Universiteit Utrecht

Analysis of Inertia Based Positioning Systems

Master Thesis

Game and Media Technology

ICA-3864669

Rik Bosma

Supervisors:

dr. ir. A.F. van der Stappen

dr. R.J. Geraerts

August, 2014

Version 1.0

Abstract

We want to create an immersive augmented reality framework. An essential part in this framework is aligning a virtual environment with the real world. To align the virtual environment we need to know our position and orientation in the real world as accurate as possible. In order to obtain our position and orientation we need a positioning system. We also want to have a smooth user experience, which means that the positioning system has to be fast. Therefore we analyze an inertia based positioning system to see whether it meets our requirements.

Inertia based positioning systems are subject to a cumulative integration error called drift. Our first objectives is to analyze the drift and to find what causes this error. Then we try to keep the drift within a certain error margin for a longer time by improving the measurement signals; which appear to be the biggest effector on the drift. We also use computer vision to verify the measurements. Our final objective is to analyze whether we can use a map-based positioning system as reference, in order to meet the requirements to the system.

The results we present show that we can improve the measurement signals using a Kalman filter with models based on the Wiener process acceleration model. We have also analyzed the drift over time and we concluded that using a map-based positioning system as reference would suffice to keep the drift within considerable error margins. However, we strived to reduce the drift using computer vision as well. Unfortunately, the hardware used in our experiments was not capable to reduce the drift further.

Contents

1 Introduction 6

1.1 History 6

1.2 Problem statement 7

1.3 Objectives 8

1.4 Related work 8

1.4.1 Environmental beacon systems 8

1.4.2 Map based positioning systems 9

1.4.3 Inertia based positioning systems 9

1.5 Contributions 9

1.6 Layout 10

2 Preliminaries 11

2.1 Positioning systems 11

2.1.1 Environmental beacon systems 11

2.1.2 Map-based systems 13

2.1.3 Global beacon systems 13

2.1.4 Inertia-based systems 14

2.2 Rift 14

3 Inertia based positioning system 16

3.1 Preliminaries 16

3.1.1 Dead reckoning 16

3.1.2 Inertial measurement unit 17

3.1.3 Signal processing 18

3.2 Research 20

3.2.1 System 20

3.2.2 Inertial measurement unit positioning 21

3.2.3 Errors in the system 22

3.2.4 Signal analysis 25

3.2.5 Filters 29

3.3 Implementation 38

3.3.1 Programming environment 38

3.3.2 System....38
3.3.3 Filters....40
3.3.4 Integration methods....41
3.4 Experiments....42
3.4.1 Setup....42
3.4.2 Cases....43
3.4.3 Kalman filter parameters estimation....44
3.4.4 Drift analysis....44
3.5 Results....45
3.5.1 Kalman filter parameter estimation....45
3.5.2 Drift analysis....48
3.6 Conclusions....53
3.6.1 Kalman filter parameter estimation....53
3.6.2 Drift analysis....54
3.6.3 Objectives....54
4 Monocular vision....55
4.1 Preliminaries....55
4.1.1 Computer vision....55
4.1.2 Optical flow....56
4.2 Research....56
4.2.1 No movement method....56
4.2.2 Correct gravity method....57
4.2.3 Direction estimation method....58
4.3 Implementation....58
4.3.1 Programming environment....58
4.3.2 System....58
4.3.3 Methods....60
4.4 Experiments....61
4.5 Results....61
4.6 Conclusions....63
5 Stereoscopic vision....64
5.1 Preliminaries....64

5.1.1 Stereo camera system 64
5.1.2 Features 65
5.2 Research 65
5.2.1 Point cloud retrieval 66
5.2.2 Velocity estimation 69
5.3 Implementation 69
5.3.1 System 69
5.3.2 Calibration 71
5.3.3 Method 71
5.4 Experiments 72
5.5 Results 73
5.6 Conclusions 74
6 Discussion 75
References 77
Appendix I I
Appendix II II
Appendix III III
Appendix IV IV
Appendix V V

1 Introduction

Immersive virtual reality technologies provide state of the art virtual experiences. Those experiences enable the user to walk, move and interact freely in a virtual environment. The ability to freely roam and engage in such environments allows us to create visualizations, training programs and education applications for all kind of industries. If we take this one step further then we would like assist the user in specialized tasks in the real world. In order to add virtual reality to the real world in a way we can freely roam and interact, we have to take a look into augmented reality.

