m/s^2, projected on the 3 axes of the body coordinate system. The accelerometer has several

applications. For example, as we will see later, it can be used in combination with the magnetometer to determine the orientation of the phone when it is stationary.

Another application is determining how the device is moving using dead reckoning. When the orientation of the phone is known in all 3 axes, the direction of gravity is also known. As such, the gravity constant g can be subtracted from the accelerometer output. The remaining acceleration is the current acceleration of the device. Integrating the acceleration of the device gives the change in velocity, and by integrating the velocity of the device the change in position can be observed. Hence, if the starting position and velocity is known, the current position can be estimated by observing the accelerometer output. Dead reckoning is investigated when constructing the INS in Chapter 4.

The accelerometer can also be used to create a step counter. When a person is walking, the steps are usually very regular and observing the output of the accelerometer enables the detection of steps. The steps can be used to approximate the distance moved.

2.1.2 Gyroscope

The gyroscope measures the angular velocity around the three axes of the phone. The measurements returned are expressed in degrees or radians per second, and indicate how the device is rotating. Integrating the gyroscope output provides the total change in angle during the integration time. Hence, if the starting orientation is known, the current orientation can be estimated by observing the gyroscope output.

2.1.3 Magnetometer

The magnetic sensor measures the strength and direction of the magnetic field affecting the phone in three axis. The measurements returned are expressed in the unit micro-Tesla. Often, the geomagnetic field is the only magnetic field affecting the magnetometer. If the rough position on earth (e.g. which country) is known, so is the orientation of the geomagnetic field. Using this information, magnetometer measurements can provide the heading of the phone if the orientation around the X and Y axes is known.

A problem arises when using the magnetometer indoors, since there are typically a lot of magnetic disturbances. One source of disturbances are the many ferromagnetic material used in the building construction, and electric wires inside walls, but exterior equipment such as whiteboards and computers also create magnetic fields.

2.2 Strapdown navigation

In some systems, the sensors are placed on a stabilized platform, that maintains (nearly) the same orientation regardless of how their host object moves. Typically this is done by mounting the platform in a number of gimbals, which are rings that are connected in only one axis of rotation. The result of the gimbaled system is that any rotation done to the host will only change the orientation of the gimbals, not the platform. Figure 2.1 shows an example of a gimbaled system.

Figure 2.1: *An example gimbaled system. The innermost circle-shaped platform retains the same orientation regardless of how the outer rings rotate*

The purpose of the gimbaled system is that the sensors mounted on the stable platform are aligned with a fixed coordinate system. For example, an accelerometer could be mounted on such a platform, aligned so that one axis of the accelerometer corresponded to the direction north. The effect is that a positive signal from that axis will always indicate an acceleration towards the north, and the estimated position can be updated accordingly.

Unfortunately, gimbaled systems require high precision mechanical parts which are both large in size and expensive. Instead, recent advances of micromachined electromechanical system (MEMS) have lead to an increased accuracy and a decrease in both their size and cost. This has allowed them to be mass produced and put into regular smart phones. The big difference compared to the gimbaled systems is that the sensors are mounted onto the phone itself (the reason for the term strapdown navigation). This has the effect that a change in orientation of the phone also changes the orientation of the sensors.

Using the example of the accelerometer, no axis can be assumed to always indicate an acceleration towards the north, since the sensor itself has been rotated. Keeping track of the orientation of the phone is therefore of highest importance, since the output of the sensors is otherwise of completely unknown direction.

For the entire thesis, all coordinate systems are assumed to be right-handed

as shown in Figure 2.2

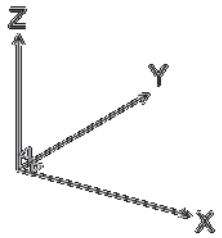


Figure 2.2: *The orientation of the coordinate systems discussed in this thesis. The axes are oriented according to the right-hand rule.*

2.3 World frame

It should be apparent that there is a need for a well defined coordinate system, that is fixed as far as the phone is concerned. One such coordinate system is the one referred to as the global coordinate system, which is commonly used for navigation.

The coordinate system of the world is defined as follows: [4]

x_w**:** A tangent to the ground of the earth at the current location, defined by $y_w \times z_w$. Points roughly east.

y_w**:** A tangent to the ground of the earth at the current location, points towards the magnetic North Pole.

z_w**:** Perpendicular to the ground of the earth, points towards the sky.

Figure 2.3 shows the orientation of the world frame.

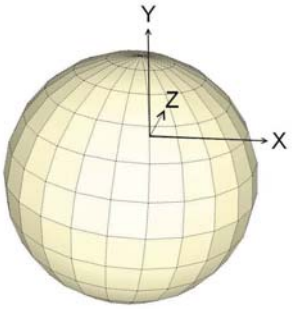


Figure 2.3: *The orientation of the world frame. At any location on earth, the X axis points east, the Y axis point north and the Z axis towards the sky*

It should be noted that the world frame is not fixed in space, since a change in location on earth changes the orientation of the system. However, since the

changes in position in indoor navigation are very small compared to the size of the earth these effects can safely by ignored. Additionally, since the earth is constantly rotating (approximately 15 deg/h), so is the orientation of the coordinate system, but just like the position change the effects are small [16].

2.4 Local frame

The world frame is an example of frame that is fixed as far as navigation system is concerned. It is sometimes useful to define another fixed frame specifically for the building in which navigation is currently taking place, referred to as the local frame.

For example, consider a building in which there are only corridors that are perpendicular to each other, and which directions are 10 degrees offset from north and east respectively. A map of the building might come across as more clear to a user if the corridors stretch perfectly horizontally and vertically instead of pointing in their correct 'world' direction. There can be more reasons for defining a local frame, as we will see in Chapter 5.

In other words, the local frame is typically defined as follows:

x_l: An axis in the same plane as x_w, offset by an angle θ.

y_l: An axis in the same plane as y_w, offset by an angle θ.

z_l: An axis perpendicular to the ground, equal to z_w.

The local frame may correspond to the world frame if the angle offset $\theta = 0$.

2.5 Body frame

The body frame is the coordinate system of the phone. The orientation of the body frame is not fixed in relation to the world and local frames, since a change in the heading of the user also changes the orientation of the phone. However, the body frame is fixed in relation to the phone itself. Figure 2.4 shows the orientation of the body frame with respect to the phone.

x_b: An axis that spans from left to right when looking at the phone in portrait mode.

y_b: An axis that spans from the bottom to the top of the phone, seen from the portrait orientation.

z_b: An axis spanning outwards from the screen.

Figure 2.4: *The orientation of the body frame relative to the phone.*

The orientation of the phone can be described as the rotation that aligns the local frame with the body frame. Rotations can be described in numerous ways. One example is Euler angles, which are a set of three angles that specify the rotations performed, in order, around three pre-defined axes. Euler angles are simple to understand, but have several shortcomings. A better system for representing rotations are the quaternions.

The naming convention for the sides of the phone is shown in Figure 2.5.

(a) (b)

Figure 2.5: *The naming convention for the various sides of the phone.*

2.6 Quaternions

The quaternions are a number system that is an extension of the complex numbers. They have several uses, but the important characteristic used here is

their ability to express 3 dimensional rotations. This section starts by defining what quaternions are and some of their properties, followed by a discussion of how they are used by us. Bear in mind that even though this section is quite short, it is very dense.

2.6.1 Theory

By definition, the quaternions are the set of numbers q that satisfy the following;

$$q = a + bi + cj + dk,$$

where a, b, c and d are real numbers and i, j and k are three new basis elements.

A quaternion is usually viewed as a scalar w and a vector $\vec{v} = \begin{pmatrix} x & y & z \end{pmatrix}^T$, leading to the following representation of a quaternion q

$$q = \begin{pmatrix} w \\ \vec{v} \end{pmatrix} = \begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix}.$$

The basis elements satisfy the following equations

$$i^2 = j^2 = k^2 = ijk = -1. \tag{2.6.1}$$

Using Equation 2.6.1, the following relations can easily be derived:

$$\begin{aligned} ij &= k, & ji &= -k, \\ jk &= i, & kj &= -i, \\ ki &= j, & ik &= -j. \end{aligned}$$

Many of the common arithmetic operations are defined for the quaternions. Most importantly for us, the quaternion multiplication (here denoted $\odot$) of two quaternions p and q is defined as

$$p \odot q = \begin{pmatrix} p_w \\ \vec{p_v} \end{pmatrix} \odot \begin{pmatrix} q_w \\ \vec{q_v} \end{pmatrix} = \begin{pmatrix} p_w q_w - \vec{p_v} \cdot \vec{q_v} \\ p_w \vec{q_v} + q_w \vec{p_v} + \vec{p_v} \times \vec{q_v} \end{pmatrix}. \tag{2.6.2}$$

Since the vector cross product is not commutative ($\vec{a} \times \vec{b} \neq \vec{b} \times \vec{a}$ in general), neither is the quaternion multiplication. It is, however, associative

$$(p \odot q) \odot r = p \odot (q \odot r).$$

Calculating the norm is done similarly to the norm of normal vectors

$$|q| = \sqrt{w^2 + x^2 + y^2 + z^2}.$$

The inverse q^{-1} of a quaternion q is defined by

$$q \odot q^{-1} = q^{-1} \odot q = \begin{pmatrix} 1 \\ \vec{0} \end{pmatrix},$$

and can easily be calculated by

$$q^{-1} = \begin{pmatrix} w \\ \vec{v} \end{pmatrix}^{-1} = \frac{1}{|q|^2} \begin{pmatrix} w \\ -\vec{v} \end{pmatrix}. \tag{2.6.3}$$

Inverting a product flips the operands,

$$(p \odot q)^{-1} = q^{-1} \odot p^{-1},$$

which can be shown using equations 2.6.2 and 2.6.3.

2.6.2 Quaternions as rotations

The *unit* quaternions, i.e. the quaternions q with $|q| = 1$ can be used to represent rotations. The background of why this is possible is beyond the scope of this report, which instead will focus on how they are used. Readers interested in the background theory are recommended to read [8] which discusses, in depth, the theory of the quaternions.

A rotation is defined by two elements: the axis to rotate around and the amount of rotation to perform. A quaternion q describing a rotation by α rad around the unit vector $\vec{n} = \begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix}^T$ is found by

$$q = \begin{pmatrix} \cos\frac{\alpha}{2} \\ \vec{n}\sin\frac{\alpha}{2} \end{pmatrix}.$$

To attain the three dimensional vector $\vec{u}_\mathrm{r}$, representing the vector $\vec{u}$ rotated by the quaternion q, the following formula can be used [2]

$$\begin{pmatrix} 0 \\ \vec{u}_\mathrm{r} \end{pmatrix} = q \odot \begin{pmatrix} 0 \\ \vec{u} \end{pmatrix} \odot q^{-1}.$$

The rotation is performed clockwise looking in the direction pointed by n. For convenience, the 0 padded on the vectors when performing a rotation will not be written out from now on.

It follows that multiplication of two rotation quaternions yields a quaternion describing both of the rotations performed in turn. For example, consider two rotation quaternions p and q. Rotation of a vector $\vec{u}$ by their product $p \odot q$ is given by

$$(p \odot q) \odot \vec{u} \odot (p \odot q)^{-1} = p \odot (q \odot \vec{u} \odot q^{-1}) \odot p^{-1}.$$

In other words, rotation by $p \odot q$ means rotation first by q, then by p.

Often, such as when integrating the output from a gyroscope, a rotation vector $\vec{r} = \begin{pmatrix} r_x & r_y & r_z \end{pmatrix}^T$ is attained. The elements in $\vec{r}$ describe the rotation (in radians) around each of three orthogonal axes in a short time span. In order to create a quaternion describing the same rotation, one can regard the vector $\vec{r}$ as the direction of the rotation axis, and the vector norm $|\vec{r}|$ as the angle of rotation. The quaternion is then simply calculated by

$$q_r = \begin{pmatrix} \cos \frac{|\vec{r}|}{2} \\ \frac{\vec{r}}{|\vec{r}|} \sin \frac{|\vec{r}|}{2} \end{pmatrix}.$$

An interesting problem is how to combine quaternions expressing rotations in different coordinate systems. For example, consider a moving body with an orientation described as a rotation q from the origin in a fixed frame. Given an update of the orientation, expressed as a rotation Δp in the body frame, the orientation q must be updated with this information.

The problem is that when combined, if the first rotation performed is q, then the rotation Δp no longer means the same thing as before because it describes the rotation in a frame that was rotated. Instead, the rotation Δp must be performed first, when the two coordinate systems are aligned with each other. The updated quaternion q_{new} is therefore given by

$$q_{\text{new}} = q \odot \Delta p. \tag{2.6.4}$$

Chapter 3

Sensor evaluation

This chapter is dedicated to discussing the errors of sensors available in the phone. It starts by describing the typical sources of errors found in these sensors and what their effect is. The error descriptions are followed up by measurements of the errors in the actual devices used.

The sensors evaluated are the accelerometer and the gyroscope, since the accuracy of these two sensors dictate how well the navigation system will perform.

For the purpose of determining the behavior of WiFi-signals, the chapter also includes a small WiFi measurement.

3.1 Sensor errors

All sensors are subject to errors which limit the accuracy to which the correct value can be measured. This section describes the errors typically found in the sensors used in the project and how they can be measured. It also discusses how these errors affect the accuracy of the INS.

The measurements from the gyroscope, ω_i can be modeled as follows[18]:

$$\omega_i = K_\omega \cdot \omega_{\text{true}} + b_\omega + N_i + R_i$$

where ω_{true} is the true value that is desired, K_ω is a scale factor, b_ω is a constant bias in the sensor, N_i is a random noise in the sensor and R_i a model for bias instability.

Similarly for the accelerometer measurements a_i,

$$a_i = K_{\text{a}} \cdot a_{\text{true}} + b_{\text{a}} + N_i + R_i$$

where a_{true} is the true acceleration value, K_{a} is the scale factor, b_{a} is the constant bias and N_i and R_i are the same noise processes as above.

3.1.1 Constant Bias

There is typically a constant bias present in the measured signal. The bias is a constant offset from the true value, and is independent of any movement or rotation the sensor might be affected by [14]. The bias can be measured by averaging the signal over a long period of time when the average expected signal is zero. For a gyroscope, that means it is not undergoing any rotation, and for an accelerometer, that it is not accelerating.

For a gyroscope, an uncompensated bias error of a constant b_ω $^\circ/\mathrm{s}$ will, when integrated, cause an error in angle proportional to time:

$$\theta(t) = \int_0^t b_\omega \, d\tau = b_\omega \cdot t.$$

For an accelerometer with a constant bias of b_a $\mathrm{m/s^2}$, double integration gives the change in position, and the bias will cause an error according to

$$s(t) = \int_0^t \int_0^t b_\mathrm{a} \, d\tau d\tau = \int_0^t b_\mathrm{a} t \, d\tau = b_\mathrm{a} \cdot \frac{t^2}{2}.$$

However, once the bias is known it is easy to compensate for it by subtracting the measured bias from the output signal [18].

3.1.2 Random walk

The signal is also affected by thermo-mechanical high frequency white noise. The noise N_i in a sample s_i is a random value with a mean of zero and a certain variance σ^2 and each value is uncorrelated with the others. When the signal is integrated, the noise will affect result as follows.

Gyroscope

The signal from the gyroscope is integrated once. With the rectangular rule the noise will result in the following error over the time $t = \delta t \cdot n$

$$\theta(t) = \int_0^t \epsilon(\tau) \, d\tau = \delta t \sum_{i=1}^{n} N_i$$

The error $\theta(t)$ is a new random variable with the following properties [18]

$$\mathrm{E}(\theta(t)) = \mathrm{E}(\delta t \sum_{i=1}^{n} N_i) = \delta t \cdot n \cdot \mathrm{E}(N) = 0$$

$$\mathrm{Var}(\theta(t)) = \mathrm{Var}(\delta t \sum_{i=1}^{n} N_i) = \delta t^2 \cdot n \cdot \mathrm{Var}(N) = \delta t \cdot t \cdot \sigma^2$$

The standard deviation of angle error grows proportionally to the square root of time. As such, the result is an *Angle Random Walk* (ARW). Manufacturers of gyroscopes usually include an ARW-measurement on the gyro in the data sheet, expressed in the unit degrees/$\sqrt{\text{h}}$.

Accelerometer

For the accelerometer the situation is similar, but worsened by the fact that the signal must be integrated twice to attain the change in position. After integrating once, the result is a *Velocity Random Walk* (VRW) with a standard deviation growing proportionally to $\sqrt{t}$. The VRW measurement is often included in accelerometer data sheets and is expressed in the unit m/s/$\sqrt{\text{h}}$.

The effect of integrating the signal a second time is the following [18]:

$$s(t) = \int_0^t \int_0^t \epsilon(\tau) d\tau d\tau = \delta t \sum_{i=1}^{n} \delta t \sum_{j=1}^{i} N_j = \delta t^2 \sum_{i=1}^{n} (n - i + 1) N_i.$$

The expected error in position and variance are

$$\mathrm{E}(s(t)) = \delta t^2 \sum_{i=1}^{n} (n - i + 1)\mathrm{E}(N_i) = 0$$

$$\mathrm{Var}(s(t)) = \delta t^4 \sum_{i=1}^{n} (n - i + 1)^2 \mathrm{Var}(N_i) \approx \frac{1}{3} \cdot \delta t \cdot t^3 \cdot \sigma^2$$

The white noise in the accelerometer creates a second order random walk with a mean of zero and a standard deviation that grows proportionally to $t^{3/2}$.

3.1.3 Bias instability

Bias instability refers to changes in the bias measurement over a period of time [13]. The bias changes over time due to flicker noise in the electronics, and is usually observed over large periods of time since in short time spans, the white noise is usually much stronger [18]. The bias instability is modeled as a random walk, and can be interpreted as follows:

If the bias at a time t is b_t, then a bias instability of σ over time δt means that at time $t + \delta t$, the bias is random variable with expected value b_t and standard deviation σ [18]. As such, the bias instability is a first-order random walk with standard deviation proportional to $\sqrt{t}$. The effect of the bias instability when integrated is a second-order random walk.

The measurement of bias instability is expressed in deg/h for gyroscopes and m/s/h for accelerometers, and is usually included in the data sheets. To measure the bias instability, the Allan variance, described in Section 3.2.1, can be used.

3.1.4 Scale factor

Scale factor errors refer to mis-calibration of the ratio between the input to be measured and the output signal. Ideally, the proportion between the true value s_{true} and the measured value s_i should be 1, but a mis-calibrated sensor might have the value off by a few percent [14]. In other words, a scale factor K will affect a measured signal s_i as follows

$$s_i = K \cdot s_{\text{true}}$$

When a signal with scale factor K is integrated, the error after integration will be the same factor K.

Measuring the scale factor can be done by affecting the sensor with a fixed input signal for a period of time while measuring the output signal. After subtracting the constant bias, the resulting signal can be divided by the expected signal to determine the scale factor K.

3.1.5 Temperature

The bias of the sensors also change as their internal temperature changes. This can be partly due to changes in the air temperature around the sensor but mostly because the sensor heats up as it is being used. The changes in bias due to temperature is often highly non-linear [18]. The sensors are usually equipped with a thermometer which makes it possible to perform corrections for the temperature.

As described in 3.1.1, a constant bias will, when integrated once, cause an error proportional to time and, when integrated twice, an error proportional to time squared.

3.2 Testing

This section is dedicated to measuring the accuracy of the sensors in the IMU, namely the accelerometer and gyroscope. The information of interest is of course the current direction of movement and the velocity the phone is moving at, provided at a reasonable frequency. The goal is to conclude how the sensors can be used to create a navigation system as robust as possible. It is important to note that the tests conducted here are specific for the sensors in the phone used. While other sensors of the same model would typically yield results of similar magnitude, the exact number would not be equal.

The easiest test to perform on the sensors is sampling it continuously for a period of time when the phone is lying still, since the correct value is usually a known constant. Long logs are preferable, since they might reveal changes over long periods of time that would not be seen otherwise. Typically, the idle

measurements conducted in this thesis will be made by logging for around ten hours over night in the office, which minimizes the noise from external sources due to less activity in the building.

It is important to be wary of the preconditions of an idle measurements as they can also influence the outcome. For example, whether the phone was just switched on or if it has been on for a long time might influence the temperature of the sensor which in turn might affect the output. The current level of charge of the battery might affect the voltage delivered to the sensor which can also affect the output. As such, it is important to be wary of the preconditions when conducting the tests and if possible try to restrain them to those of a typical use case of the finished system. For all tests described here, it is assumed that the phone is almost completely charged and the sensors have been at rest for at least an hour.

3.2.1 Allan variance

As described in Section 3.1, the sensor measurements are affected by various random noise processes. A way to describe the characteristics of these noise processes is needed in order to understand the accuracy of the sensor. For example, when measuring changes in the bias, it is not obvious how to compare the bias at different times. If data was collected for an hour of time, comparing the bias during the first t minutes with the bias during the last t minutes is one way, but how does one choose the number t?

A systematic approach for this exists. The Allan variance is an IEEE-adopted technique[1] for measuring sensor noise. The name is from its originator, Dr David Allan, who developed the technique when measuring the accuracy of atomic clocks. It is applied to a collection of data that is expected to be constant. The Allan variance $\sigma_y^2(\tau)$ is a function of an averaging time τ and is calculated as follows.

1. Divide the data into a number of bins N, where each bin contains data corresponding to the averaging time τ.
2. Calculate the average $y_i(\tau)$ of the data in each bin i.
3. Compute the differences of the averages of all successive bins and square the number, $(y_{i+1}(\tau) - y_i(\tau))^2$.
4. Add all these values together and rescale by multiplying with a number based on the number of bins, $\frac{1}{2 \cdot (N-1)}$.

Mathematically, the Allan variance can be expressed as such:

$$\sigma_y^2(\tau) = \frac{1}{2(N-1)} \sum_i (y_{i+1}(\tau) - y_i(\tau))^2$$

The accompanying Allan deviation σ_y is defined as

$$\sigma_y = \sqrt{\sigma_y^2}$$

The interesting part comes when combining the Allan deviation of different averaging times and showing them together in a log-log scale plot. In such a plot, different kinds of noise appear as different asymptotes in the plot. Specifically, a random walk introduced by white noise will show up in the Allan deviation plot as an asymptote with slope -0.5. Bias instability due to random flickering in the sensor shows up as a flat region between the slopes. In order to get a numerical reading of the ARW (for gyroscopes) or VRW (for accelerometers), one can fit a line through the measured points that form the downward slope and read its value at $\tau = 1$. The bias instability can be measured by reading the minimum value on the Allan variation plot. Refer to the IEEE standard [1] or [5] for a explanations of why the values can be read this way.

Characterizing the sensor noise is also of use for another purpose, namely generating simulated sensor signals. With enough information of the sensor noise, a very similar noise can be generated by a computer system. If accurate enough, algorithms can be tested on our own produced signals rather than those measured by the actual sensors. This is good for many reasons, one is that generating huge data sets for stress tests is much easier than having to run around manually and collect the same data. Another is that with control over the sensor noise, it is possible to study the different parts of the system individually. For example, how would the system perform if the accelerometer used was better, but using the same gyroscope? What if a much better gyroscope was used?

When examining our sensors, the Allan variance method will be used to examine their noise.

3.2.2 Accelerometer

The accelerometer used was a model BMA150 from Bosch Sensortec[1]. It has an acceleration measurement range of $\pm 4\,\mathrm{g}$ and 10 bit resolution, resulting in a sensitivity of approximately $0.078\ \mathrm{m/s^2}$. The data sheet specifies it has a constant bias of between -150 mg and 150 mg at 25 °C, or between $\pm\ 1.5\,\mathrm{m/s^2}$. Additionally, it has a temperature drift of $1\,\mathrm{mg/K} \approx 1\,\mathrm{cm/s^2/K}$.

As explained in Section 2.1.1, the accelerometer measures the acceleration of the phone and the force of gravity. For example, when the phone is lying still on its back, the total acceleration measured should be $9.81\,\mathrm{m/s^2}$ in the Z-axis, calculated by 0 (acceleration of the phone) minus the gravity ($-g \approx -9.81\,\mathrm{m/s^2}$).

[1] http://www.bosch-sensortec.com

Logging the accelerometer output when the phone is lying still can be used for two purposes. First, since the output is expected to be constant (no acceleration is applied to the phone), the Allan variance technique can be used to determine the characteristics of the noise in the sensor.

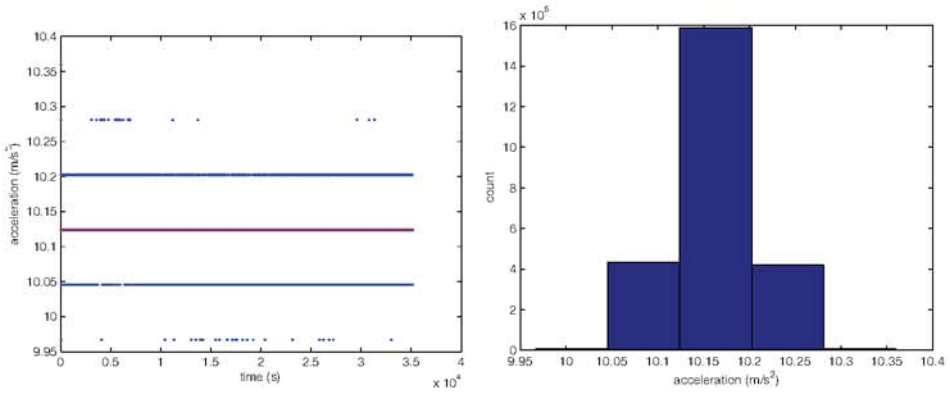
Second, since the only force affecting the phone is the force of gravity, which is a known constant (g in the global Z-axis), the scale factor error can be calculated. Positioning the phone in such a way that two axis are perpendicular to gravity will result in all the gravitational force being applied to one axis. An average output different to g means the axis has a scale factor error.

Measuring the constant bias in the accelerometer is difficult, since there is no position the accelerometer can be placed in that will make the desired value zero. Instead, the phone would have to be in free fall for a longer time which is impractical for obvious reasons.

Idle, Z axis

The phone was left idling for approximately 10 hours, placed on its back, attempting to align it such that the force of gravity would affect only the Z axis, and not the X and Y axis. During sampling, 2,441,915 samples were gathered, corresponding to an average frequency of 69.5 Hz.

Figure 3.1 shows the output of the Z axis from the accelerometer from approximately 10 hours of logging. Table 3.1 summarizes the results for all axes.



(a) Output versus time (b) Histogram of output

Figure 3.1: *The output of the Z axis of the accelerometer from 10 hours of logging, with the phone positioned on its back. The red line shows the mean value.*

The X and Y axis contributions to the total magnitude of the accelerometer signal is low, and the scale factor error of the Z axis, $K_{a,z}$, can be approximated by

$$K_{a,z} = \frac{10.1236}{g} = \frac{10.1236}{9.81} \approx 1.0320.$$

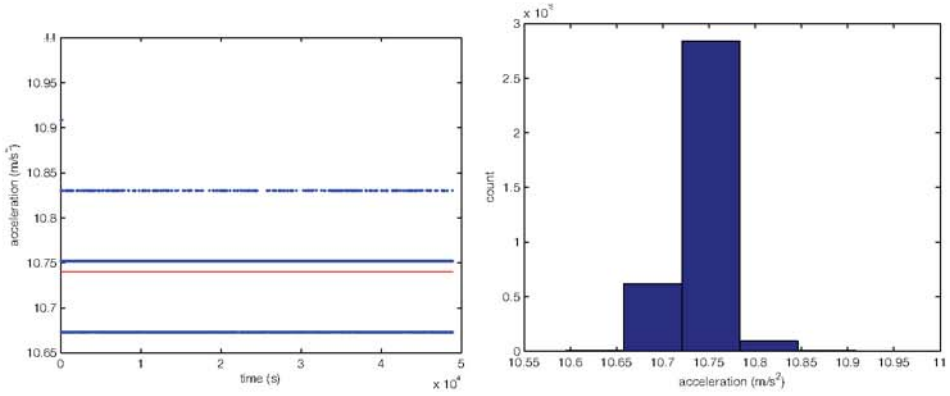
	Mean (m/s^2)	Std. dev
X Axis	0.5567	0.0320
Y Axis	0.7582	0.0468
Z Axis	10.1236	0.0472

Table 3.1: *A summary of the output from the accelerometer when the phone was idling on its back for approximately 18 hours*

Idle, X axis

The setting was similar to the previous test, only the phone was placed on its left side so that the X axis was aligned with gravity. During logging, 3,540,011 samples were gathered, an average sampling frequency of 72.4 Hz.

Figure 3.2 and Table 3.2 summarize the results from the approximately 14 hours of logging.



(a) Output versus time (b) Histogram of output

Figure 3.2: *The output of the X axis of the accelerometer from 14 hours of logging, with the phone positioned on its left side. The red line shows the mean value.*

	Mean (m/s^2)	Std. dev
X Axis	10.7402	0.0330
Y Axis	0.4848	0.0493
Z Axis	0.4265	0.0463

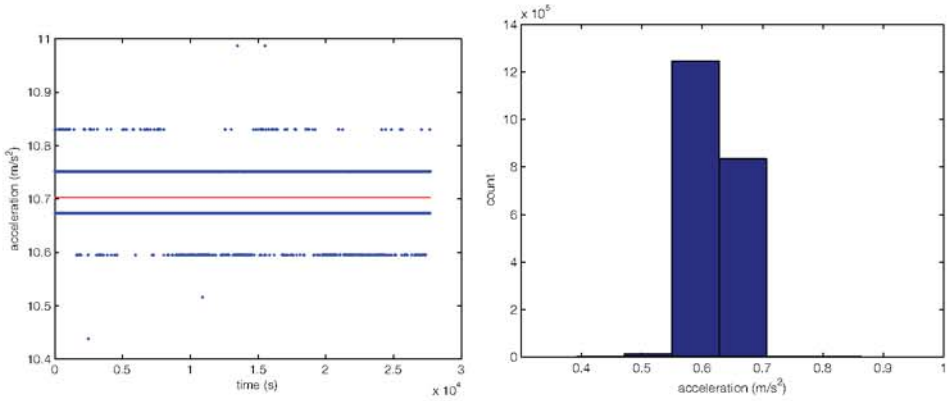
Table 3.2: *The table summarizes the output from the accelerometer when the phone was idling on its left side for approximately 14 hours*

Using the same method for approximating the scale factor yields

$$K_{a,x} = \frac{10.7434}{g} = \frac{10.7434}{9.81} \approx 1.0951.$$

Idle, Y axis

In the final test, the phone was standing up on its feet, in order to align the Y axis with gravity. 2,087,715 samples were gathered in approximately 8 hours, an average of 75.3 Hz. The results are shown in Figure 3.3 and Table 3.3.



(a) Output versus time (b) Histogram of output

Figure 3.3: *The output of the Y axis of the accelerometer from approximately 8 hours of logging, with the phone positioned on its feet. The red line shows the mean value.*

	Mean (m/s^2)	Std. dev
X Axis	0.6588	0.0393
Y Axis	10.7010	0.0447
Z Axis	0.7039	0.0490

Table 3.3: *The table summarizes the output from the accelerometer when the phone was idling on its feet for approximately 8 hours*

The scale factor of the Y axis can be found by

$$K_{a,y} = \frac{10.6463}{g} = \frac{10.6463}{9.81} \approx 1.0908.$$

Noise characteristics

In order to measure the noise in the accelerometers, the Allan variance measurement is used. From each of the idle logs above, the data for the axis aligned with gravity was analyzed, in order to make sure the conditions were the same for each axis. Figure 3.4 shows the Allan deviation plot, and the numerical readings of the noise values can be found in Table 3.4.

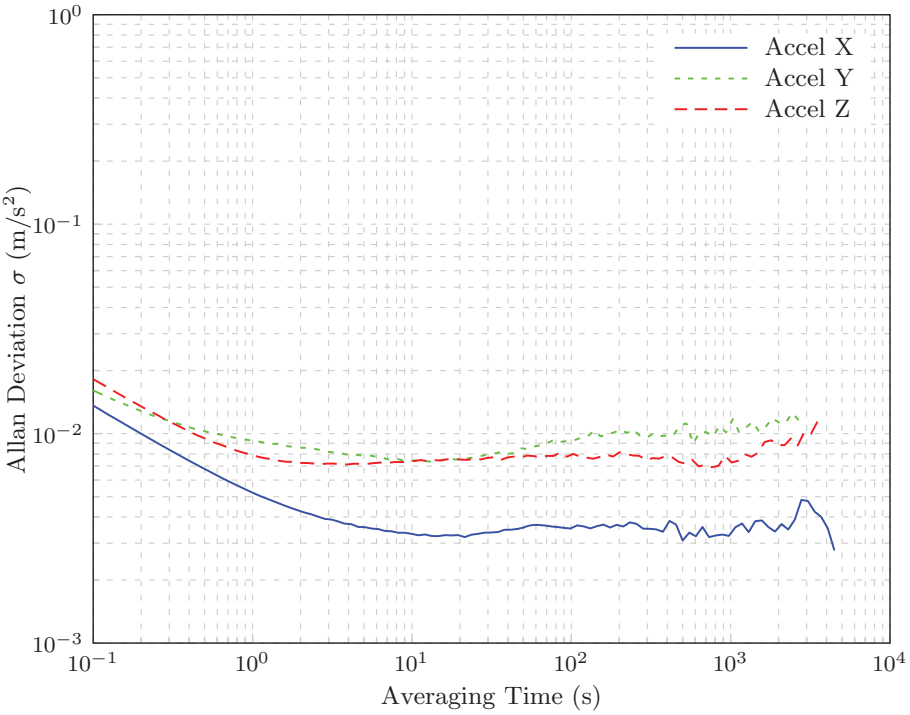


Figure 3.4: *Allan deviation plot for the accelerometer*

	Bias Instability ($m/s^2/s$)	Random Walk ($m/s^2/\sqrt{s}$)
X Axis	0.0028 (at 4478 s)	0.0057
Y Axis	0.0074 (at 13 s)	0.0097
Z Axis	0.0069 (at 763 s)	0.0090

Table 3.4: *Accelerometer noise measurements. The bias instability measurement is the lowest value of the Allan deviation. The random walk measurement is the Allan deviation value at $t = 1\,\mathrm{s}$.*

Accelerometer results overview

To conclude, the scale factor of the accelerometer was calculated to

$$K_\mathrm{a} = \begin{pmatrix} K_{\mathrm{a,x}} & K_{\mathrm{a,y}} & K_{\mathrm{a,z}} \end{pmatrix}^T = \begin{pmatrix} 1.0951 & 1.0908 & 1.0320 \end{pmatrix}^T .$$

Compensating for the scale factor is trivial – simply dividing the measurement values of an accelerometer axis by its scale factor yields a more accurate acceleration.

The noise in the sensor was analyzed and the results are summarized in Table 3.4. Knowledge about the performance of the sensor will be crucial when determining the overall accuracy of the navigation system.

3.2.3 Gyroscope

The gyroscope used was a L3G4200D by ST Microelectronics[2]. The L3G4200D has a range of $\pm 2000\,\mathrm{deg/s}$ and 16 bit resolution, resulting in a sensitivity of approximately $0.070\,\mathrm{deg/s}$. The data sheet includes an ARW measurement of $0.03\,\mathrm{deg/s}/\sqrt{\mathrm{Hz}}$, and specifies a temperature drift of $\pm 0.04\,\mathrm{deg/s/K}$.

This section includes measurements to determine the constant bias, scale factor error and noise characteristics of the gyroscope.

Idle measurement

For a gyroscope, a measurement of the sensor in idle can be used for two purposes. First, since the gyroscope is lying still and not rotating, the expected output is zero. Any offset from zero is therefore a bias in the sensor, and the mean value from a long idle log is the constant bias from the sensor. Second, a long idle log can be used to determine the characteristics of the noise in the sensor, by using the Allan variance method discussed in Section 3.2.1.

In the experiment, the phone was placed on its back and left idling for approximately 12 hours, yielding $9,451,383$ samples, an average of 207 Hz. The results are shown in Figure 3.5 and summarized in Table 3.5. Figures (a), (c) and (e) shows the output plotted versus time, showing that there are no major offsets during the experiment. Figures (b), (d) and (f) shows the distribution of the output values. It is clear that the output is centered around the mean values presented in Table 3.5, which are the constant bias values of this gyro.

	Mean (deg/s)	Std. dev
X Axis	−1.4720	0.1563
Y Axis	2.3168	0.1642
Z Axis	0.6774	0.1900

Table 3.5: *A summary of the output from the gyroscope when left idling for approximately 12 hours*

Noise characteristics

To determine the characteristics of the sensor noise, the Allan variance method is used.

Figure 3.6 shows the Allan deviation plot of the idle gyroscope measurement, for all three axes. Numerical readings from the plot are provided in Table 3.6.

The bias instability is the minimum value of the Allan deviation, and the random walk measurement is the value at $t = 1$.

[2] http://www.st.com

	Bias Instability (deg/s)	Random Walk (deg/$\sqrt{s}$)
X Axis	0.0038 (at 174 s)	0.0185
Y Axis	0.0053 (at 98 s)	0.0191
Z Axis	0.0035 (at 81 s)	0.0224

Table 3.6: *Gyroscope noise measurements. The bias instability measurement is the lowest value of the Allan deviation. The random walk measurement is the Allan deviation value at $t = 1$ s.*

Rotation measurements

In order to determine the scale factor errors of the gyroscope, the phone would ideally be rotated by a known angular velocity in each axis while the output of the gyro was logged. Determining the scale error would then be simple by first subtracting the constant bias and compare the output to the angular velocity used to rotate the phone.

Unfortunately, the equipment needed to rotate by a fixed velocity was not available in the experiment, so an alternative method was used. The phone was placed on a rotary tripod, carefully trying to place it so that the X-axis of the gyro would align with the rotational axis of the tripod. The tripod was then rotated 10 full turns (3600 degrees) clockwise, followed by 10 full turns counter clockwise.

The accumulated rotation around the Y and Z-axes was small (less than 100 degrees) in comparison to the X-axis (> 3600 degrees), meaning that the rotation almost exclusively involved the X-axis of the phone. After the 10 full 360 degree turns, the X-axis had accumulated a total rotation of 3778 degrees, indicating a scale error of

$$K_{\omega,x} = \frac{3778}{3600} \approx 1.049$$

The same technique was used to determine the scale factor error in the Y and Z-axes, yielding the scale factors

$$K_{\omega,y} = \frac{3836}{3600} \approx 1.066 \qquad K_{\omega,z} = \frac{3626}{3600} \approx 1.007$$

Gyroscope results overview

The constant bias in the 3 axes of the gyroscope sensor (in deg/s) is

$$b_\omega = \begin{pmatrix} -1.4720 & 2.3168 & 0.6774 \end{pmatrix}^T.$$

In theory, these values can easily be subtracted from the measured angular velocities before performing further calculations. It was expected that the bias in the gyroscope would be constant enough that measuring it once would be

enough. However, it was found that the bias changes over periods significantly longer than 12 hours, and as a result must be re-measured for a short time before each usage of the gyroscope.

The scale factor of the gyroscope was determined to be

$$K_\omega = \begin{pmatrix} K_{\omega,\mathrm{x}} & K_{\omega,\mathrm{y}} & K_{\omega,z} \end{pmatrix}^T = \begin{pmatrix} 1.049 & 1.066 & 1.007 \end{pmatrix}^T.$$

Compensating for the scale factor errors is as simple as dividing measured values from one axis with the corresponding constant (after removing the drift).