Augmented reality is not a new concept, but to determine the first record of this idea we need to know the definition of augmented reality. The definition of augmented reality according to the Oxford dictionaries [1] is:

> *A technology that superimposes a computer-generated image on a user's view of the real world, thus providing a composite view.*

It is essential when a computer-generated image is overlaid on the real world that the position and orientation in the real world are known, in order to place virtual objects on a fixed position in the real world.

1.1 History

The first time augmented reality is mentioned is in the novel The Master Key written by L. Frank Baum in 1901. In the novel a fictional device called the 'character marker' is mentioned. This device is an electronic display that overlays a character in front of people their forehead. In 1957 a simulation device called the Sensorama was developed by the cinematographer Morton Heilig [2]. The Sensorama is one of the earliest known immersive multimodal devices. It allowed users to experience stereoscopic 3D images, stereo sound, vibrations and scents during a movie. The immersive experience which was created by stimulating multiple senses can be considered as breakthrough. Ivan Sutherland had two breakthroughs, the first is the head mounted display he invented in 1966 [2]. The second is in 1968 when he invented the first who created an augmented reality system which had a see-through HMD. In the year 1981 the scientist and inventor Steve Mann developed one of the first wearable computers, the EyeTap. The EyeTap is an augmented reality device that is both a display and a camera, which is placed in front of one eye of the user. This device allows the users to record and playback videos while they are still able to see the world via the see-through display in their everyday life.

Since the nineties virtual and augmented reality has become more popular. However, in that period of time the hardware was not yet capable to make it into a successful commercial device. Later in the mid-zeroes the rapid development of mobile phones provided huge improvements in all the required technologies. The displays and cameras have seen a significant increase in quality. The chips required to perform the calculations have improved to provide the necessary computation power for high-end mobile devices, like the iPhone 4 and Samsung

Galaxy series. These high-end mobile devices are used for all kinds of tasks, of which some include simple augmented reality. Thanks to the innovations in the mobile and wearable computer industry large tech and startup companies are willing to invest in those technologies. In April 2012 Google Inc. introduced a device called Google Glass which is comparable to the EyeTap, but less intrusive [3]. Oculus VR is a startup company which currently develops a virtual reality head mounted display [4]. These two projects led to a number of spin-off projects, but freely roaming in an immersive augmented reality environment remains challenging.

1.2 Problem statement

The goal is to have an accurate and fast positioning system to be able to place the virtual environment into the real world. This is required to be able to roam freely in an immersive augmented reality environment. The virtual environment and real world are combined when the virtual environment is aligned to the real world. To realize this accurate alignment between the worlds we need to know the orientation and the position of the user with respect to the real world. The virtual world is then aligned to the real world relative to the orientation and position of the user. The orientation and position with respect to the real world can be acquired by a positioning system. To do this we require next two points.

- The positioning system has to be fast enough to keep up with the refresh rate of the virtual environment, in order to avoid disturbances such as lag and jitter effects.
- The system has to be accurate to establish a solid alignment between the virtual environment and the real world.

Map based positioning systems could be a feasible category of positioning systems to align the virtual environment. In the field of robotics there has been a lot of research on this topic. A known technique for map based positioning is simultaneous localization and mapping (SLAM) [5]. This technique firstly uses real-time measured environmental data to construct a map while the user is moving around, called mapping. While this map is being constructed, the technique uses an algorithm to determine the location of the user within this map. The accuracy of this SLAM technique depends on the type and the accuracy of the used sensors. Even if the accuracy matches the requirements for an immersive augmented reality application then we still require the method to operate at least at the same speed as the refresh rate of the display. Unfortunately real-time implementations of SLAM are not fast enough to meet the speed requirement [6]. Therefore we need to look into other systems that might meet the requirements. We discuss several categories of positioning systems in Subsection 2.1. We will see that inertia based positioning systems allow us to roam freely in an arbitrary environment where beacon based systems do not. However, a major problem arises when we investigate inertia based positioning systems. Due to the algorithm used to calculate the localization an integration error occurs; we refer to this integration error as drift. This drift increases every time the position is determined, quickly resulting in a situation in which the position is too far off to meet the accuracy requirement. However, there are ways to reduce this drift. In order to reduce the drift we need to analyze the drift first.

1.3 Objectives

Our first objective is to analyze the drift. We want to know how the drift evolves over time. To do this we need to analyze the whole inertia based positioning system. We need to know what causes errors in the system that lead to the drift. For this analysis we use an inertial measurement unit (IMU) as sensor in this system. An IMU is a combination of an accelerometer, gyroscope and magnetometer. We want to find out what causes the drift, this can either be due to the measurement error or due to the applied integration algorithms. When the main causes of the drift are known, we would like to extend the period in which the drift is within a certain error margin. We found that the measurement noise is responsible for a large part in the drift, in order to reduce the drift we try to minimize this noise using a filter. We also use computer vision to reduce the drift. The third objective is to analyze whether it is feasible to use the system in combination with another system. This means that when the drift stays within the error margin of the other system between the samples we consider it feasible.

1.4 Related work

We analyze an inertia based positioning system in which we consider the system as an indoor positioning system. Multiple studies have been conducted in pursuit of accurate indoor positioning systems. In Subsection 2.1 we discuss the categories of positioning systems we distinguish. In this subsection we discuss several applied systems within those categories. Environmental beacon system related studies are considered first. Then we discuss several studies which involve maps and finally we consider inertia based positioning systems. We do not consider implementations that use the global positioning system (GPS), because this technique fails indoors.

1.4.1 Environmental beacon systems

Environmental beacon systems requires external hardware to be equipped to the operation environment. Several sensors can be used for environmental beacon systems. For example RFID beacons are placed in the environment; the location of each beacon is known and based on the position of those beacons the positioning is performed [7] [8]. Another environmental beacon system involves ultra-wideband (UWB) radios which are used for localization. Several UWB beacons are placed into an environment then the position can be calculated based on the time of arrival of the detected beacons [9]. Both implementations only work within a very limited operation area.

For larger operation areas Wi-Fi transmitters can be used. A Wi-Fi-based positioning system (WPS) uses the received signal strength indication (RSSI) of Wi-Fi hotspots or wireless access points [10]. Using the RSSI of a couple of signals an environmental profile is created, in order to determine the accuracy. The accuracy of this technique depends on the density of wireless access points and hotspots in an area. The method as presented in [10] reports a resolution of 1.5 meters with an accuracy of 83%.

1.4.2 Map based positioning systems

When map based positioning systems are used there are two types of scenarios. In the first scenario the map is known, however, unlike the environmental beacons that map does not contain predefined beacon locations nor is any kind of transmission used. In the other scenario the map is not known. When the map is known we need to locate ourselves within this map. The localization is performed by using a filter; a particle filter like Monte Carlo localization [11] can be used. There are implementations which use another filter, for example an extended Kalman filter is used as well [12]. Localization is past of SLAM, therefore we would use the SLAM algorithm in both scenarios. If the map is unknown then the map is built based on the sensor input which can range from laser range data to a set of features retrieved by a camera. Every time step the sensor inputs are combined to obtain a map. This can be performed in multiple ways, the standard approach is to run a filter for each particle, where a particle is an input point, as described in [5]. Another way to match the inputs of different time steps is using the iterative closest points (ICP) algorithm to compare the data sets, as done in [13].

1.4.3 Inertia based positioning systems

Inertial navigation has been developed to guide and navigate rockets. For example the on-board guidance, navigation and control system of Apollo used inertial navigation [14]. Later the system was used for guidance in all kinds of aerial applications, such as aircraft and missiles. When the IMU became more widely available for less high-tech applications researchers studied the uses of the device for pedestrian navigation purposes. This is useful in any situation where the GPS cannot be used. One of the most promising applications is to strap an IMU to a foot of the user as done in [15]. This method allows several assumptions which make it able to detect steps. The so called step detection allows for much more accurate positioning. There are also studies which use this principle as a base solution which is then improved using methods such as ramp detection [16]. Analyzing inertia based positioning systems for head mounted displays such as in [17], involves IMUs that have only three degrees of freedom. Also the technology of this sensor has evolved, since 1998.