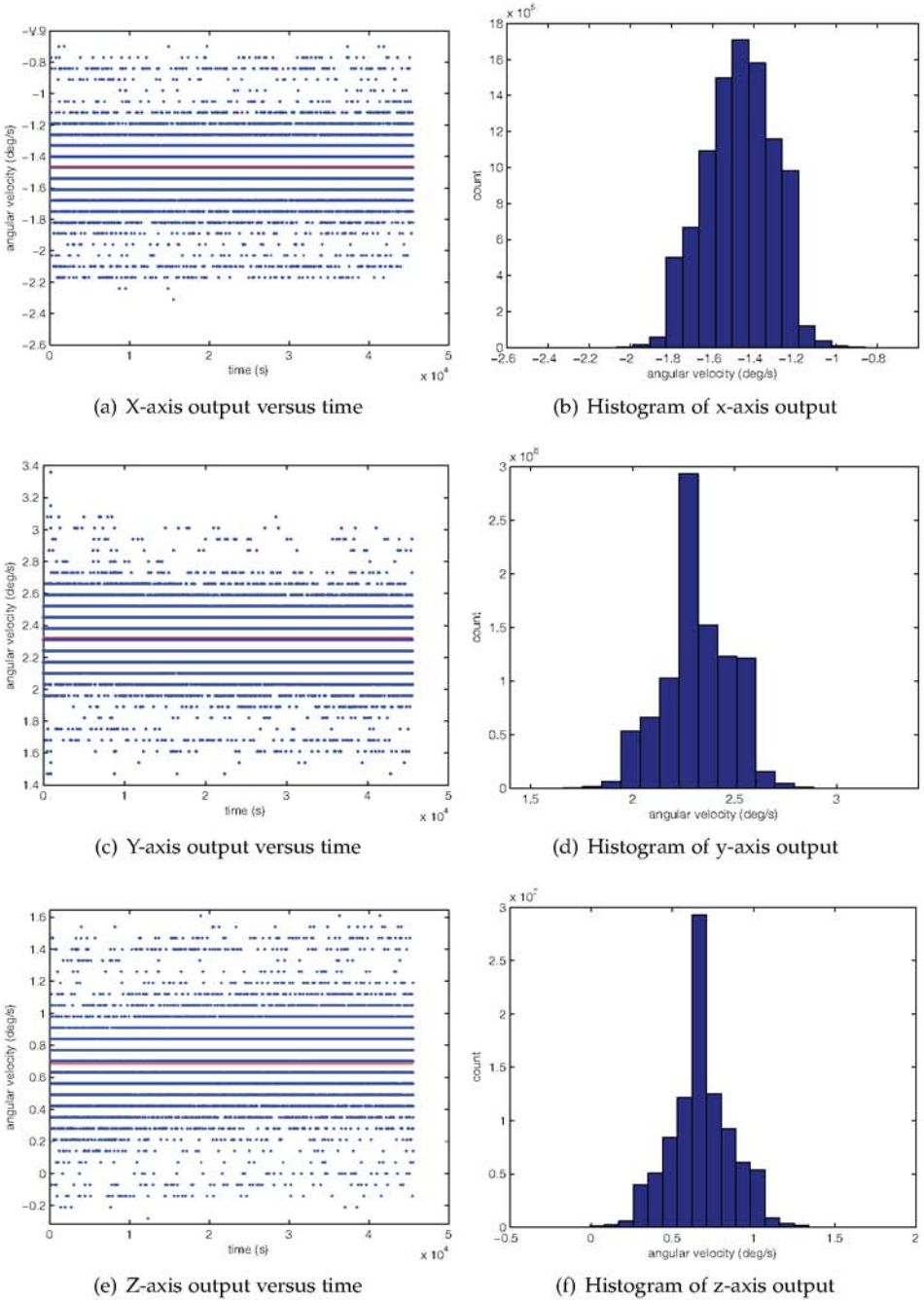
3.2.4 WiFi measurements

In order to determine the characteristics of the RSSI measurements received from WiFi scanning, measurements were collected for two hours of scanning while the phone was lying still. The measurements were collected during normal office hours, with people moving around in the vicinity of the phone. While this makes the results somewhat random, the environment was similar to an environment where the system is expected to be used. The phone was positioned in such a way that it had direct line of sight to one AP (LOS), obstructed line of sight to a second AP (OLOS), and no line of sight to a third AP (NOLOS). The results are shown in Figure 3.7, both as RSSI values over time and as histograms of measured RSSI values. The resolution of the measured RSSI values is 1 dBm.

The histograms reveal that the measurements fit well with a normal distribution. Table 3.7 summarizes the mean values and standard deviation of the measurements.

	Mean RSSI (dBm)	Std. dev
LOS	-58.5	1.9
NOLOS	-89.8	1.6
OLOS	-79.8	2.2

Table 3.7: *The results of logging WiFi RSSI values from three different access points for two hours.*



(a) X-axis output versus time

(b) Histogram of x-axis output

(c) Y-axis output versus time

(d) Histogram of y-axis output

(e) Z-axis output versus time

(f) Histogram of z-axis output

Figure 3.5: *The output of all three gyroscope axes when the phone was lying flat on its back, idling for approximately 12 hours. The red line shows the mean value.*

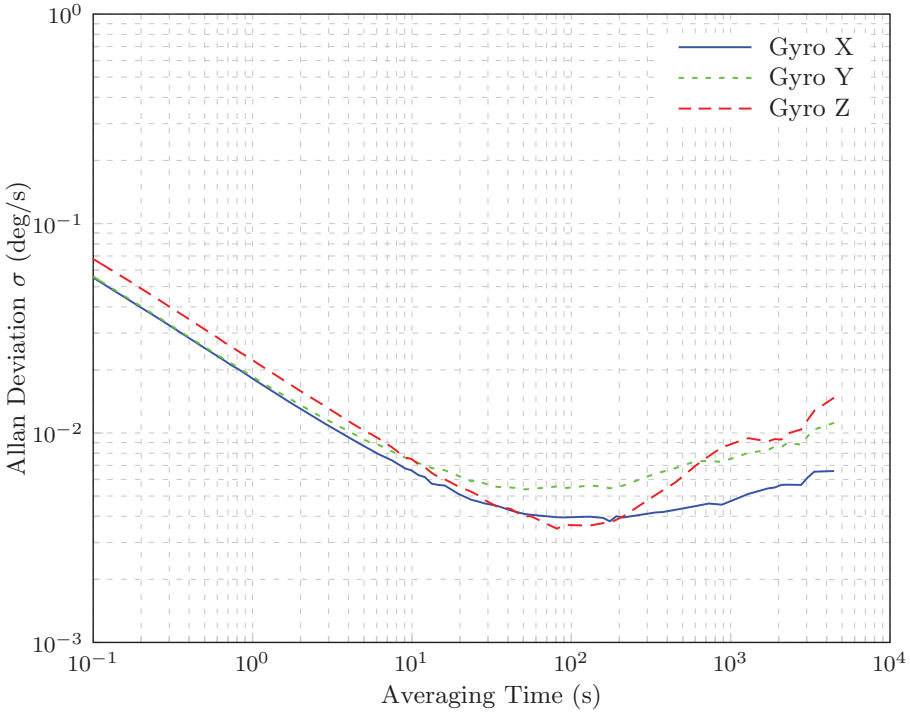
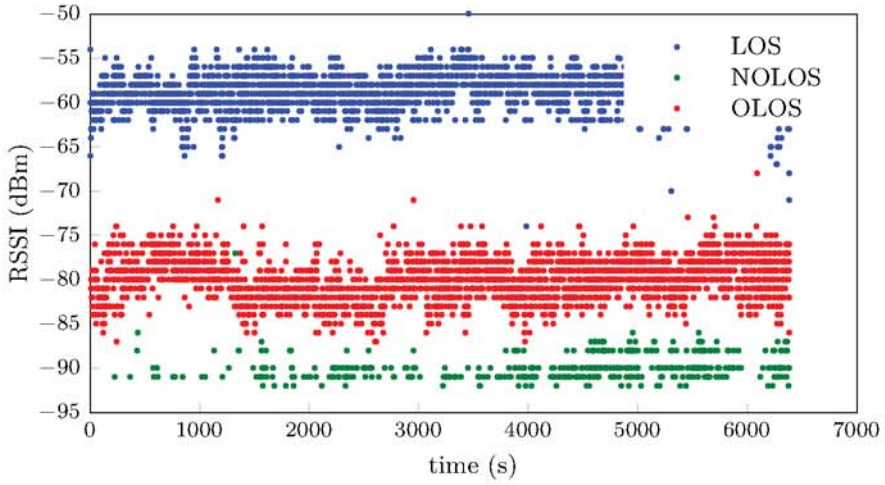
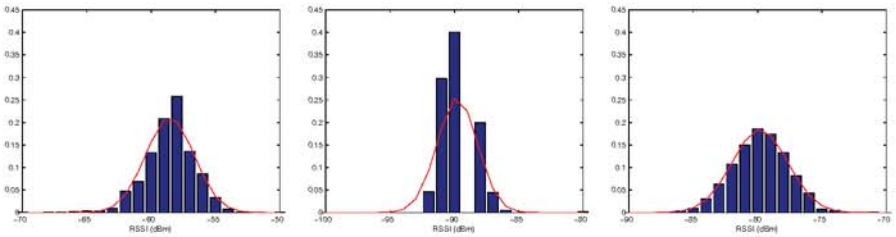


Figure 3.6: *Allan deviation plot*



(a) The RSSI values over time



(b) Histogram of LOS measurements

(c) Histogram of NOLOS measurements

(d) Histogram of OLOS measurements

Figure 3.7: *The output of all three gyroscope axes when the phone was lying flat on its back, idling for approximately 12 hours*

Chapter 4

Inertial Navigation System

An Inertial Navigation System (INS) estimates the position and orientation of a moving body by continuously tracking the output from a number of sensors in an Inertial Measurement Unit (IMU) attached to the body. The sensors in the IMU typically include at least a three axis gyroscope and a three axis accelerometer, but it can also contain other sensors such as magnetometers and barometers. This chapter describes the idea behind how INS works and presents results of simulations using signals similar to the sensors examined in Chapter 3.

The details of how the INS is implemented is different to the finished system developed in this thesis, described in Chapters 7 and 6. Because the important part of this chapter is the results of the simulations, it will focus on discussing those results rather than describing the details of orientation and distance tracking.

4.1 Description

An INS tracks the orientation of the moving body by integrating the output signals from the sensors in the system. To determine the transliteration of the body, the accelerometer is used. Since the accelerometer output is given in the body frame, the orientation of the body must be tracked in order to translate the output to a fixed frame (see Chapter 2 for a discussion of the various frames of reference).

Another reason why orientation must be known is to be able to remove the force of gravity affecting the accelerometer. As we will see, it is crucial for the accuracy of the navigation system to subtract the gravity in the right direction.

Tracking the orientation is done using the dynamics described in [3, p. 344], which uses quaternions. When the orientation (described by the quaternion

q) is known, an accelerometer measurement $\vec{a}^b = \begin{pmatrix} a_x^b & a_y^b & a_z^b \end{pmatrix}^T$ in the body frame can be transformed to the world frame by the rotation

$$\vec{a}^w = q \odot \vec{a}^b \odot q^{-1}.$$

Given an initial position $\vec{p}^w(0)$ and an initial velocity $\vec{v}^w(0)$, the position $p^w(t)$ can be tracked by integrating the output

$$\vec{v}^w(t) = \vec{v}^w(0) + \int_0^t \vec{a}^w(\tau) - \vec{g}^w d\tau$$
$$\vec{p}^w(t) = \vec{p}^w(0) + \int_0^t \vec{v}^w(\tau) d\tau.$$

Because the gravity vector $\vec{g}^w \approx \begin{pmatrix} 0 & 0 & 9.81 \end{pmatrix}^T$ m/s^2 is being subtracted from the accelerometer measurement $\vec{a}^w(t)$, it is important that $\vec{a}^w(t)$ is properly transformed to the world frame. If not transformed correctly, subtracting the gravity vector will not remove all of the gravitational effect in the accelerometer measurement. The result is a component of gravity that still points in the xy-plane of the world frame.

Because a random walk is introduced in the orientation estimation, the error of the orientation will typically not be centered around zero, but will drift in a random direction. Due to this effect, the erroneous acceleration in the xy-plane from gravity will point roughly in the same direction in each update. Combined with the fact that the force of gravity is a quite strong, a small error in the orientation estimate is enough to cause a substantial component left in the xy-plane.

As an example, consider a constant orientation error of 1 degree around the y-axis and a perfect accelerometer measurement. A perfect translation to the world frame would yield a vector

$$\vec{a}^w_{\text{ideal}} = \vec{g}^w = \begin{pmatrix} 0 & 0 & 9.81 \end{pmatrix}^T,$$

but due to the error in orientation, we attain the vector

$$\vec{a}^w = 9.81 \cdot \begin{pmatrix} \sin 1^\circ \\ 0 \\ \cos 1^\circ \end{pmatrix} = \begin{pmatrix} 0.1712 \\ 0 \\ 9.8085 \end{pmatrix} \text{ m/s}^2.$$

Starting from a velocity of zero, after 30 seconds the velocity will be

$$\vec{v}(30) = \vec{0} + \int_0^{30} \begin{pmatrix} 0.1712 \\ 0 \\ 9.8085 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 9.81 \end{pmatrix} d\tau = \begin{pmatrix} 5.1362 \\ 0 \\ -0.0448 \end{pmatrix} \text{ m/s}.$$

The position drift from the origin after 30 seconds is given by

$$\vec{p}(30) = \vec{0} + \frac{\vec{v}(30)}{2} \cdot 30 = \begin{pmatrix} 77.0436 \\ 0 \\ -0.6723 \end{pmatrix} \text{ m}.$$

In conclusion, an orientation error of just 1 degree around the y-axis causes a position drift of over 70 m after only 30 seconds. Clearly, very accurate sensors are needed to attain a meaningful position by integrating the output signal.

4.2 Simulations

This section determines, by simulations, how accurate an INS equipped with the sensors examined in Chapter 3 would be. There are two advantages of using simulated signals instead of the real signals from the sensors. First, it is easy to run a large number of tests without the need to manually collect all the data. Second, when simulating the noise in the sensors, it is possible to selectively remove various noise processes and analyze what actually causes the errors to grow.

4.2.1 Noise parameters

In order to simulate signals with similar noise to the actual sensors, the parameters of the underlying noise process must be determined. The table does not include the scale factor and constant bias measurement conducted. The noise process are the following

- **White Noise:** Each sample from the sensors is conceptually affected by a Gaussian white noise. The noise N_i in sample i is a drawn from a zero mean normal distribution with variance σ^2_{wn}.
- **Bias Instability:** Each sample i is also affected by a bias instability modeled as a random walk R_i. R_i is given by the equation

  $$R_i = \sum_{k=1}^{i} N_k,$$

  where N_k is random variable drawn from a normal distribution with mean 0 and variance σ^2_{bi}.

Scale factor errors and constant bias are not included, because once measured, these error sources can easily be removed by simple arithmetic. Therefore, including these errors would not improve the error modeling.

In Chapter 3, noise data was collected for the accelerometer and gyroscope. The random walk and bias instability measurements must be converted to σ values of the underlying noise processes. This was done using the methods described in [18].

The σ for the white noise sequence is given by converting the random walk measurement RW from Tables 3.4 and 3.6 using the following formula

$$\sigma_{\text{wn}} = \frac{\text{RW}}{\sqrt{\delta t}},$$

where δt is the sampling period of the device. Because our device does not have a constant sampling period, the average sampling period during the logging was used.

The bias instability measurements from Tables 3.4 and 3.6 were converted to σ values using the formula

$$\sigma_{\text{bi}} = \sqrt{\frac{\delta t}{t}} \cdot \text{BI},$$

where t was the averaging time where the BI was conducted.

Table 4.1 summarizes the sigma values used for the various noise processes after conversion from the Allan variance measurements.

	White noise σ_{wn}	Bias Instability σ_{bi}
Gyro X (deg/s)	0.185	$0.288 \cdot 10^{-4}$
Gyro Y (deg/s)	0.191	$0.535 \cdot 10^{-4}$
Gyro Z (deg/s)	0.224	$0.389 \cdot 10^{-4}$
Accel X (m/s^2)	0.057	$0.042 \cdot 10^{-4}$
Accel Y (m/s^2)	0.097	$2.052 \cdot 10^{-4}$
Accel Z (m/s^2)	0.090	$0.250 \cdot 10^{-4}$

Table 4.1: *Sigma values used for the underlying noise processes in the INS simulation.*

4.2.2 Simulation results

Figure 4.1 shows the average position drift from 100 simulations of the INS system. After 15 seconds, the average recorded drift was almost 50 m, and after 30 seconds, it was over 250 m. Clearly, the errors grow faster and faster and quickly become large enough for the INS position to become unusuable.

The figure also shows the results of simulating what would happen if one of the sensors was noise free, and the other sensor was as noisy as before. The green line representing a noisy gyro but a perfect accelerometer closely follows the blue line, where both sensors are noisy. This experiment shows that even if the accelerometer is perfect, the gyro will cause the position to drift almost as quickly as before.

The results are similar when the gyroscope is perfect but the accelerometer is noisy. Even though the error does not grow as quickly as for the gyroscope, it still suffers from quadratic growth and quickly drifts away. For example, after 2 minutes, the average drift in simulations for a perfect gyro and noisy accelerometer is over 500 meters.

In order to test the validity of the simulations, it was necessary to test the INS on signals from the real device. Figure 4.3 shows the results of these tests. The

behavior is similar, showing the same quadratic trend as the simulations. The error does not grow quite as quickly on the real device as in the simulations, but the results are comparable. On the real device, it takes 30 seconds to reach an error of over 130 meters, while it takes 23 seconds in the simulations. While the results show that the noise from the real device was not perfectly modeled in the simulations, the simulations still demonstrate an important point.

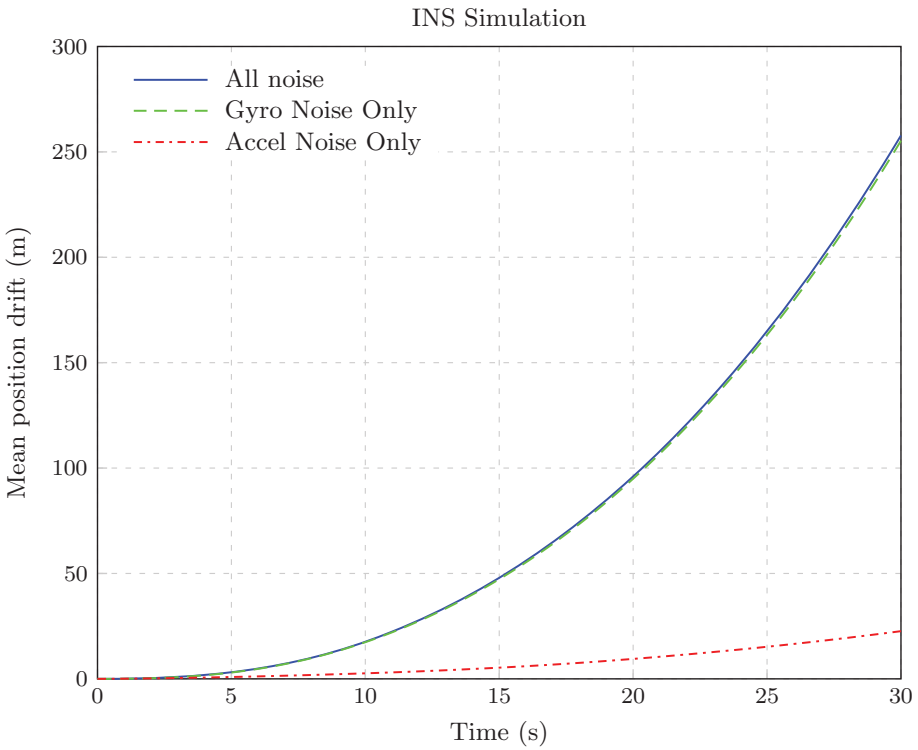


Figure 4.1: *The mean position drift from 100 simulations of the simulated INS system. The red line shows the mean drift if a perfect gyroscope was used with a noisy accelerometer, and the green line shows the mean position drift if a perfect accelerometer but a noisy gyro is used.*

A test was conducted to determine approximately how good the sensors must be in order to efficiently track the phone position for a longer time. Given a requirement that the drift can be no larger than 5 meters on average after 120 seconds, it was found that the error measurements in the sensors must be reduced by a factor of $1/2000$. Figure 4.2 shows the result of this simulation. The quadratic trend is still present and the error will still grow beyond acceptable limits quickly after the first 120 seconds, even with sensors significantly better than those currently available.

It is natural to wonder if there was some kind of fix that could be applied when the velocity estimate drifts out of control. One way to achieve fixes is to impose constraints on the inertial navigation system. One such constraint is that the sensors must be mounted on the foot of the person navigating.