1.5 Contributions

The main contributions of this thesis are:

- The most promising filter for processing the accelerometer signal is researched in detail. The results can be used to select the best set of parameters according to the usage of the positioning system.
- The performance of an inertia based positioning system for augmented reality systems is presented. We will see how the drift evolves over time.
- Several ways of using optical flow to improve the accuracy of the positioning system are presented. Their improvements are compared to the performance of the unmodified positioning system.
- We present a way to construct a sparse three-dimensional point cloud from stereo images. The three-dimensional point cloud is used to determine the movement direction, in order to improve the accuracy of the positioning system. The improvements are

evaluated against the performance of the unmodified positioning system and the monocular vision aided system.

1.6 Layout

In Section 2 we discuss the general preliminaries, which summarizes the categories of positioning systems we distinguish. Then we analyze the inertia based positioning system and the drift in Section 3. In this section we also propose methods to reduce the drift considering the measurements and the implemented methods. We try to reduce the drift by using additional measurements which involve computer vision. In Section 4 we consider to use a method which uses monocular vision to detect motion in sequential images. Because this approach does not satisfy in our experiments we investigated how point clouds could be of assistance in reducing the drift. The point clouds are constructed using stereoscopic vision as we discuss in Section 5. Finally, we have a final discussion about this work in Section 6.

2 Preliminaries

In this section we consider several principles and ideas, which serves as foundation for the rest of the work. In Subsection 2.1 we discuss the positioning systems researched in the pre-study. We distinguish several categories of positioning systems. Finally, we discuss the Oculus Rift is in Subsection 2.2.

2.1 Positioning systems

We distinguish four categories of positioning systems, each category depending on a different principle. In Subsection 2.1.1 we consider environmental beacon systems and in Subsection 2.1.3 we discuss global beacon systems. We explain map-based systems in Subsection 2.1.2 and inertia-based systems are discussed in Subsection 2.1.3.

We discuss those system to see which system is most likely to give an accurate result at a high speed performance. This is required since the alignment of the real and virtual world needs to be updated at a rate of 60 Hz.

2.1.1 Environmental beacon systems

Positioning systems that require hardware to be equipped to the environment are categorized as environmental beacon systems. Figure 1 illustrates an environmental beacon system used to estimate the unknown position. The blue dots in the figure represent the beacons, which the first principle uses. In this case a device attached to the user either measurers or is measured by the beacons [18] [19]. The measurements performed by the beacon setup either measures distance or the angles with respect to each beacon. Triangulation or trilateration calculations are used to estimate the position.

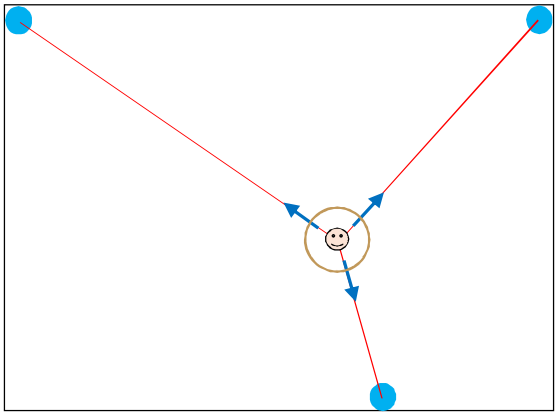


Figure 1. Environmental beacon systems. Blue dots represent beacons with distance or angular measurements.

Triangulation

Triangulation uses the measured angles to determine the position. For example in the two dimensional case as illustrated in Figure 2 [20] the position of R is estimated. The location of the beacons, denoted as B_1, B_2 and B_3, are known by the system. The angles α_1, α_2 and α_3, are measured relative to the orientation θ of R based on the locations of the beacons. Each angle is known, so the position and orientation can be calculated.