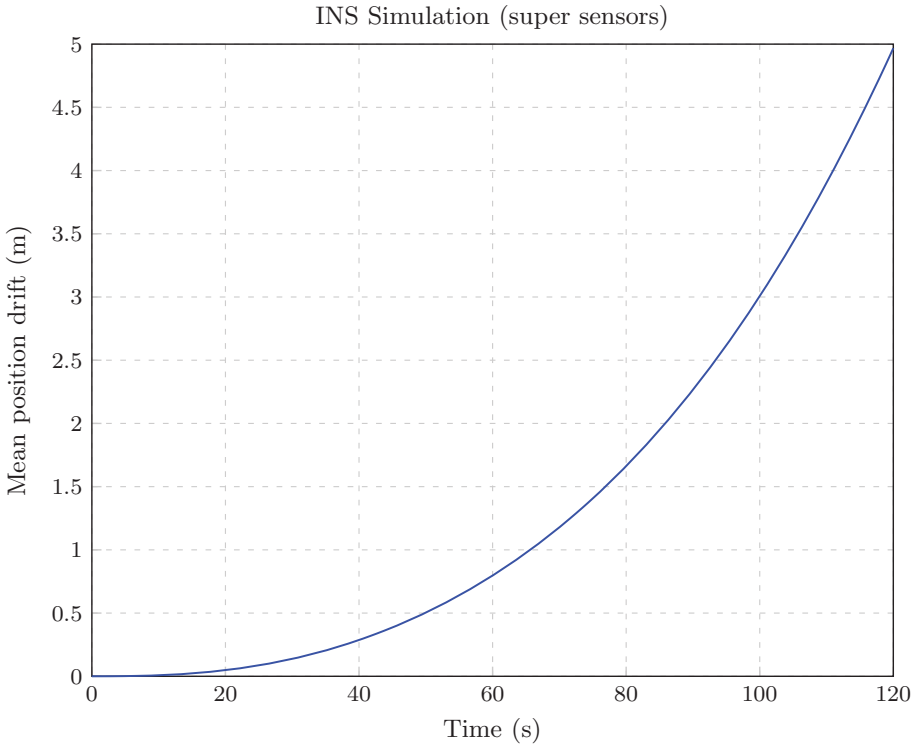


Figure 4.2: *The mean position drift from 100 simulations of an INS system using sensors with errors of size* 1/2000 *of those found in the sensors of the phone. While the position error is less than five meters on average after two minutes, the same fast-growing trend is still present.*

When a person is walking, it is possible to detect when a foot is in stand still, for example using one of the techniques evaluated in [12]. When the foot is standing still, the velocity estimate can be updated by a *Zero-Velocity Updates (ZVUs)*, which sets the velocity to 0 m/s.

It is also possible to apply fixes to the orientation when the foot is in stand still, since the foot is usually tilted the same way in every step. See for example [17] and [11] for examples of systems using a foot mounted IMU to decrease drift. In our simple simulations, the results of applying ZVUs and orientation fixes once per second is shown in Figure 4.4. While the improvements are large, foot mounted INS can be made even better as demonstrated in [19], creating a system that achieves an average drift of less than 5 cm after 120 seconds of walking.

For the purpose of this thesis, where mounting the IMU on the foot is a constraint too severe for practical use, different techniques than those described here must be used to track position.

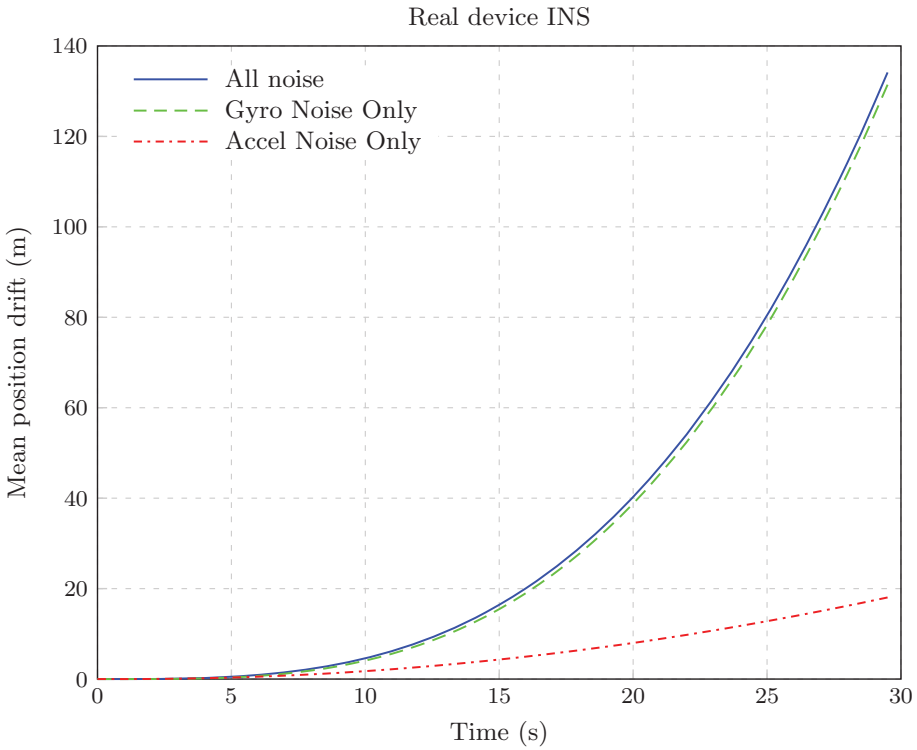


Figure 4.3: *The mean position drift from 10 runs of an INS system running on the real device, with selective error sources removed.*

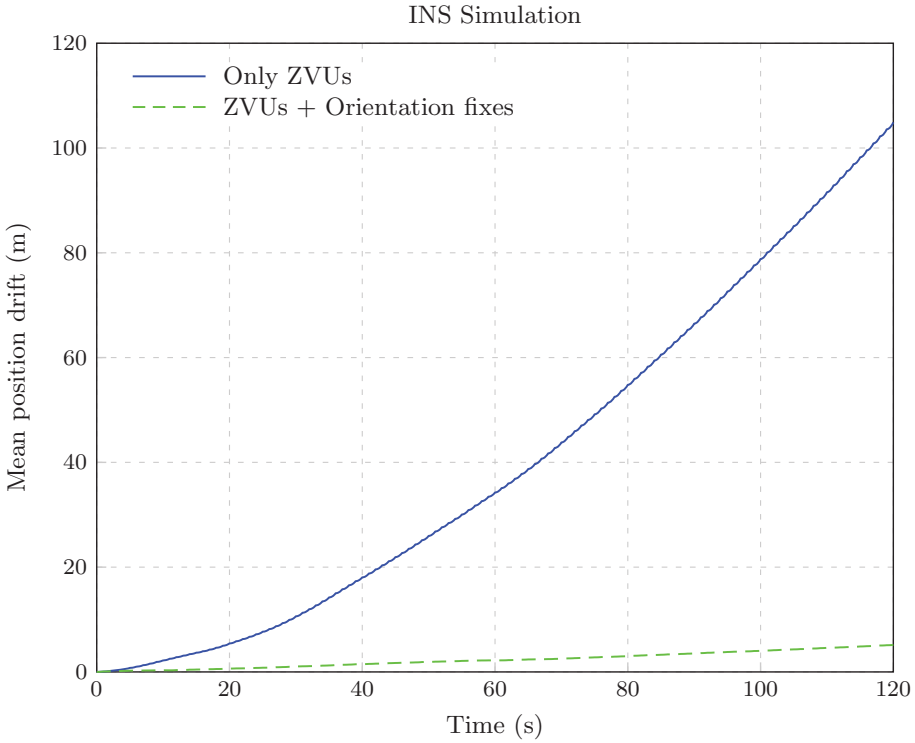


Figure 4.4: *The results of simulating a naive foot mounted INS, correcting the velocity (blue solid line) and velocity+orientation (green dashed line) once per second. Note that the time scale is different to the previous plots.*

Chapter 5

Definition of Local Frame

As mentioned in Section 2.4, a local frame is often used for indoor navigation. One reason is discussed is that maps might be rotated from the world frame to include

5.1 Motivation

In order to evaluate the accuracy of the developed system, it is necessary to have an accurate measurement of the true values. For instance, when navigating, it is useful to know exactly where the user was walking in order to compare it to the estimate of the system.

There is a need for a well defined coordinate system. Several coordinate systems exist for this reason, e.g. WGS84, ITRF and ETRS89[10]. Typically, they use longitude and latitude pairs to uniquely identify different points on earth. Using one of these, however, has several drawbacks for our uses.

First, it is difficult to determine the true coordinates of a point at a certain location when inside a building without consulting external tools such as Google Maps[1] and perform some sort of estimation.

Second, latitude and longitude coordinates are not intuitively easy to interpret, since they are defined as fractions of the earth radius. For a frame as small as a single building, it is more convenient to work with the meter system.

Third, the building in which this work was conducted has a perfect $0.5\,\mathrm{m} \times 0.5\,\mathrm{m}$ grid layout in the floor, which if incorporated in our coordinate system would make it much easier to keep track of the current position.

[1] http://maps.google.com

5.2 Definition

A new two dimensional coordinate system was defined. The two axes of the coordinate system are roughly aligned with the x and y axes of global 3D coordinate system defined in Section 2.3, only slightly rotated. As such, the x axis points east and slightly north, and the y-axis north and slightly to the west.

The unit of the coordinate system is meters, $1\,\text{unit} = 1\,\text{m}$. The origin was chosen so that it would be positioned south west of the building, ensuring that all points inside the building had positive coordinates which simplifies processing. More specifically, the coordinate $(50, 50)$ was aligned perfectly with a floor grid tile close to the desk where the work was conducted which made it easy to navigate. Figure 5.1 shows a map of the office building with the coordinate system included. The tiles in the figure matches the tiles on the building floor.

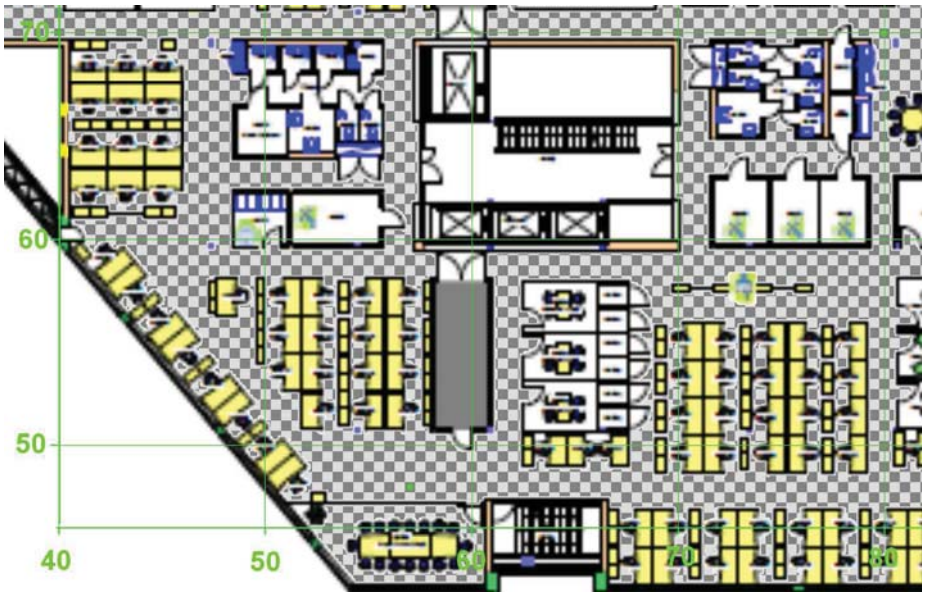


Figure 5.1: *A map of the office building where the work was conducted, including the coordinate system defined in Chapter 5.*

Chapter 6

Distance tracking

This chapter discusses a method for determining the distance the phone is moved. As seen in Chapter 4, the accelerometer can be used to track the movement of the phone, but is not accurate enough to simply be integrated twice to attain the distance moved.

However, the accelerometer does give an indication of when the phone is moving, and adding the fact that it is being carried around by humans can help when analyzing the signal. Humans typically move around by walking, which is a very predictable movement since practically every step consists of the same pattern. Assuming that the length of a step can be measured or estimated, tracking the number of steps should be a viable method for determining the distance the phone moves.

6.1 Accelerometer output when walking

Figure 6.1 shows the output from the accelerometer when the phone was located in the front left jeans pocket of a 178 cm tall man during a walk. The phone was positioned such that the Y axis of the phone was roughly aligned with the Z axis of the world. In other words the force of gravity is mostly seen in the Y axis of the accelerometer.

By looking at the figure it is easy to see that there is a periodicity in the output signal, corresponding to the steps taken while walking. It can also be seen that the mean output from the X and Z axis are roughly 0 m/s^2 while the Y axis has a mean of roughly 10 m/s^2, corresponding to the force of gravity. As such, the output of each axis is highly dependent on the orientation of the phone, which may change between different uses or users.

It is therefore useful to look at the total magnitude m of the output signal, rather than each individual axis. The magnitude, or norm, of an accelerometer

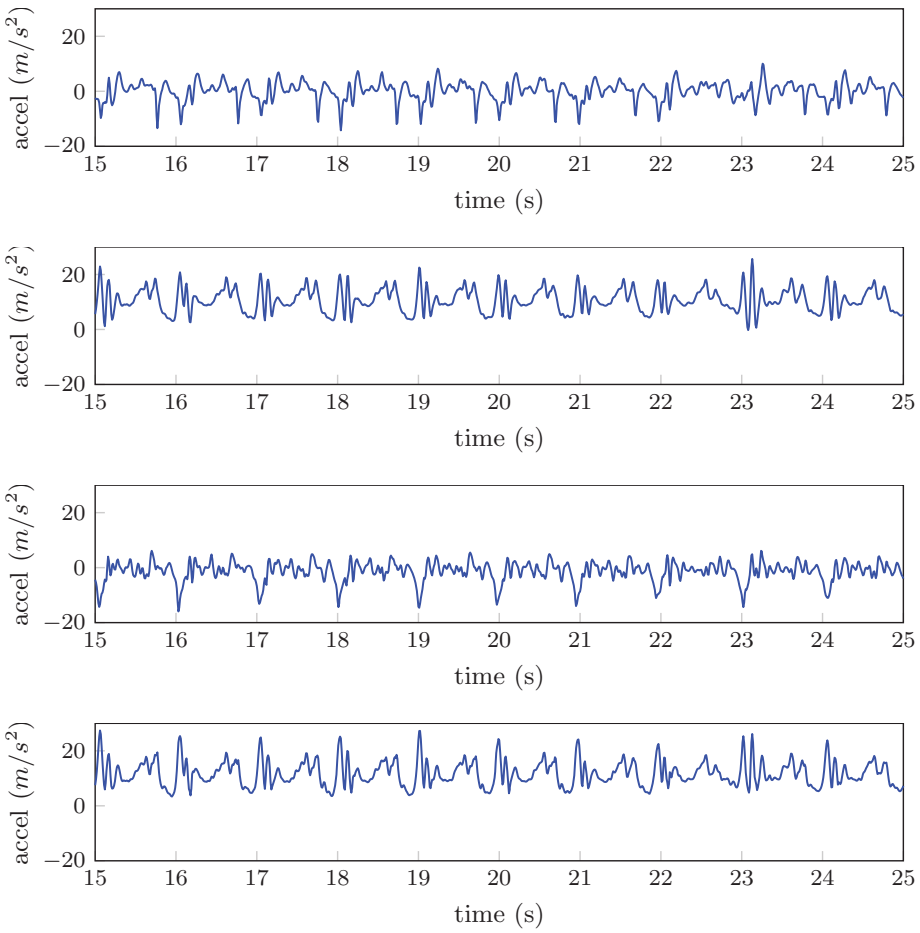


Figure 6.1: *The output from the X, Y, Z-axis and the combined magnitude of accelerometer while the phone was placed in the front left jeans pocket of a walking man.*

measurement a is defined as its length,

$$m = |a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

The magnitude of the accelerometer output is also visible in Figure 6.1.

6.2 Eliminating noise

The output from the accelerometer is very noisy and contains frequencies higher than the frequency of the steps occurring. Since the only information of interest in the signal is the number of steps, filtering the data with a low-pass filter makes the signal easier to analyze. In order to create a good filter,

a few things need to be known about the signal.

First, it must be decided what the frequency of interest is. A reasonable assumption is that the user will not walk faster than two steps per second, i.e. 2 Hz. Therefore, the low-pass filter should have a cut-off frequency of no less than 2 Hz, or it might dampen the step frequency.

Second, a sampling frequency must be chosen. A practical problem arising here is the fact that the Android platform delivers data from its sensors in an event based fashion, at non-constant time intervals. Since the filter assumes constant time intervals, the signal must be re-sampled with a constant frequency. In Section 3.2.2, it was found that the device delivers accelerometer values at roughly 70 Hz on average. Any frequency above the Nyquist frequency $2 \cdot 2\,\text{Hz} = 4\,\text{Hz}$ is good enough to find the target frequency. For the purposes described, 20 Hz was decided to be a reasonable trade off between performance and battery consumption. Figure 6.2 shows the result of re-sampling the signal.

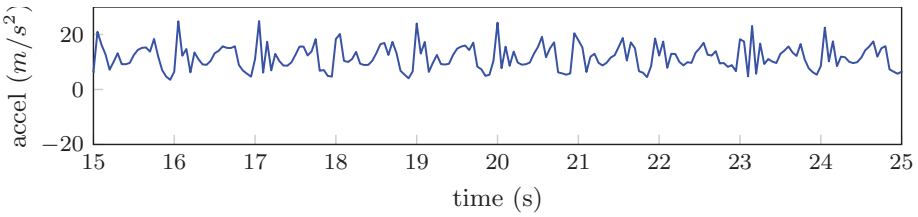


Figure 6.2: *The magnitude of the signal in Figure 6.1 after re-sampling.*

A higher order filter dampens frequencies above the cut-off frequency faster, but is also more costly to implement. A 4th order Butterworth filter quickly dampens frequencies above the cut-off frequency and works well for the purposes described. Figure 6.3 shows the signal in Figure 6.1 re-sampled and filtered.

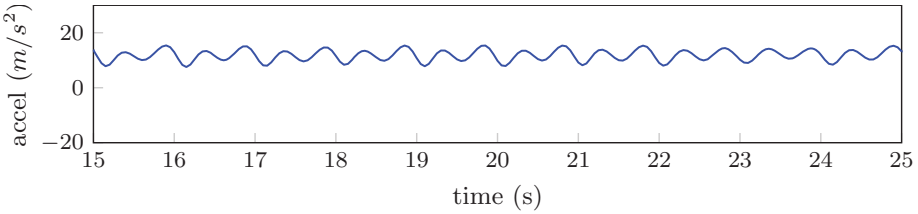


Figure 6.3: *The accelerometer output after filtering with a low pass filter.*

6.3 Different use(r)s

In Figure 6.3 it is easy to see the step events, but judging what constitutes as a step has to be made by programming code. An intuitive solution would be to

use a threshold for the magnitude, which when crossed would result in a step event. However, Figure 6.4 shows the filtered output from the accelerometer from another walk, when instead of being placed in the jeans pocket the phone was placed in bag carried by the walking person.

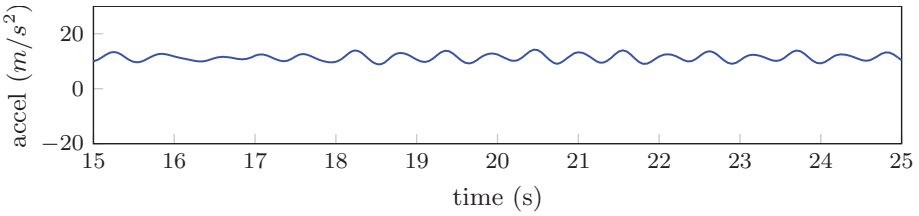
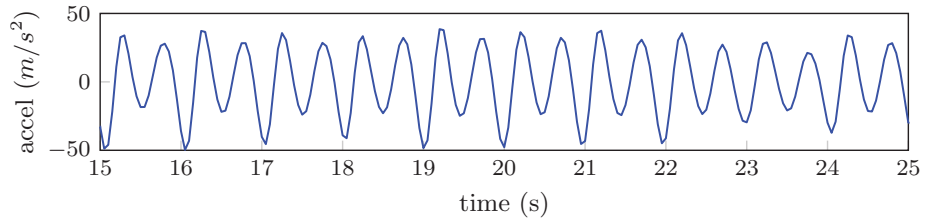


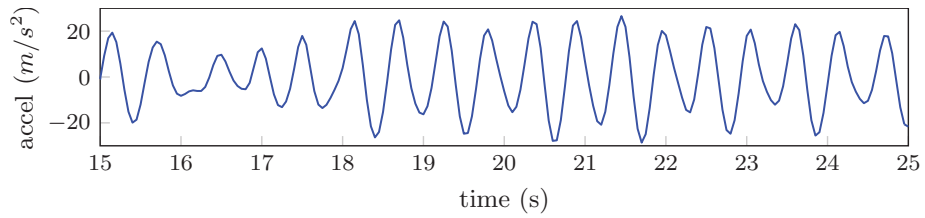
Figure 6.4: *The filtered magnitude from a walk when the phone was placed in a handbag.*

The figure shows two things. First, the average value output from the accelerometer is slightly higher than when the phone was placed in a jeans pocket (Figure 6.3). Second, the steps vary in magnitude, with every second step creating a slightly lower peak in the plot than the others. The conclusion is that using a single threshold value for the magnitude to judge when a step occurs is difficult.

A more robust method is to focus on the slope of the curve. Differentiating the signal yields a new signal that will be centered around 0 and alternate between positive and negative. Figure 6.5 shows the signals from Figures 6.3 and 6.4 after differentiation.



(a) The derivative of the signal in Figure 6.3.



(b) The derivative of the signal in Figure 6.4.

Figure 6.5: *Derivatives of the accelerometer signals..*

6.4 Algorithm

The only thing remaining is to determine from the differentiated signal whether a step has occurred or not. A threshold value is used to determine when a movement is intense enough to be considered a step. Figure 6.6 shows the idea: whenever a value above the threshold value is followed by a value below the threshold, a step has occurred. In mathematical notation, if the i:th sample from the accelerometer is denoted $y(i)$, and the differentiated signal is defined as such

$$y'(i) = \frac{y(i) - y(i-1)}{t(i) - t(i-1)}.$$

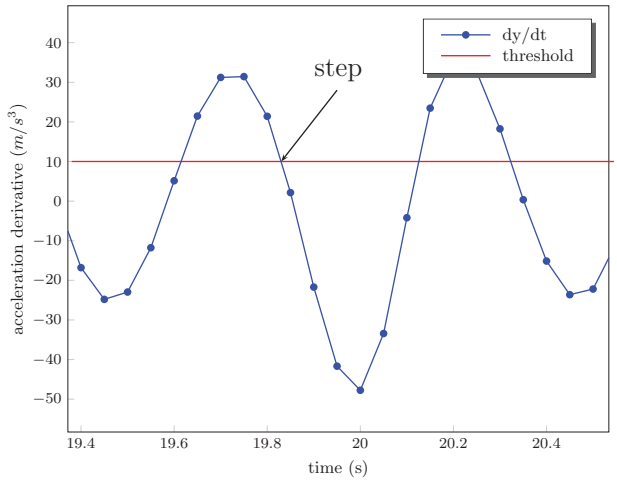


Figure 6.6: *Step detection on the derivative of the filtered magnitude*

Whether or not a step has occurred given the threshold T is determined by

$$step(i, T) = \begin{cases} \text{true}, & \text{if } y'(i-1) - T > 0 \ \& \ y'(i) - T < 0 \\ \text{false}, & \text{otherwise} \end{cases}$$

6.5 Step length considerations

Assuming the number of steps is known, determining the distance moved is a matter of determining how long each step taken was. Mathematically, for n steps with lengths $d_1, d_2, ..., d_i, ...d_n$, the total distance moved, D, can be calculated by

$$D = \sum_{i=1}^{n} d_i$$

Determining the step length by looking at the accelerometer output is difficult, because integrating the output from the accelerometer leads to the same problems as when constructing an INS system (discussed in Chapter 4).

It is easier to estimate the length of a step by a constant, d, corresponding to the average step length of the current user of the system. There are two problems with this. First, the average step length must be known and calibrated. This can be done, for example by having each user walk a fixed distance once, and divide this distance with the number of steps needed to complete the walk. Second, as noted in [9], the stride of a person is anything but constant. Step lengths of a person varies and is influenced by a number of factors such as the pace of the walk, the inclination of the terrain and even shoes worn.

For a system aided with more information more sophisticated methods can be used to determine the step length. For example in [7], the IMU is mounted on one of the user's feet, which when walking behave like a pendulum. Observing the accelerometer output in such a situation can make it more apparent how long a step is.

When outdoors, counting steps while observing the GPS location can be used to calibrate the length of the steps. See for example [9] for examples of such a system.

For the system described in this report, a constant step length is used that corresponds roughly with the average step lengths of the two authors (0.9 m). For a commercial system, one of the methods described above would be used to determine the users step length.

6.6 Results

The step counter was evaluated in three different tests. In all tests, the user was randomly walking around without stopping on a single floor, alternating right and left turns with walking straight. In the first test, one of the authors held the phone still against his chest while walking around. 200 steps were taken, and the step counter recorded exactly 200 steps during the walk.

Two more tests were conducted, one with the phone placed in the jeans pocket and one from a different test person walking with the phone against his chest. In both tests, 200 steps were taken, and the step counter reported 200 and 201 steps, respectively.

These results show that counting the steps of a naturally walking pedestrian is a simple task. The errors in the distance estimation will therefore be mostly due to badly calibrated step length estimations.

Chapter 7

Orientation tracking

Using step counting in combination with a known step length it is possible to tell how far someone has moved. This information is useless for positioning unless the heading of the person is known.

To keep track of the heading of the person carrying the phone it is assumed that the relationship between the person and the phone is constant during navigation. If the phone orientation is constant, relative to orientation of the person, any changes in the heading of the person will be reflected in the phone orientation. As such, the heading of the person is determined by tracking the orientation of the phone.

7.1 Gyroscope tracking

To keep track of the phone orientation, a gyroscope is used. The gyroscope measures the angular velocities of the phone expressed in the coordinate system of the phone. Since the heading must be expressed in a fixed coordinate system to be useful, for example as an angular offset from north, the rotation must also be converted to a fixed coordinate system, such as the world coordinate system described in Section 2.3. The relationship between the phone coordinate system and the world coordinate system is established before navigation begins.

Additionally, the phone orientation does not necessarily align with the heading of the user. For example, the phone might be placed vertically in a pocket or be held in a hand. It is therefore necessary that the heading of the user is known at the start of navigation.

The procedure is as follows:

1. At the start of the navigation the person must be facing north in the local frame.

2. The phones orientation is locked relative to the body of the person navigating. The offset between orientation of the person and orientation of the phone is stored as a rotation quaternion q_{p2p}.
3. The orientation of the phone in the world coordinate system is determined (described below in Section 7.1.1).
4. The relationship between the phone orientation and the coordinate system used for navigation is stored.
5. The orientation of the phone is tracked continuously by integrating the gyro. The heading of the person is determined by looking at the difference between the current phone orientation and the phone orientation at the start.

7.1.1 Initial orientation

It is possible to find the orientation of the phone in the world using the accelerometer and the magnetometer. Two physical quantities that always affect the phone, and are fixed in world coordinate system are measured in order to establish the relationship between the coordinate system of the phone and the world coordinate system. These are the geomagnetic field and the gravitational force of the earth.

If the phone is not accelerating in any direction, it will only measure the gravitational force of the earth. This provides a vector, $\vec{g}^b$, pointing along the z-axis of the world coordinate system ($\vec{z}_{\text{w}}$), expressed in phone coordinates.

The magnetometer measures the strength and direction of the magnetic field in phone coordinates. If not affected by disturbances, the only magnetic field measured is the geomagnetic field, which is a vector, $\vec{m}^b_{\text{w}}$, that points towards the magnetic north pole. The vector $\vec{m}^b$ is a linear combination of the positive world y-axis, $\vec{y}^b_{\text{w}}$, and the world z-axis, $\vec{z}^b_{\text{w}}$. The inclination of the geomagnetic field changes with the latitude and the z component of $\vec{m}^b$ is negative in the northern hemisphere and positive in the southern hemisphere. Figure 7.1 shows an illustration of the earths magnetic field.

The axes of the world coordinate system can be expressed in body coordinates using these two vectors. First, the world z-axis points in the same direction as gravity. It can easily be calculated by normalizing the gravity vector

$$\vec{z}^b_{\text{w}} = \frac{\vec{g}^b}{||\vec{g}^b||}$$

In order to determine the x-axis, consider the cross product of the gravity vector and the magnetometer vector. The result is a vector orthogonal to both, corresponding to the world x-axis expressed in phone coordinates.

$$\vec{x}^b_{\text{w}} = \frac{\vec{m}^b \times \vec{g}^b}{||\vec{m}^b \times \vec{g}^b||}$$

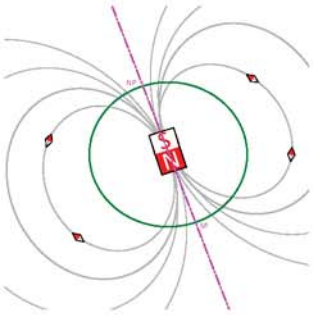


Figure 7.1: *An illustration of the earths magnetic field. The inclination of the field is positive in the southern hemisphere and negative in the northern hemisphere*

Finally, to determine the y-axis, $\vec{y}_w^b$, consider the cross product of the world z-axis and the world x-axis. The result is the world y-axis, orthogonal to both and pointing towards the magnetic north pole, expressed in phone coordinates.

$$\vec{y}_w^b = \vec{z}_w^b \times \vec{x}_w^b$$

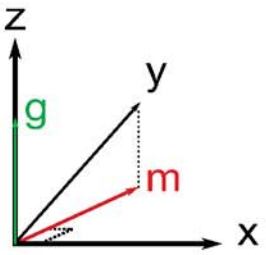


Figure 7.2: *The x-, y- and z-axis of the world coordinate system can be expressed in the body coordinate system by measuring $\vec{m}$ and $\vec{g}$.*

7.1.2 Integrating the gyroscope

The orientation of the phone at time t is represented by a rotational offset from the start orientation. The rotation is described by a quaternion, $q_{wb}(t)$, describing the rotation of the body frame relative to the local frame at time t.

To update the orientation, the rotational change must be tracked by observing at the output from the gyroscope. The output from the gyroscope is a vector

$$\vec{\omega}^b(t) = \begin{pmatrix} \omega_x^b(t) & \omega_y^b(t) & \omega_z^b(t) \end{pmatrix}^T,$$

the rotation speed of the phone in the 3 axes of the body coordinate system, at time t. In order to determine the rotational change, the output must be

integrated. Integration is done using trapezoid integration, yielding a vector $\vec{r}^b(t)$ describing the rotational change since the previous output sample at time t_0

$$\vec{r}^b(t) = \int_{t_0}^{t} \vec{\omega}^b dt = (t - t_0)\frac{\vec{\omega}^b(t) + \vec{\omega}^b(t_0)}{2}.$$

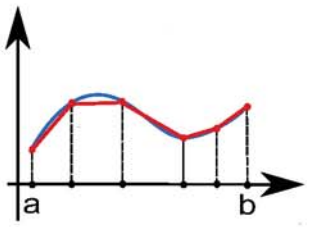


Figure 7.3: *Trapezoid integration. The blue line shows the actual function while the red line is the estimate when using trapezoid integration*

In order to update the previous phone orientation (represented by a quaternion $q_{\text{wb}}(t_0)$), a new quaternion, $\Delta q(t)$, that describes the rotation change is created. The quaternions describe rotations in different coordinate systems; $q_{\text{wb}}(t)$ is a rotation in the world coordinate system and $\Delta q(t)$ is a rotation in the phone coordinate system. As shown in Section 2.6, the updated $q_{\text{wb}}(t)$ can be calculated by first performing the local rotation and then performing the former global rotation. Mathematically,

$$q_{\text{wb}}(t) = q_{\text{wb}}(t_0) \odot \Delta q$$

7.1.3 Finding the heading

The relation between the phone orientation and heading of the person is determined before navigation starts, and stored as a rotation quaternion q_{p2p}. Using the orientation of the phone, $q_{\text{wb}}(t)$, and the rotation q_{p2p}, it is easy to determine the heading. Multiplying $q_{\text{wb}}(t)$ by q_{p2p} yields a new quaternion $q_{\text{heading}}(t)$ that describes the change in orientation that the person has performed since the start of the navigation.

The heading should be a vector in the world xy-plane, meaning $q_{\text{heading}}(t)$ should describe a rotation solely around the z-axis. However, any change in orientation that the phone has undergone in the other two axes (due to noise or otherwise) will result in $q_{\text{heading}}(t)$ representing rotation in these axes as well.

At the start of navigation, the heading of the person is aligned with the y-axis of the local frame, y_{l}. Rotating y_{l} by $q_{\text{heading}}(t)$ yields a new vector. The heading $\theta(t)$ is found by ignoring the z-axis component of this vector and only looking at the angle offset from y_{l} in the xy-plane.

The angle offset of the rotated y-axis is the same as the angle offset of the

rotated x-axis, which is easily calculated using the atan2 function.

$$h(t) = \begin{pmatrix} h_x & h_y & h_z \end{pmatrix} = q_{\text{heading}}(t) \odot x_1 \odot q_{\text{heading}}(t)^{-1}$$
$$\theta(t) = \text{atan2}(h_y, h_x).$$

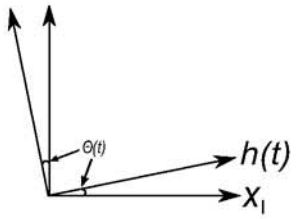


Figure 7.4: *The heading of the user is the same as the angular offset of the rotated x-axis, h(t).*

7.2 Magnetometer

Relying solely on integrating the gyroscope to determine the orientation is problematic since errors accumulate over time. The orientation updates provided by the gyroscope are all relative to the previous orientations. The magnetometer provides a vector m^b pointing towards the earths magnetic north pole. This vector can potentially be used to provide an absolute orientation of the phone, since it is independent of previous measurements.

As shown in Section 7.1.1, the absolute orientation of the phone can be found using the two vectors $\vec{a}^b$ and $\vec{m}^b$, measured by the accelerometer and magnetometer, respectively. This requires that $\vec{a}^b$ and $\vec{m}^b$ can be measured correctly at a given time.

However, indoors, the geomagnetic field is usually not the only magnetic field affecting the phone. For example, ferromagnetic materials such as metal bars are found in the floor and walls, and electronic devices create magnetic fields often stronger than the geomagnetic field. This causes the magnetic field to be inhomogeneous and unpredictable indoors. Figure 7.5 shows an example of the directions of the magnetic vectors in a room with disturbances.

The remainder of this section discusses the possibility of using magnetometer and accelerometer data to aid the orientation estimate.

7.2.1 Detecting disturbances

Given a magnetometer measurement $\vec{m}^b(t)$, a decision has to be made whether the measurement is believed to be affected by a disturbance or not.

The method presented here is based on the idea that it is known approximately what $\vec{m}^b(t)$ should be. To achieve this, the true magnetic field vector

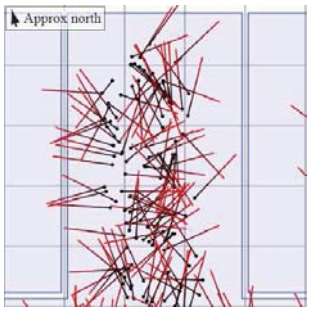


Figure 7.5: *An example of how the magnetic field might be pointing in a room with disturbances. [19]*

in world coordinates, $\vec{m}^w_{\text{true}}$ is measured just before navigation starts. It is possible to measure $\vec{m}^w_{\text{true}}$ because the phone orientation is correct since no drift has occurred yet, and it is assumed that the magnetometer is not affected by disturbances at the initial position. Later, during navigation, a magnetometer measurement $\vec{m}^b(t)$ is compared to the true magnetic field vector in body coordinates, $\vec{m}^b_{\text{true}}$. The vector $\vec{m}^b_{\text{true}}(t)$ can be calculated using the current orientation of the phone, $q_{\text{wb}}(t)$ and $\vec{m}^w_{\text{true}}$,

$$\vec{m}^b_{\text{true}}(t) = q_{\text{wb}}(t) \odot \vec{m}^w_{\text{true}} \odot q^{-1}_{\text{wb}}(t). \tag{7.2.1}$$

When the vectors are compared and found not equal this is due to one of two reasons.

1. The magnetometer is currently affected by a disturbance, and $\vec{m}^b(t)$ is incorrect.
2. The magnetometer measurement $\vec{m}^b(t)$ *is* correct, but the current phone orientation estimate is incorrect, and as a result $\vec{m}^b_{\text{true}}$ is incorrect.

$\vec{m}^b(t)$ should be used to aid the orientation estimate in case 2, but not in case 1. To achieve this, it is determined whether the two vectors $\vec{m}^b(t)$ and $\vec{m}^b_{\text{true}}(t)$ are "close" to each other. If they are close to each other, then $\vec{m}^b(t)$ is *not* currently affected by a magnetic disturbance, but it is the phone orientation that is slightly wrong. The rationale behind this reasoning is that the magnetic field of the earth is very weak in comparison to other magnetic fields typically found indoors. As a result of this, $\vec{m}^b(t)$ will be very different from $\vec{m}^b_{\text{true}}$ in the case of a disturbance.

As a measurement of closeness, the norm of the difference is used

$$n = |\vec{m}^b(t) - \vec{m}^b_{\text{true}}(t)|$$

$$close = \begin{cases} \text{true} & n < \alpha \\ \text{false} & n \geq \alpha \end{cases}$$

The norm measurement captures errors in both magnitude and distance. In the methods discussed below, various values of α have been tested and evaluated.

7.2.2 Correcting the orientation estimate

If the magnetometer vector, $\vec{m}^b$, is a correct measurement of the magnetic field, then the difference between $\vec{m}_b$ and $\vec{m}^b_{\text{true}}$ is caused by errors in the gyroscope signals.

To correct the drift one can calculate a quaternion that expresses the rotation of the body frame in such a way that $\vec{m}^b$ and $\vec{m}^b_{\text{true}}$ become aligned with each other. However, since there is only one restriction on the orientation of the body frame, there is an infinite number of rotations that achieves this. Below follows discussions of various rotation techniques.

The shortest rotation

Consider n, the normalized cross product between $\vec{m}_b(t)$ and $\vec{m}^b_{\text{true}}(t)$. n is the normal of the plane in which the vectors $m^b(t)$ and $m^b_{\text{true}}(t)$ reside. The shortest possible rotation, ϕ, is a rotation around this vector and is calculated by

$$n = \frac{\vec{m}^b(t) \times \vec{m}^b_{\text{true}}(t)}{||\vec{m}^b(t) \times \vec{m}^b_{\text{true}}(t)||}$$
$$\phi = \arccos(\vec{m}^b(t) \cdot \vec{m}^b_{\text{true}}(t))$$

Using this method, there are no restrictions to around which axis the rotation will be performed. As a result, in our simulations, this method repeatedly destroys the phone orientation estimate of the world z-axis. The result is that true rotations in the world z-axis (such as when the user makes a turn) becomes wrong and the error grows rapidly.

Measuring the world z-axis

When the phone is stationary, it is possible to determine its orientation using the accelerometer and magnetometer as described in Section 7.1.1. However, when the user is walking, the phone is constantly changing acceleration, as seen in Chapter 6 where the accelerometer signal is used for step detection. Therefore, during navigation, the accelerometer signal does not represent the world z-axis and the orientation can not easily be measured.

An idea would be to low pass filter the accelerometer signal which could yield a vector representing the world z-axis. This approach would suffer from the

same shortcomings of dealing with the posture changes described in the next paragraph.

Using the initial world z-axis

The orientation of the phone, relative the person, is assumed to be constant throughout navigation. Because the phone never rotates around the x and y axes, the z-axis expressed in body coordinates should be constant. Using this fact, the world z-axis can be measured in body coordinates before navigation starts, and the same vector can be used in conjunction with the latest magnetometer measurement to determine the absolute orientation of the phone, as described in Section 7.1.1.

This method was evaluated, and while it is a good idea in theory, it does not work in practice due to how humans behave when walking. Typically when a person is walking, the body is tilted slightly forward, and at each step taken the body changes posture in a rocking motion. What that means for the validity of this method is that the magnetometer vector represents the value for the current posture, while the z-axis used represents the posture when navigation started. It is unlikely that these two postures match each other and the result is an orientation estimate worse than simply relying on the integration of the gyroscope.

Using the current phone orientation

Since the navigation system already has a reasonably accurate estimate of the orientation, the estimate of the z-axis can be used. Combining the z-axis estimate with the magnetometer measurement yields a new orientation according to Section 7.1.1.

The problem with this approach is that it can never correct drift around the x and y axes since the z-axis is always trusted. For the purpose of 2 dimensional navigation, the orientation of the z-axis is the most important since it controls the behavior when the user turns. Even if the z-axis can not be corrected, this method could potentially aid the orientation of the x and y-axes if the magnetometer measurements were correct. However, the false positives when detecting good and bad magnetometer measurements is typically greater than the error in the gyroscope estimation of the x and y axes. The result is that the method ends up doing more harm than good.

7.2.3 Magnetometer conclusion

Various methods of correcting the orientation have been discussed, assuming the system is provided with magnetometer measurements that are not affected by disturbances. It should be clear that without being able to correct the

estimate of the world z-axis, none of the methods is guaranteed to improve the phone orientation estimate.