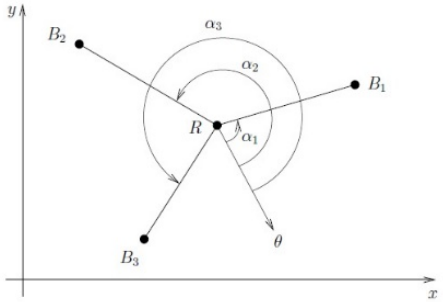


Figure 2. Triangulation setup in 2D plane. R is the position estimated. The angles (α) are used in combination with the beacon locations (B) to determine the position of R. [7]

Trilateration

Trilateration uses measured distances to determine the position. The global positioning system (GPS) uses trilateration calculation to determine ones position as well. Figure 3 [21] illustrates a two dimensional example. The distance from the unknown position, the white dot, to each beacon is known. A circle is spawned with as radius the measured distance and the center on the position of the corresponding beacon. The position is then on the intersection of all three circles.

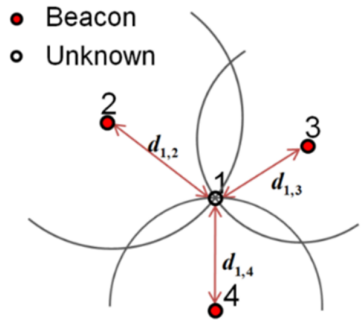


Figure 3. Trilateration setup in 2D plane. The white dot is the unknown position. Using the distance to each beacon the unknown position is determined. It is where all circles intersect. [8]

Advantages and disadvantages

An advantage of an environmental beacon system is that the position is estimated with an accuracy of several centimeters. This accuracy is the same over time, because the positioning is absolute to the environment. Environmental beacons have two disadvantages as well. The

first disadvantage is that external hardware needs to be installed in the environment. The second disadvantage is that the position system is limited to the area covered by the beacons.

2.1.2 Map-based systems

Another category of positioning systems are systems using a map of the operating area. The map which represents the operation area can be anything from a floor plan to a high-density 3D point cloud. The position is estimated relative to the map. The map is build based on measurements from sensors used in the application. The same measurements are used to align the measurements and the map. The approach used to align the input data and the map is called simultaneous localization and mapping (SLAM) [22].

The quality of a map-based positioning is depending on the accuracy of the sensor, which means an accurate sensor can provide relative positional information with an error within a centimeter. Having a map is also an advantage, since a virtual world can be designed within the assembled map. A disadvantage is the speed, because it is a challenge on its own to establish a real-time SLAM-based positioning system. Besides speed the system needs to cope with a dynamic environment as well. When movable objects like humans, movable industrial machinery and vehicle obstruct the fixed environment then the alignment process can suffer severely.

2.1.3 Global beacon systems

We categorize systems a global beacon system if the system uses GPS. GPS-based system use multiple satellite signals to estimate the absolute position of the receiver on Earth. The distance to each received satellite is determined using the time the signal needs to travel. Using those distances the position of the receiver can be calculated by trilateration, as discussed in Subsection 2.1.1.

Global navigation satellite system augmentation

Global navigation satellite systems (GNSS) augmentation is used to improve accuracy, availability and integrity of the GPS [23]. There are two types of augmentation systems, satellite-based augmentation systems (SBAS) and ground-based augmentation systems (GBAS). SBAS uses geostationary satellites to support wide-area augmentation. Geostationary satellites are satellites that rotate around Earth with the same orbital period as Earth. That makes the satellites fixed to a specific location on Earth. Europa has a SBAS called EGNOS and the USA has one called WAAS, both are mainly used for air navigation tasks like landing approach guidance. GBAS uses beacons on the ground and spawn a smaller area than SBAS, it is often used to ensure the integrity of positional data around airports. A specific GBAS technique, Real time kinematic (RTK), is more widely used.

Real Time Kinematic

RTK is used to acquire highly accurate position estimations. A GPS-receiving ground beacon is used as reference station to correct the signal received by the mobile receiver. A reference station that receives the carrier wave of the GPS signal is used to correct for errors. The carrier wave is a waveform with a higher frequency than the commonly used signal. Due to the higher

frequency the error in position estimation is smaller. The ground beacon is placed within a range of 20km of the operating area [24]. The GPS location retrieved by the mobile receiver is corrected using the beacon's GPS data. The accuracy of such a positioning system is reported to be 2mm, in theory [25].