On top of that, detecting whether the magnetometer is affected by disturbances is a difficult task. Section 7.2.1 suggests a method which compares the measured magnetic field to the expected magnetic field. However, our trials have shown that this method is not a robust way to detect magnetic disturbances.

One reason for this is the fact that magnetic disturbances gradually becomes stronger as the distance to the source of the disturbance becomes smaller. As such, when the user approaches a disturbance, the error in the magnetometer vector also grows gradually. The first few of the disturbed measurements will only be slightly different to $\vec{m}^b_{\text{true}}$, and will therefor be incorrectly classified as valid measurements. The invalid measurements will be trusted, and therefore used to update the phone orientation estimate. However, this causes the phone orientation to become even worse, and as a result, $\vec{m}^b_{\text{true}}$ will not be accurately calculated (according to Equation 7.2.1). When $\vec{m}^b_{\text{true}}$ no longer represents an accurate estimate of the true magnetometer vector (the one pointing to the magnetic north), the method loses its credibility.

A more robust method for detecting magnetic disturbances is suggested in [15], where all magnetometer measurements in a time span (a sliding window) are compared to the true magnetic vector. Only if all the magnetometer values in the window were similar to the true vector would they be trusted and used to correct the phone orientation.

Since we have been unable to reliably incorporate the magnetometer in the orientation estimator, this thesis limits itself to using the gyroscope for updating the phone orientation.

Chapter 8

Merged navigation system

In Chapter 6, a method for determining distance moved was presented. In Chapter 7, a method for tracking the heading of a user was presented. This chapter is dedicated to merging the two system into one system that will provide an estimate of the users position.

8.1 System

There are two components in the navigation system, the orientation tracker and the step counter. The orientation tracker provides the system with the heading $\alpha(t)$, defined as an angular offset from north in the local frame at time t. The heading is defined as positive in the counter clockwise direction.

The second component is a step detector that analyses the accelerometer values and detects steps. The step counter provides the system with a step event for each step i, containing the time stamp $t(i)$ of when the step occurred. The position of the user $p(i) = \begin{pmatrix} p_x(i) & p_y(i) \end{pmatrix}^T$ represents the x and y coordinates of the user in the local frame after step i occurred.

A step event is generated when the step completes and the foot hits the ground and it is assumed that the step was taken in the direction of the latest heading update α_i. Updating the position is done by

$$p(i) = \begin{pmatrix} p_x(i) \\ p_y(i) \end{pmatrix} = \begin{pmatrix} p_x(i-1) - \sin(\alpha_i) \cdot l_s \\ p_y(i-1) + \cos(\alpha_i) \cdot l_s \end{pmatrix},$$

where l_s is the estimated step length. In this thesis, the step length is assumed to be constant and equal to 0.9 m.

8.2 Constraints

There are constraints that must be fulfilled in order for the system to work;

- **Fixed orientation relation:** During navigation, the phone must stay in the same orientation relative to the body of the user from start to end. This is a must because phone movement is assumed to represent body movement. If the orientation relation changes, it is impossible to tell what is a change in the heading of the user and what is a change in orientation relation.
- **Known start heading:** The user is required to start facing a known heading, for example north. This is because the heading estimator only provides headings relative to a start heading. Without a known starting orientation of the user the heading estimates provided by the orientation tracker are meaningless.
- **Known start location:** The user is required to start in a known location, since the system only tracks relative position changes. Without a known starting position the system will not work.
- **Calibration:** It was discovered that the constant bias of the gyroscope changed over longer periods of time than the measurements in Chapter 3 were conducted over. Therefore, since the constant bias is a large error source, it was required to re-calibrate this measurement before each navigation. This was done by simply by letting the phone idle on its back for a short time before navigation started.

8.3 Error sources

There are several sources of error in the position estimate. The system is however far more robust than an INS system and the errors are all of linear nature.

8.3.1 Distance errors

When the step detection algorithm misses a step or detects a false positive step an error in the position will be introduced. This error is constant and equal to the step length l_s. Moreover, the position estimate is modified only once by this error, and the error of future position estimates due to the missed step is the same constant.

Errors in the estimated step length l_s add a scale factor to the estimated distance from the starting position, p_s. Consider a user who has a step length of 0.8 m, but l_s is set to 0.9 m. If the user walks 100 steps away from p_s in an

arbitrary direction, the estimated position will be 90 m away from p_s instead of the true distance of 80 m. At this point there is an error of 10 m in the estimated position. If the user then turns 180 degrees and takes another 100 steps the user will be back at p_s. The estimated position of the user will also be p_s even though the estimated distance traveled is 180 m instead of the actual 160 m.

8.3.2 Orientation errors

The main source of error is the orientation estimate. A constant error in the heading would introduce an error that grows linearly with the distance from where the heading error was introduced. Once an error has been introduced to the heading it will affect all future updates of the position, since the orientation updates are relative to the previous estimate. The result of this is that if a heading error is introduced early while navigating, moving away from the error point causes the error to grow linearly, but moving back to the error point causes it to decrease linearly.

The errors in orientation are of such a nature that the drift around the world x- and y-axis will harm the heading estimate the most. This is because during navigation, the majority of the orientation changes are around the world z-axis. If the estimated orientation of the phone is incorrect and direction of the world z-axis is incorrect the changes to the heading will also be incorrect.

8.4 Test

In order to evaluate the navigation system a number of test walks were performed. The test walk was designed in such a manner that it would match a typical use case, where a user starts at a known starting location and navigates for approximately two minutes. A path containing both left and right turns were used as the test path, shown in Figure 8.1.

Along the path, 14 control points were set up, with known coordinates and known headings of the user at that point. When the test user reached a control point, a button on the phone was pressed in order to log the current position estimate, heading estimate and time. The logged data was then matched to the corresponding control points to determine the error in position and heading.

A typical result is shown in Figure 8.2.

In total, 10 walks were performed for a total of 140 data points. The distance between the real position and the estimated position as a function of time is presented in Figure 8.3(a). From the data, it is clear that the position error is not only correlated with the time passed, since the error starts decreasing after approximately 80 seconds. However, it is clear the variance of the position error increases with time.

Figure 8.1: *The path where the navigation system tests was conducted. The cyan dot represents the starting point and the yellow dot represents the end point.*

The reason for this effect can be seen in the example walk above in Figure 8.2. What happens in the tests is typically that an error in the heading is introduced sometime in the first minute of navigation, at which point the test path starts to move far away from the start position. The error grows as the distance to the start point increases, and then starts decreasing again as the user walks back to the start position. Figure 8.3(b) shows the heading error as a function of time. The figure shows a clear trend that the heading error increases as time passes, but regardless of this, early heading errors have a larger effect on the position error when moving away from the start position. This source of error was discussed above in Section 8.3.2, and was an expected result.

Figure 8.4 shows the result of plotting the position error as a function of estimated distance to the start position.

While the magnitude of the error clearly grows as the estimated distance to the start point grows, it does not tell the whole story. By looking at the plot it is clear that there are large variance differences between the errors at short distance from start point intervals. For example, the data points around 20 meters from the start point and the errors at approximately 25 meters from the starting point are very different. This is due to the effect of time.

Modeling the error as a linear function of two variables $e(t, d)$, where t is the time since start of navigation and d the estimated distance to the start point gives

$$e(t, d) = c + k_1 t + k_2 d + \epsilon,$$

where ϵ is a random error.

Calculating the values of the coefficients c, k_1 and k_2 can be done by linear

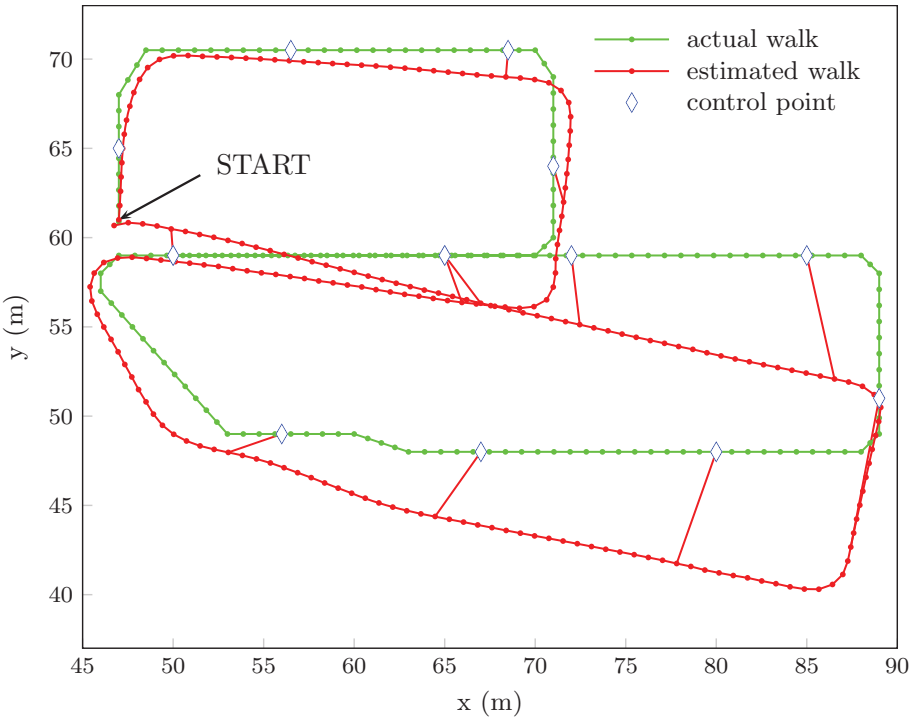


Figure 8.2: *An example result from one the walks done to measure the performance of the navigation system. The green line shows the real path walked, and the red line shows the estimation. Lines between the blue control points and the estimated walk shows the error when a control point was passed.*

regression on our sample data. Given the two matrices

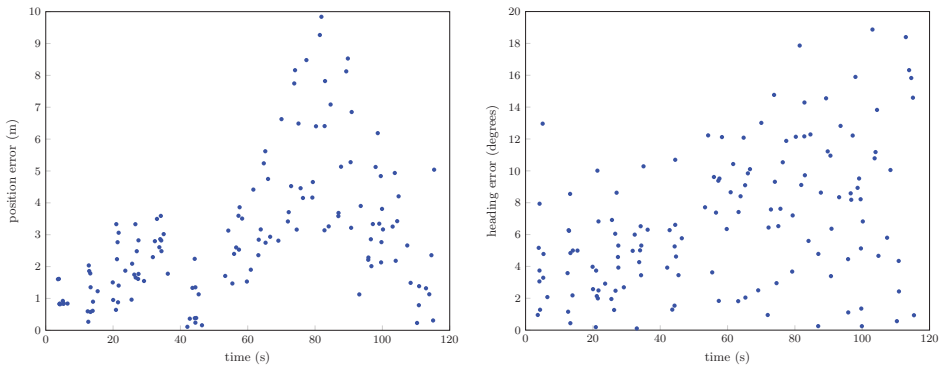
$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix},$$

where y_i is the position error of sample i, and

$$X = \begin{pmatrix} 1 & t_0 & d_1 \\ 1 & t_1 & d_2 \\ \vdots & \vdots & \vdots \\ 1 & t_N & d_N \end{pmatrix},$$

where t_i is the time since start and d_i is the estimated distance from the start location, the least squares solution is given by

$$\begin{pmatrix} c \\ k_1 \\ k_2 \end{pmatrix} = (X^T X)^{-1} X^T Y.$$



(a) The position error as a function of time.

(b) The heading error as a function of time.

Figure 8.3

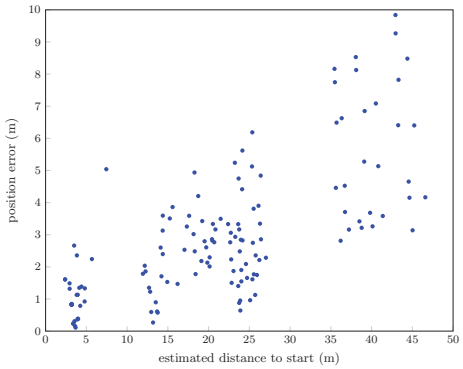


Figure 8.4: *The position error as a function of estimated distance to the starting point.*

The coefficients were calculated using the test data collected and the model was determined to be

$$e(t,d) = -0.2398 + 0.0152t + 0.1081d, \tag{8.4.1}$$

where e is the estimated error in meters, t is the time since start of navigation in seconds, and d is the distance from the start in meters.

The results of the least squares fit is shown in Figure 8.5.

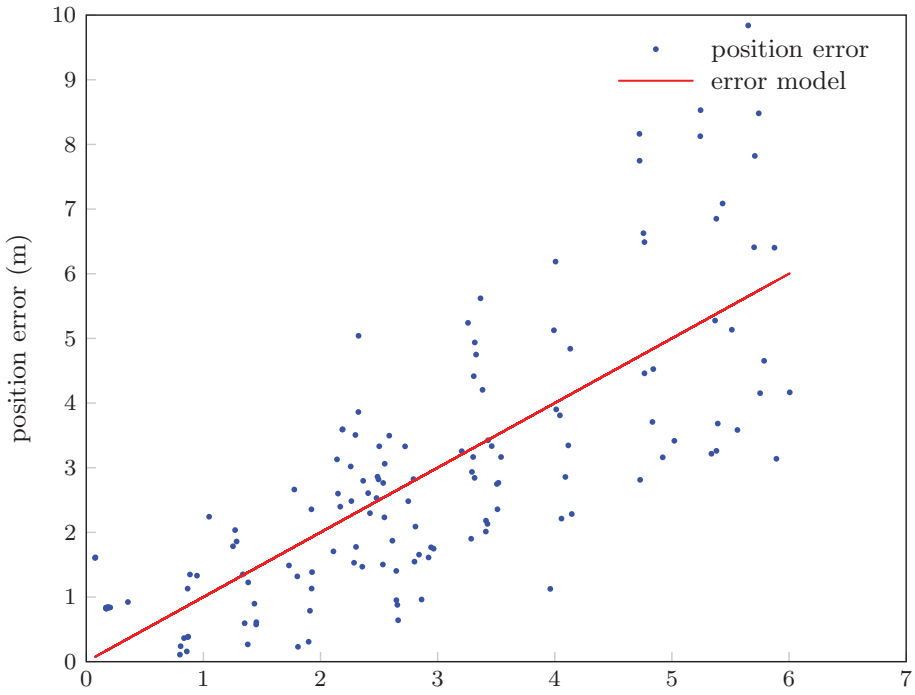


Figure 8.5: *The error model as a linear function of the time since start and the estimated distance from the starting point, determined by a least squares fit. Since the model is a linear function of two variables, it is really a plane. The figure is only an illustration of how well the model fits the measured data and the values on the x axis are not important.*

Chapter 9

Access point trilateration

This chapter describes the basics of radio propagation and suggests an algorithm that trilaterates the position of an access point based on pairs of positions and signal strengths.

9.1 Signal strength as a distance indication

The received signal strength indication (RSSI) is a measurement of the power in a received radio signal. An RSSI measurement of a radio signal is a single number, usually expressed in dBm, which is the power ratio with 1 mW as reference point, expressed in decibel. For example, a received signal strength of 1 μW gives an RSSI measurement of

$$R = 10 \log_{10} \frac{1\,\mu\text{W}}{1\,\text{mW}} = 10 \log_{10} \frac{1 \cdot 10^{-6}}{1 \cdot 10^{-3}} = -30\,\text{dBm}$$

The power of the received signal generally decreases as the distance to the transmitter increases. For an RSSI measurement to mean anything in terms of distance to the transmitter, two things are required.

- A reference point, such as what the typical RSSI measurement is at distance 1 m from the AP.
- A model of how the RSSI changes as the distance increases.

To derive a model, it is necessary to review the sources of the path loss that affects the signal.

9.1.1 Signal path loss

There are several reasons for radio path loss.

Free space path loss (FSPL) occurs when the signal propagates through space. Additionally, walls and objects can absorb, refract and reflect the signal which creates further losses. The FSPL is the only loss that is easy to calculate mathematically. Losses caused by other sources can be estimated either by analyzing the environment or by measuring the signal behavior and empirically estimate the average loss.

Free space path loss

The free space path loss occurs when the signal spreads out in space. One way to derive the FSPL is to think of the signal as a sphere with its midpoint at the transmitter and a radius that changes with the distance of the receiver. The total power of the signal is the same regardless of the radius of the sphere, but since the area increases when the radius increases, the power per area will be smaller.

The area of a sphere with radius r is given by

$$A = 4\pi r^2.$$

The losses also depend on the aperture of the receiving antenna, which adds a frequency dependency. The total free space path loss, L_{FS}, in decibel, is given by [6]

$$L_{\text{FS}} = 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) + G_{\text{TX}} + G_{\text{RX}},$$

where d is the distance between the transmitter and receiver, λ is the wavelength of the signal and G_{TX} and G_{RX} are the gains of the transmitter and receiver, respectively. Because decibel is a relative measurement, d is not the actual distance between transmitter and receiver, but rather the distance relative to 1 m (which means the actual distance will be the same number anyways).

The signal used here operates only on the 2.4 GHz band, and assuming the antenna gains are constant, the free space path loss can be written

$$L_{\text{FS}} = K_1 + 20\log_{10}(d)$$

for some constant K_1.

Other losses

In addition to the free space losses, there are other losses in an indoor environment due to the many objects that typically exist indoors.

One of these losses is a shadowing effect that occurs when an object is placed between the transmitter and the receiver. The effect is that signal becomes

attenuated by the object. It is possible, albeit difficult to calculate the attenuation, since it depends on many things such as the permittivity and conductivity of the shadowing object.

However the biggest problem with indoor propagation is the multipath situation. Reflective surfaces cause the signal to bounce and create a delayed signal with a different amplitude and phase. In a complex indoor environment, a number of these reflected signals might reach the receiver. Depending on its phase, a signal might interfere with another signal and the effect can be modeled as an additional stochastic loss.

It is common to assume the sum of the other losses also increases linearly with the logarithm of the distance. In other words, the sum of the other losses is (in dB)

$$L_{\mathrm{O}} = K_2 + n_1 log_{10}(d),$$

where K_2 and n_1 are constants describing the propagation environment.

9.1.2 Modeling

An RSSI measurement R is a measurement of the power of a received signal. One way to model the measurements is to suggest R is the power of the transmitted signal at a distance of 1 m, minus the losses described in the previous section. In other words

$$\begin{aligned} R &= P_{\mathrm{out}} - L_{\mathrm{FS}} - L_{\mathrm{O}} = \\ &= P_{\mathrm{out}} - K_1 - 20\log_{10}(d) - K_2 - n_1 \log_{10}(d). \end{aligned}$$

Grouping the constants together yields the following equation

$$R = C - 10n \log_{10}(d) \tag{9.1.1}$$

for some constants C and n.

Determining values for the constants in Equation 9.1.1 is a way to model the radio environment where the measurements are being made. The constant C effectively describes the expected RSSI measurement at a distance of 1 m from the transmitter, and n describes the average attenuation in the environment. For example, $n = 2$ models the case of only free space attenuation and little or no attenuation due to objects, while $n = 4$ models a dense environment with significantly more attenuation [6].

In order to get an estimate of the distance between receiver and transmitter, solving Equation 9.1.1 for d gives

$$d = 10^{(R-C)/-10n}. \tag{9.1.2}$$

9.2 Ground truth

It was deemed necessary to collect a large amount of RSSI measurements from known locations. Such a collection has several uses. First, the algorithm operates on two types of data in pairs: position estimates and distance estimates, both of which have a given uncertainty. How well the algorithm will perform depends on how accurate the data is. By measuring RSSI at known locations, one of the uncertainties is removed, and evaluating the algorithm becomes easier.

Second, by knowing the true locations of where measurements were made, and the true location of the access points, the distance between them is also known. The RSSI and distance can be plugged in to Equation 9.1.2 and allow estimation of the parameters C and n that characterizes the radio environment of the building.

9.2.1 Method

The measurements were done in the same building and uses the same coordinate system as the local frame defined in Chapter 5. In order to make the measurements resemble the measurements to be expected from the finished system, the positions chosen were all part of corridors in the office, where people are most likely to be walking.

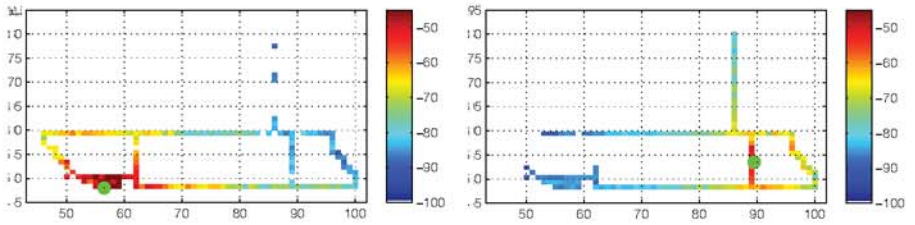
Figure 9.1 shows the coverage of the measurements done.

Figure 9.1: *The figure shows where the measurements for the ground truth were conducted. The cyan path shows the path of the measurements*

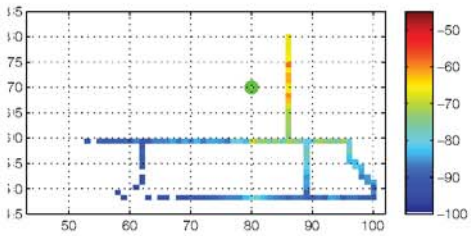
The body of the person conducting the measurements has an attenuation ef-

fect on radio signals if positioned between the AP and the measuring device [6]. In order to eliminate this effect, 8 measurements were done at each location, where the person measuring was facing in different direction. Each measurement consist of a position, a list of AP:s available and the corresponding RSSI measured for each.

In total, 1448 measurements were done. Figure 9.2 shows the median RSSI value received in each position.



(a) The access point is in the lower left corner (b) The access point is in the lower right corner



(c) The access point is in the top right

Figure 9.2: *Median RSSI values to various access points at the locations in the ground truth*

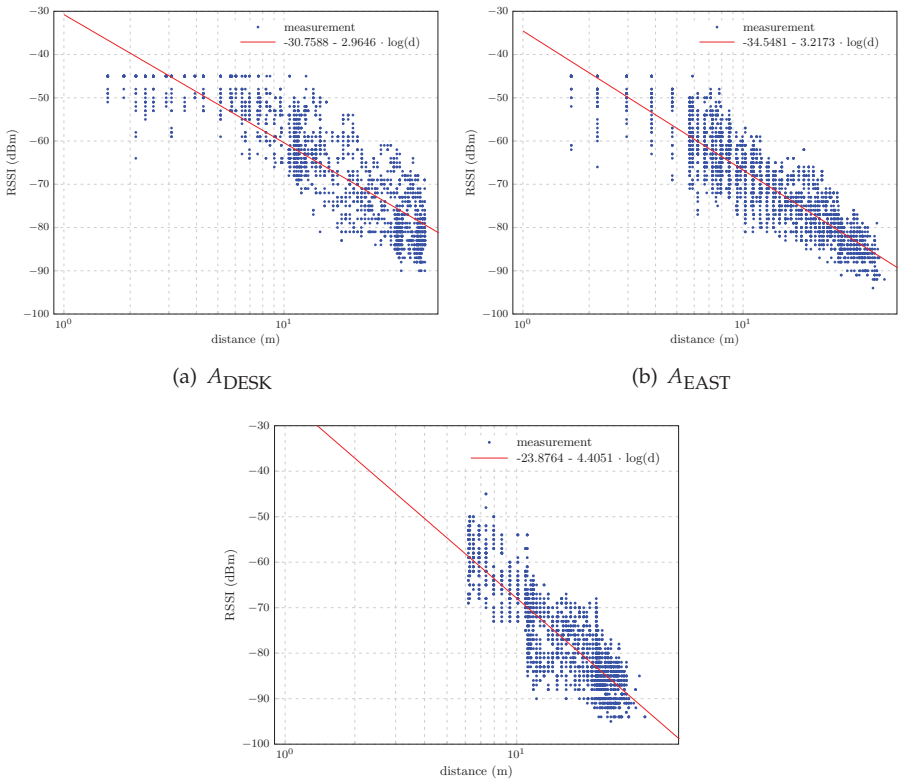
9.2.2 Analysis

Three access points are found in the vicinity of where the measurements were done, and will be referred to as A_{DESK}, A_{EAST} and A_{KITCHEN}.

The distance between the access points and the receiver is known for each given measurement. The RSSI is modeled to depend linearly on the logarithm of that distance. In order to evaluate this model, the measurements were plotted with their RSSI versus distance.

Figure 9.3 shows the plots for the three access points with a linear fit to estimate the parameters C and n. Table 9.1 summarizes the results of the linear fits.

Both of the calculated parameters C_{fit} and n_{fit} fluctuate greatly between the three access points. The reason for this is that the propagation environment to the measuring device is very different for the three access points. The access point A_{DESK} which presents the lowest attenuation constant $n_{\text{fit}} = 2.96$ is the AP that has line of sight (LOS) with the most measurements. Figure 9.4 shows



(a) A_{DESK} (b) A_{EAST}

(c) A_{KITCHEN}

Figure 9.3: *The ground truth measurement data for three different access points, with a linear fit of the propagation model*

a zoomed in version of the map in Figure 9.1 with the position of A_{DESK} and the paths that has LOS with the AP.

Figure 9.4: *The filled black circle shows the location of A_{DESK} and red paths mark the location where the measurement device had LOS with the AP*

On the other hand, A_{KITCHEN} is the AP with the least amount of LOS with the measurement device in the ground truth measurements. It is evident that there is no single n that will fit all measurements with a certain AP. For a measurement that has LOS with the AP, a low value of n is necessary, and for measurements with a lot of attenuating objects between the AP and the device, a high value of n is necessary.

	C_{fit}	n_{fit}
A_{DESK}	-30.8	2.96
A_{EAST}	-34.5	3.22
A_{KITCHEN}	-23.9	4.41

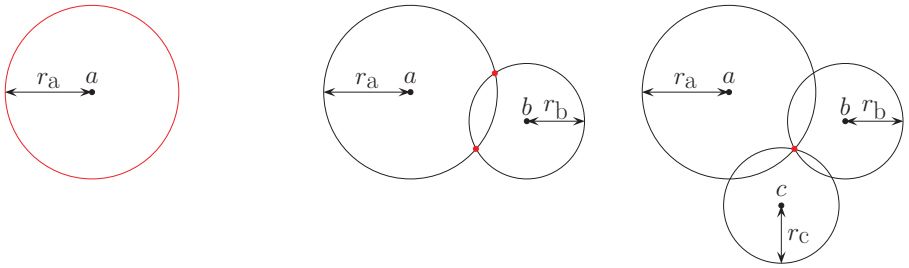
Table 9.1: *Estimated parameters for the propagation model, in a linear fit of the ground truth data for three different access points*

9.3 Trilateration

Trilateration is a method to determine the position of an object based on the distance to other objects with a known location. This section begins with describing the idea behind trilateration, followed by the development of an algorithm to estimate the position of APs.

9.3.1 Basic trilateration

Consider three linearly independent points, a, b and c, with known locations in a 2 dimensional coordinate system. Given the distance from each of the points to a fourth point, p, it is possible to calculate the position of the point p.



(a) The case of one measurement. All points on the perimeter of the circle are equally likely positions of the trilaterated object

(b) Two compatible measurements form two circles that intersect in one or two points. In the case of two points, one of them is the position of the trilaterated object

(c) Three compatible measurements yield exactly one possible point

Figure 9.5: *Example of trilateration*

By first considering one of the points, a with distance r_a to p, it can be concluded that p must be somewhere on the perimeter of the circle with midpoint a and radius r_a, see Figure 9.5(a). Given the second point b, with distance r_b to p the same reasoning can be used; p must be somewhere on the circle centered at b with radius r_b. If the measurements are compatible with each other, meaning their distance measurements do not suggest different locations of p, the two circles will intersect each other in one or two places, see Figure 9.5(b).

Because the location of p can only be on the perimeter of both circles, only the (maximum) two intersections are possible position of p. Finally, adding the last point c with distance r_c to p, there is only one point where all three circles intersect each other, p (Figure 9.5(c)).

Mathematically the equivalent situation is expressed by the system of quadratic equations

$$r_a^2 = (a_x - p_x)^2 + (a_y - p_y)^2$$
$$r_b^2 = (b_x - p_x)^2 + (b_y - p_y)^2$$
$$r_c^2 = (c_x - p_x)^2 + (c_y - p_y)^2,$$

which is solvable if the measurements are compatible with each other.

The situation is different when given distance estimated by signal strength, since they are not exact. With inexact distances, the measurements are almost guaranteed to be incompatible, and finding the location of p can not be done algebraically.

9.3.2 WiFi trilateration

Section 9.3.1 shows that it is possible to determine the position of an AP with only three distance measurements. However as shown in section 3.2.4 one RSSI value received by the phone does not correspond to a fixed distance, but rather a range of possible distances to the AP.

Model

Instead of modeling the possible locations of an AP as a circle with radius d around the point of the measurement, they are modeled as a wider band of possible locations, following a Gaussian distributions with the expected mean value of d. Figure 9.6 illustrates the idea.

There are two main reasons for modeling the possible locations L between the phone and the AP as a Gaussian distribution. First, the measured RSSI values from an AP were found to follow a Gaussian distribution in Section 3.2.4, and since the distance d is derived from the RSSI, a Gaussian distribution for the distance is sensible. Second, the propagation attenuation n is not actually constant, as in the model, but varies in different parts of a building. For example, the number of walls and other objects between the transmitter and receiver affects the correct value of n. As such, there is a need to account for n not being the correct value for a given measurement, and still allow locations at distances further or closer away than d from the measurements point to be possible (although less likely).

There are two parameters of the Gaussian distribution; the mean and the

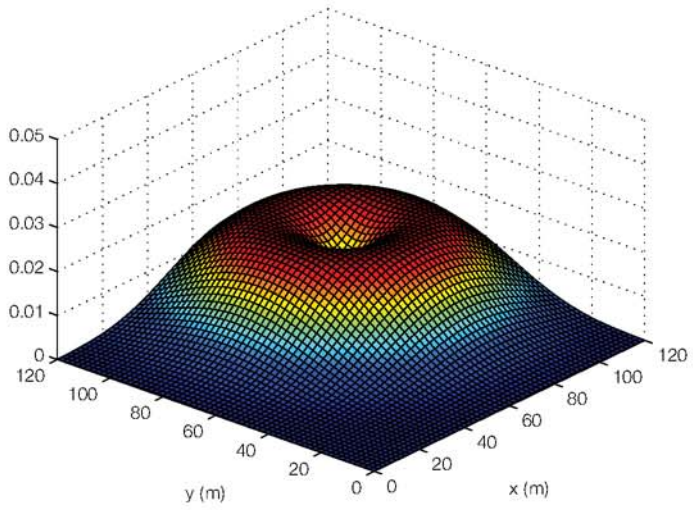


Figure 9.6: *The probability that an AP is positioned at a certain distance from where a measurement was taken follows a Gaussian distribution. The figure shows the probabilities for all locations for a measurement taken at the coordinate (80, 80)*

standard deviation. The mean must be equal to the estimated distance d, since it is the most likely spot given our model.

The standard deviation is a measurement of how aggressive the algorithm is. For example, a very high standard deviation will consider all possible location equally likely. A very low standard deviation takes the algorithm back to the exact algorithm presented in Section 9.3.1, which requires completely compatible measurements.

Another parameter affecting the suitable standard deviation is the strength of the WiFi signal. The impact of an error in measured RSSI has a much greater impact for weaker signals then it does for stronger signals. Consider the case of measuring an RSSI of -70 dBm and measuring an RSSI of -71 dBm. Using Equation 9.1.2, the difference in expected distance between the two RSSIs, Δd_1 is 0.7739 m. Now consider the case of measuring an RSSI of -90 dBm and measuring an RSSI of -91 dBm. The difference in expected distance between the two, Δd_2 is 2.6867 m. As such, the same relative difference in RSSI does not correspond to the same Δd.

To account for this effect, the standard deviation σ must increase when the estimated distance d increases. Figure 9.7 illustrates this concept for three different RSSI measurements.

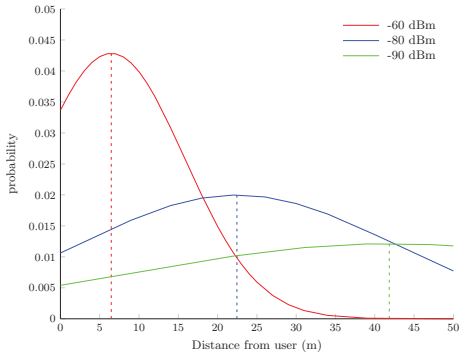


Figure 9.7: *The standard deviation of the Gaussian distribution increases as the estimated distance to the AP increases. The figure shows the Gaussian distribution for three different RSSI measurements.*

Algorithm

A matrix G, where the elements match the discrete points in the local frame is defined and maintained. Each value, $g_{i,j} \in G$ represents the probability that the AP is located at the point (i, j). At the beginning the distribution is uniform and the AP is equally likely to be located at any point (i, j) in G. Measurement number k, denoted $m(k)$, consists of three variables; the x-coordinate $m(k)_{\mathrm{x}}$, the y-coordinate $m(k)_{\mathrm{y}}$ and the measured RSSI $m(k)_{\mathrm{R}}$. For each measurement the estimated distance, $d(k)$, from the AP to the device is given by

$$d(k) = 10^{(C - m_{\mathrm{R}}(k))/10n}.$$

Instead of letting each measurement, $m(k)$, conclude that the only possible location for a given AP is on the perimeter of the circle with radius $d(k)$, each measurement contributes with a band of probable locations, as discussed above. Let $C(k)$ be the circle centered at $(m(k)_{\mathrm{x}}, m(k)_{\mathrm{y}}$ with radius $d(k)$. The further a point in the grid is from the perimeter of $C(k)$, the less likely it is to be the position of the AP. The distance $\Delta d_{x,y}(k)$ between a point with coordinates (x, y) to the perimeter of $C(k)$ is given by

$$\Delta d_{x,y}(k) = \sqrt{(m(k)_{\mathrm{x}} - x)^2 + (m(k)_{\mathrm{y}} - y)^2} - d(k)$$

The probability that the AP is located at (x, y) is then given by the Gaussian probability function. For each measurement, $m(k)$, the probability, $p_{i,j}(k)$, that the AP is at a certain point (i, j) in the matrix is given by

$$p_{i,j}(k) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\frac{-\Delta d_{i,j}(k)^2}{2\sigma(d)^2}},$$

where $\sigma(d)$ is the standard deviation for the Gaussian distribution.

By adding the probabilities from each measurement a final probability for each position can be calculated

$$\forall g_{i,j} \in G : g_{i,j} = \sum_{k=1}^{N} p_{i,j}(k).$$

The coordinate with the highest probability is chosen as the estimated position of the AP.

Chapter 10

Complete system

This chapter provides an overview of how the different parts of the system work together to localize WiFi APs. The two parts are the navigation system discussed in Chapter 8, which collects data, and the trilateration algorithm discussed in Chapter 9, which operates on the data.

10.1 System description

The two parts of the system are separated, with one part running in real time on a number of phones and one part running on a server.

10.1.1 Phone

The navigation system is implemented on Android and runs on the phone itself. The position of the phone is tracked using the techniques describes in Chapters 6, 7 and 8.

Throughout navigation, the phone continuously performs scanning of the WiFi environment to determine the RSSI values for all nearby APs. On the device used in this thesis, one complete scan takes approximately 1.5 s, and results in a list of found APs and their corresponding RSSI. For each RSSI attained, a measurement vector of the form

$$m_k = \begin{pmatrix} I_k & R_k & x_k & y_k & e_k \end{pmatrix}^T$$

is created. I_k is the identifier of the AP, R_k is the measured RSSI value, x_k and y_k are the position coordinates and e_k is the estimated error of the position. Determining the coordinates of the measurement is done simply by choosing the latest coordinates determined by the navigation system at the time the scan completed.

All measurements are communicated to the server where the trilaterion algorithm is implemented.

10.1.2 Server

The trilateration algorithm runs on a server and is implemented in MATLAB[1]. While a single phone could potentially run the trilateration algorithm, the strength of the system is to have a large number of users and phones contribute with independent measurements. As such, the server contains all the measurements collected from the contributing phones, as illustrated in Figure 10.1.

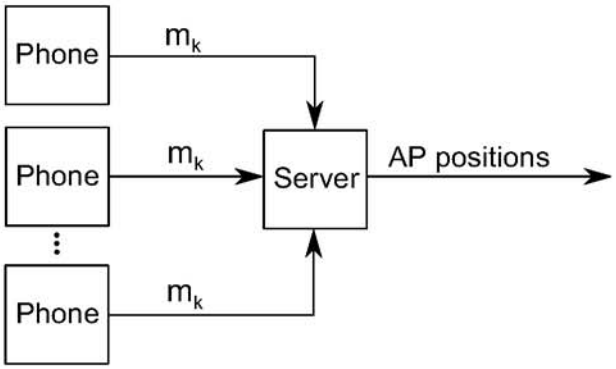


Figure 10.1: *The overall system architecture.*

The trilateration algorithm could be scheduled to run on a regular basis based on some criteria that warrants an update of AP positions. For example, the algorithm could run once per day or after a given number of measurements have been collected.

10.1.3 Algorithm parameters

The trilateration algorithm relies on three parameters,

- C: A phone and AP specific constant that describes the highest attainable RSSI at the distance 1 m.
- n: The propagation constant that quantizes how the signal strength losses depend on the distance between a phone and an AP.
- σ: A measurement of the uncertainty of a given measurement and overall aggressiveness of the algorithm.

[1] http://www.mathworks.com/products/matlab/

While C is actually unique for all phone and AP combinations, it was necessary to choose a constant C for each phone participating. Different APs may require different values of C, but it is difficult to determine the value dynamically. In Section 9.2.2, measurements were conducted in the building to determine the constant. Table 9.1 presented the results, with C ranging between -34.5 and -23.9 for the different APs. For the result measurements, C will be fixed at -30.

The n parameter is highly variable and depends on where a measurement was taken, since the propagation environment can change very suddenly between two locations that are close to each other. Choosing an n for the algorithm therefore has to be an estimation of the average propagation loss that can be expected between a phone and an AP in the building where measurements are done. Since it depends not only on the building but also exactly where the users are walking, it is difficult to determine a good value. The analysis in Section 9.2.2 showed that n varies between values near 3 for APs that often had line of sight with the measurement phone, up to 4.4 for APs that seldom had line of sight with the measurement phone. A trade off between these two extremes is to assume that the measurements phone will sometimes have LOS and sometimes not have LOS. Therefore, n will be fixed at 3.75 for the result measurements.

The uncertainty measurement σ depends on the following parameters:

- The variance of the RSSI measurements at a static location. Assumed to be constant.
- The distance between the measuring phone and the AP, d.
- The uncertainty of the position estimate, e.

Choosing a function for σ does not affect the estimated AP position as strongly as the propagation constant n, but is more of a security device that determines how trustworthy a single measurement is. In a public setting where a large number of devices are contributing to the system, a function that gives large σ values might be preferable to eliminate the effect of bad measurements. For the result measurements in this thesis, the σ values were calculated by the function

$$\sigma(d, e) = 5 + \frac{d}{3} + e,$$

where e is given by Equation 8.4.1. The constants 5, $1/3$ and 1 were chosen somewhat arbitrarily and may require more accurate tweaking in a production quality system.

10.2 Test

In order to evaluate the performance of the system it was tested how well it could position a number of APs. To make the results representative of a general case, the system was tested in a different part of the building than it was developed in.

10.2.1 Method

The goal of the testing was to simulate the typical use case of the system. This was done by mimicking the behavior of a typical employee arriving to work in the morning. Nine employees were simulated, each assigned a desk at a random location in the office landscape (within the local frame from Chapter 5).

Each employee would do the following

- Enter the office from one of two keycard-protected entrances (a known location).
- Walk to his desk (perhaps to hang his coat and turn on his computer).
- Walk to a random point of interest in the office, such as a meeting room, the coffee machine or the bathroom.
- Walk back to his desk.

This sequence of action typically lead to walks taking between 1 and 2 minutes. Each of these routes was walked twice, once by each of the authors, for a total of 18 walks. The map in Figure 10.2 shows all the paths walked and the AP positions.

In the area where the walks were done, four access points are located (AP1-AP4). The access points are positioned in such a way that all seating areas in the office have strong signal power from at least one AP. In addition to these four APs, another two APs (AP5 and AP6) are located outside the area of where the walks were done. The distance to these two APs is large, and all measurements are taken from approximately the same direction. The expectation is that AP5 and AP6 will be much harder to position than the other four APs.

To determine the quality of AP trilateration, the distance between the estimated position and the true position is calculated. Additionally, as a measurement of how certain the algorithm is of the estimated position, a value A is calculated for each AP. A is is defined as the total area of the locations where the probability is at least 75% as large as the probability at the most probable location.

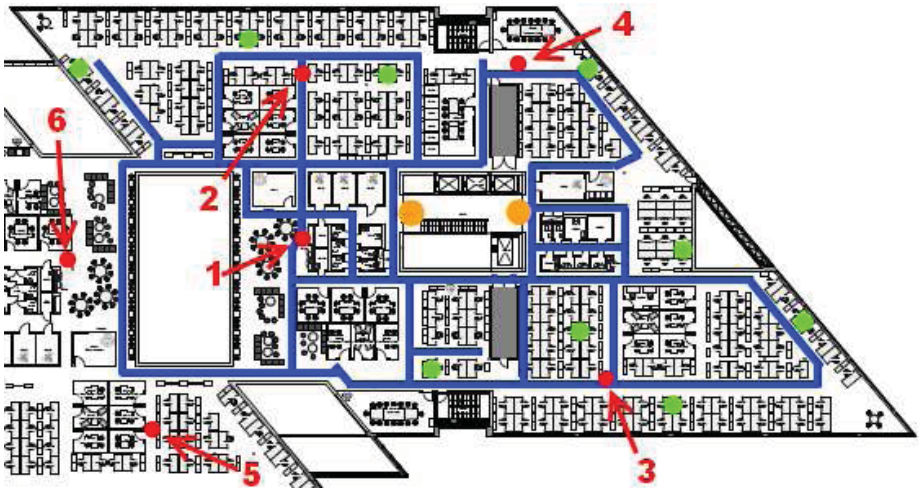


Figure 10.2: *A map of the where the result measurements were conducted. Red dots and labels mark AP positions and names, orange dots mark entrances to the office, green dots mark the random employee desks and the blue lines show all the paths walked in the tests.*

10.2.2 Results

The typical evolution of an AP position estimate is shown in Figure 10.3. With only a few measurements, the uncertainty is large and the position estimate inaccurate, but as more measurements are added the estimation becomes better. This is in line with the expected and desired behavior.

Table 10.1 and Figure 10.4 presents the results of applying the trilaterion algorithm on all the collected measurements, for each of the APs. The APs that were expected to be positioned well were all estimated at a position within 5 meters of their true position. AP5 and AP6, which were located outside the testing area could be positioned within 10 meters.

Below follows a discussion of the algorithm performance on each of the APs.

- **AP1**: The AP is positioned just west of an area in the building with several thick walls close to each other, while LOS is available from most other angles around the AP. The result is that measurements just east of the AP will have an RSSI much lower than those in other direction, causing the algorithm to push the AP away. The result is that the AP is positioned too far to the west.
- **AP2, AP3**: The results of the two are similar. Measurements have been collected on all sides of the APs and there are no unusual propagation obstruction in any direction. The result is a very accurate position estimate.
- **AP4**: Because the AP is positioned close to the outside wall of the build-

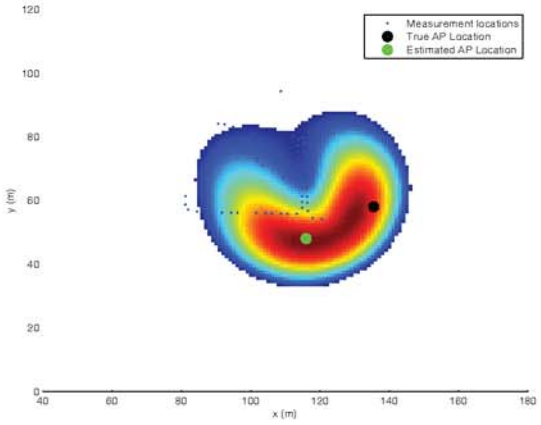
	True Position	Estimated Position	Error (m)	A
AP1	$(104.0, 72.5)$	$(99, 72)$	5	761
AP2	$(104.5, 86.0)$	$(106, 88)$	2.5	787
AP3	$(135.5, 58.0)$	$(135, 59)$	1	725
AP4	$(125.0, 91.0)$	$(125, 91)$	0	1034
AP5	$(89.5, 53.5)$	$(94, 60)$	8.5	2357
AP6	$(80.0, 70.0)$	$(91, 71)$	10	2025

Table 10.1: *A summary of the estimated position, distance error from the true position and the uncertainty measure A for all the access points in the result measurements.*

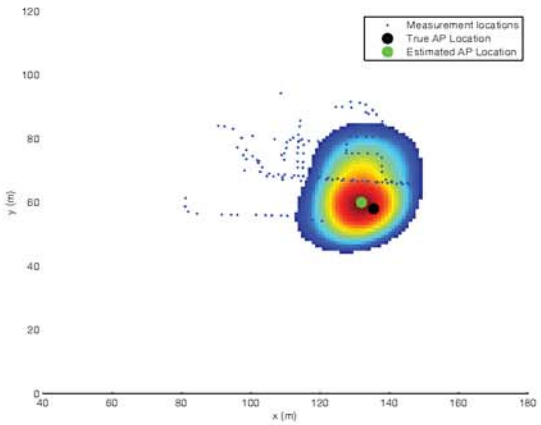
ing, measurements were only collected on one side of the AP. There are no unusual propagation obstructions in any of the direction leading to a very accurate position estimate. However, the A-measurement for AP4 is higher than for AP2 and AP3, suggesting that the estimate is not as certain.

- **AP5, AP6**: The results of the two are similar. Almost all measurements collected are done far away from the actual AP position and almost always from the same angle. Further, most measurements from the AP were done at a location with LOS between the device and the AP. Because higher RSSI values are obtained with LOS, the algorithm estimates the AP to be close to the device, resulting in the AP being pulled towards the measurement locations.

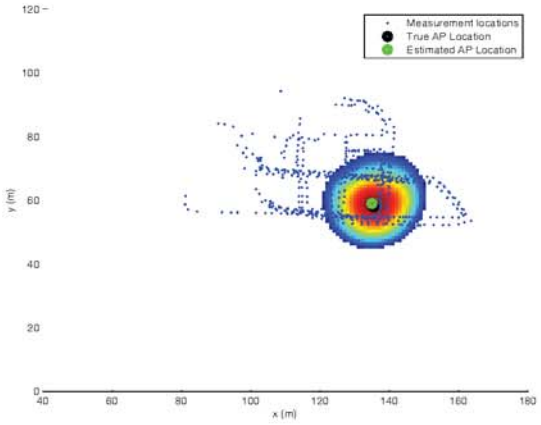
The error in estimation position is significantly higher for three of the APs in the experiment; AP1, AP5 and AP6. This is due to the characteristics of the environment in the center of the building, which is made up of an atrium providing LOS in many directions over long distances. The APs are therefore pulled towards the other side of the center (approximately at coordinate $(92, 70)$) , since these measurements will have higher RSSI than those from the other direction.



(a) After only a few measurements on one side of the AP. The position is very uncertain and the estimated position is off by 21 m.

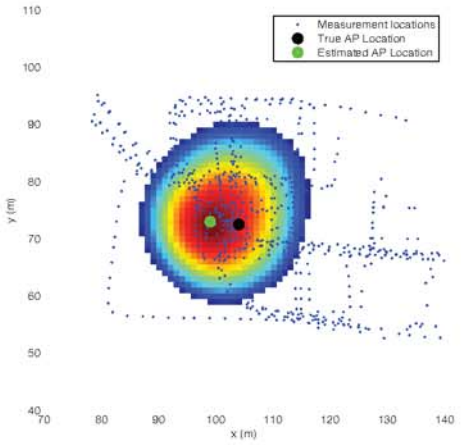


(b) As more measurements are done, the area of likely positions shrinks. Position error: 4 m.

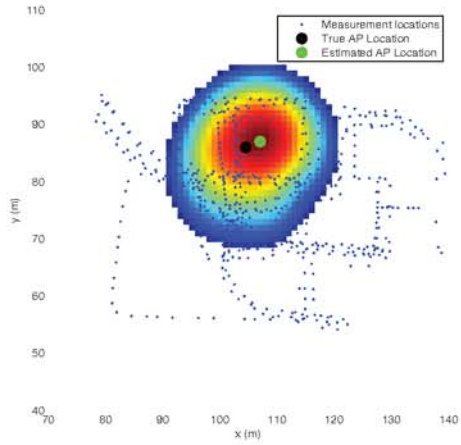


(c) After walking on all sides of the AP several times, the AP is positioned only approximately 1 m from its true position.

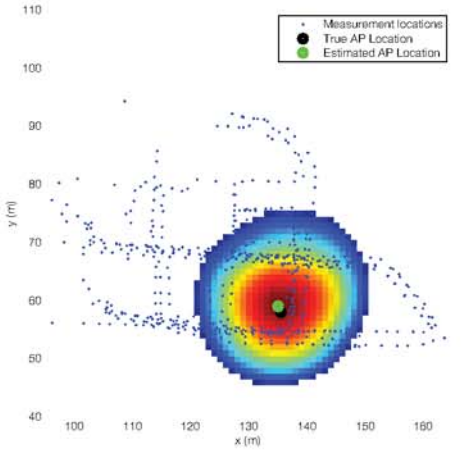
Figure 10.3: *The evolution of an AP position estimate as the number of measurements increases. The heat map shows the likelihood that the AP is positioned at each location, the darker red, the more likely.*



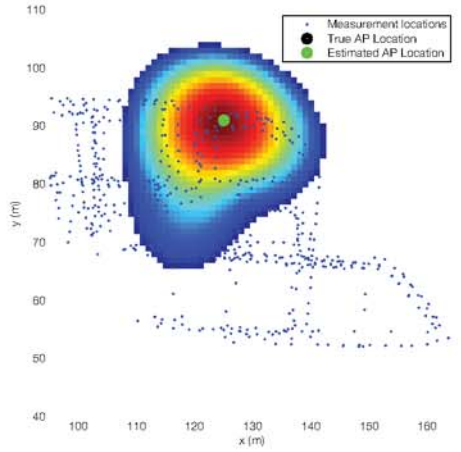
(a) AP1. Walks were done on all sides of the AP, but the east side LOS is greatly obstructed. Error: 5 m.



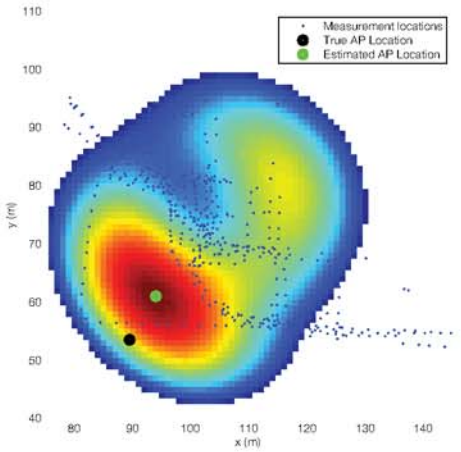
(b) AP2. Walks were done all around the AP. Error: 2.5 m.



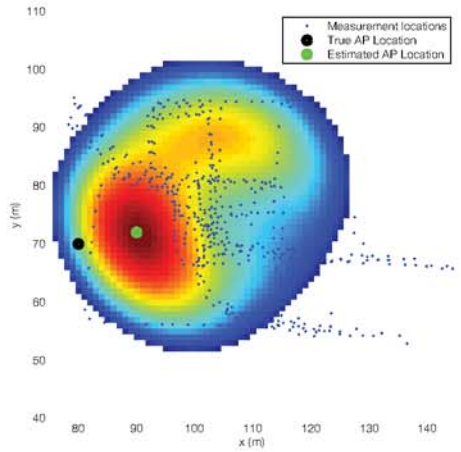
(c) AP3. Walks were done all around the AP. Error: 1 m.



(d) AP4. Walks were done around approximately a half circle around the AP. Error: 0 m.



(e) AP5. Walks were done in only a quarter of a circle around and far away from the AP. Almost always in LOS. Error: 8 m.



(f) AP6. Walks were done on only one side of the AP. Always in LOS. Error: 11 m.

Figure 10.4: *The true and estimated positions along with a heatmap showing the probability area for all APs in the area of the result measurements.*

Chapter 11

Conclusion

This chapter presents the conclusions that can be made from the material presented in this thesis. It also describes possibilities of how the system could be made better through future work.

11.1 Contributions

There are two contributions made in this thesis. The first is a navigation system implemented on the Android platform which is able to track the position of a walking person. The second contribution is an algorithm that given a set of position and signal strength measurements can estimate the positions of access points.

11.1.1 Navigation using sensors

The thesis examined two different ways to perform navigation by using the sensors available in the phone. Both rely on using a three-axis accelerometer accompanied by a three-axis gyroscope.

Inertial Navigation

The possibility of creating an INS to navigate indoors was examined and it was concluded that it is not a feasible approach. The reason is that the error in position grows proportionally to t^3, where t is the time since navigation started. Even by using much better sensors, the error growth can only be stalled for a short time before growing extraordinary large. Further, even if one of the sensors is perfect and always measures the correct values, an error from the other sensor causes the error to grow at least quadratically.

In order to be able to perform inertial navigation, navigation must be constrained in some way. One example of a constraint is to mount the sensors on the foot of the user. Using this technique, one can detect when the foot is stationary which makes it possible to calibrate the navigation system. If a map of the building of navigation is available, other constraint can be added, such that the user can not move through walls or other objects.

With no constraints on the navigation, an inertial navigation system quickly introduces errors too large for practical use.

Step detection

A navigation system that can track the position of a walking person was developed. The size of the position error when navigating grows depending on two units, the time since navigation started and the distance from the start location. In a typical two minute walk around an office area, the navigation system is able to track the position of a user within a couple of meters at all times.

The navigation system is constrained by the assumptions that at the start of navigation, location and orientation of the user is known. Additionally, the orientation of the phone with respect to the walking person can not change during navigation. These constraints are not very severe, since they can easily be overcome using e.g. a known calibration point such as a keycard entrance or GPS if navigation starts from outside.

11.1.2 Trilateration of access points

An algorithm has been developed that, given a set of RSSI measurements at different location can estimate the position av an access point. The performance of the algorithm depends on where the RSSI measurements are collected in relation to the AP. For a typical AP where measurements can be collected from all directions around the AP, the position can be estimated correctly within a few meters.

The largest challenge in applying the algorithm is choosing a propagation constant n that fits the building. Since n is constant, the signal propagation characteristics must be similar throughout the building, in particular in all directions around an AP. This assumption is true for the building where the work was conducted, with the exception of an atrium in the middle of the building which causes significantly less attenuation than the other parts of the building.

Additionally, the value of n must reflect the average attenuation at a given distance from an AP, which might be different for different buildings. As such, the system in its current form cannot instantaneously be applied on measurements from another building since a new approximation of n must

be done. However, profiling the radio environment in a building only has to be done once to acquire a value of n, which can be reused in case the position of APs changes and must be found again.

11.2 Future work

Two problems are present in this thesis, determining the position of a user and determining the position of access points. Several enhancements can be made to the system proposed in this thesis, in both areas.

11.2.1 Improve usability

With a large user base, the system could become useful in a large setting where AP positions are determined all over the world. However, in its current form, the system requires the user to manually start the navigation application at a known location and a known heading. While this works for enthusiasts, it is not likely to gain wide use. Ideally, the system would only need to be installed by the user and then operate entirely on its own. For this to work, it would be necessary to automatically overcome the constraints mentioned.

One way could be to incorporate position data into Near Field Communication (NFC) tags that are read by many phones. If the user is coming from the outside, GPS could be used to determine current position and heading.

11.2.2 Using a map

In this thesis, it is assumed that there is no map information available for the building in which navigation and access point trilateration takes place. However, using a map can assist both navigation and the trilateration algorithm.

When navigating in the current system, small errors in the heading produces large errors in the position estimate the further the user walks. By using a map, it would be possible to align the path of the walk with corridors found in the map, thus removing the heading error early. In the same manner, step lengths could automatically be calibrated by matching lengths of paths with numbers of steps needed to complete it.

Map information could also be used by the trilateration algorithm to aid the distance estimate to an AP given an RSSI measurement. For example, the number of walls between the phone and the AP would affect the distance estimate.

11.2.3 Navigation improvements

The navigation algorithm developed uses only the accelerometer and gyroscope to track the position changes. There are several more sources of information that could be used to help the navigation system. Examples include

- **Known WiFi APs**: Access points that are already accurately positioned can be used to aid the position estimate.
- **GPS**: While generally not available indoors, sometimes a strong enough signal for an accurate position can be acquired.
- **Beacons**: Other beacon systems than WiFi, such as bluetooth could potentially be used.
- **Calibration points**: In some locations, calibration points such as NFC tags could be available together with access to a navigation application.

11.2.4 Trilateration algorithm improvements

The trilateration algorithm presented is simple, and could be enhanced or changed in several ways;

- In the present algorithm, only signal strength measurements of an AP can affect its position. However, not receiving a signal at all gives an indication the AP is probably far away. This information is currently not used.
- The estimated position of an AP is chosen simply as the coordinate with the highest probability. Better results could potentially be attained by not only looking at a single point but also at nearby points.
- The uncertainty measurement A of an estimated AP position is not a very accurate approximation of how well the AP is actually positioned. An algorithm that gives a better measurement of how well positioned an AP is must be developed before the AP positions can be used for positioning.

References

[1] IEEE standard specification format guide and test procedure for single-axis laser gyros. *IEEE Std 647-2006 (Revision of IEEE Std 647-1995)*, September 2006. doi: 10.1109/IEEESTD.2006.246241.

[2] James Diebel. Representing attitude: Euler angles , unit quaternions, and rotation vectors. Technical report, Stanford University, October 2006.

[3] Fredrik Gustafsson. *Statistical Signal Processing*. Studentlitteratur, 2010.

[4] Google Inc. Android Developers - SensorManager, May 2011. URL `http://developer.android.com/reference/android/hardware/SensorManager.html`.

[5] Thomas Johansson. *Utvärdering av precisionen hos sensorer för tröghetsnavigering*. Halmstad University, School of Information Science, Computer and Electrical Engineering (IDE), 2009. URL `http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-2575`.

[6] Peter Karlsson. *Measuring and Modelling of the Indoor Radio Channel at 1700 MHz*. PhD thesis, Department of Applied Electronics, Lund University, August 1992.

[7] J.W. Kim, H.J. Jang, D.H. Hwang, and C. Park. A step, stride and heading determination for the pedestrian navigation system. *Journal of Global Positioning Systems*, 3(1-2):273–279, 2004.

[8] J.B. Kuipers. *Quaternions and rotation sequences*. Princeton Univ. Press, 1999. ISBN 0691058725.

[9] Q. Ladetto. On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. In *ION GPS*, volume 2000, 2000.

[10] Ordnance Survey Great Britain's national mapping agency. A guide to coordinate systems in Great Britain, December 2009. URL `http://www.ordnancesurvey.co.uk/`.

[11] Jouni Rantakokko, Peter Strömback, S-L Wirkander, M Alexandersson, K Fors, Isaac Skog, and Peter Händel. Foot-mounted inertial navigation and cooperative sensor fusion for indoor positioning. In *ION 2010 International Technical Meeting (ITM'2010)*, San Diego, CA, January 2010.

[12] Isaac Skog, John-Olof Nilsson, and Peter Händel. Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. In *2010 INTERNATIONAL CONFERENCE ON INDOOR POSITIONING AND INDOOR NAVIGATION (IPIN)*, ZURICH, SWITZERLAND, September 2010.

[13] Dr. Walter Stockwell. Bias stability measurement: Allan variance. Technical report, Crossbow Technology, Inc., 2009. URL `http://www.xbow.com`.

[14] D.H. Titterton, J.L. Weston, and Institution of Electrical Engineers. *Strapdown inertial navigation technology*. ISBN 9780863413582.

[15] David Törnqvist. *Statistical Fault Detection with Applications to IMU Disturbances*. Licentiate thesis no. 1258, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, June 2006.

[16] David Törnqvist. *Estimation and Detection with Applications to Navigation*. Linköping studies in science and technology. dissertations. no. 1216, Linköping University, November 2008.

[17] Oliver Woodman and Robert Harle. Pedestrian Localisation for Indoor Environments. In *Proceedings of the Tenth International Conference on Ubiquitous Computing (UbiComp 08), Seoul, Korea*, September 2008. URL `http://www.cl.cam.ac.uk/research/dtg/www/files/publications/public/ojw28/Main-PersonalRedist.pdf`. Conference Paper.

[18] Oliver J. Woodman. An introduction to inertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, August 2007. URL `http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf`.

[19] Oliver J. Woodman. *Pedestrian localisation for indoor environments*. PhD thesis, University of Cambridge, Computer Laboratory, September 2010.