Advantages and disadvantages

The advantage of GPS is an absolute position on Earth is estimated, which can be mapped directly into a positioning system. However there are two major disadvantages making GPS useless as input signal for our positioning system. The first disadvantage is in accuracy, since this is always off several centimeters even for the most expensive commercial configuration, RTK. Then the GPS signal bounces off materials like brick, metal and other building materials. When a signal is received indoors then the signal has bounced and gives a wrong approximation of the distance to the satellite, resulting in enormous errors.

2.1.4 Inertia-based systems

Inertia-based systems are systems which are based on sensors that measure angular velocity and linear accelerations. The measurements are performed using a combination of gyroscopes and accelerometers often referred to as an inertial measurement unit (IMU) [26]. We consider the IMU in more detail in Subsection 3.1.2. To acquire the translation and rotation relative to last known pose the measurements are integrated. Then those are added to the last known position to estimate the new pose, this is called dead reckoning. Because every new pose is based on the previous pose, the errors made each estimation accumulate resulting in a phenomena called drift. Inertia-based systems are often combined with beacon-based systems like GPS in research to deal with this drift. Others propose systems taking advantage of geometrical features, resulting in an inertia-based system combined with a map-based systems. And last but not least research shows computer vision can be used to contain the drift within proportions by taking advantage of feature tracking and object recognition for example.

Drift is a huge disadvantage of inertia-based systems, but solutions exist to deal with it. A huge advantage is no external modifications are required to the operating space, what means any arbitrary space can be used to employ an inertia-based system. We research inertia-based systems using dead reckoning without dealing with drift.

2.2 Rift

The Oculus Rift is a head mounted display (HMD) that is in development. We use the first development kit as shown in Figure 4 [27] in this thesis. The HMD has one display (1280x800pixels LCD) with two lens cups in front. Three sets of lens cups come with the developers kit; set A is for normal/far-sighted people, set B is for moderately near-sighted and set C for very near-sighted people. Inside the HMD there is an IMU fastened to the front that is named Oculus Tracker V2.

The Tracker is custom designed IMU that is optimized for a 1000 Hz refresh rate for the gyroscope [28]. The Tracker consists of a microcontroller with a 72 MHz CPU (STMicroelectronics 32F103C8 ARM Cortex-M3 microcontroller), marked by the red square

in Figure 5 [27]. A six-axis gyroscope and accelerometer combination (Invensense MPU-6000) that is marked by the orange square and a three-axis digital compass (Honeywell HMC5983) marked by the yellow square in the figure [29].

A control box is attached to the HMD. The control box contains a power converter, a display interface controller and 256KB serial flash memory [29] inside. The interfaces available on the control box are HDMI, DVI, mini USB and DC-in. On top of the control box are five buttons available: one power button and four buttons to adjust the brightness and contrast of the display.

Figure 4. Oculus Rift - Development Kit [14]

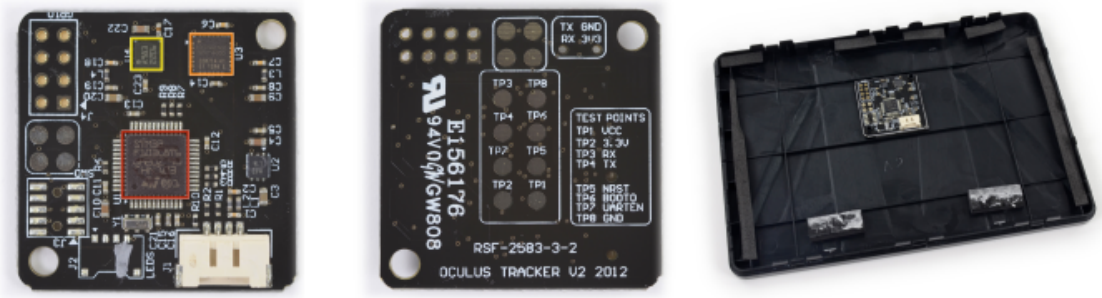


Figure 5. Inertial measurement unit of the Rift [14]

3 Inertia based positioning system

In this section we perform the analysis of the inertia based positioning system. First we discuss the positioning algorithm, the IMU and the basics of signal processing in the preliminaries. Then in the research section we consider the positioning system and how the IMU is applied to the positioning algorithm. After that we look into potential errors in the system in order to determine what we can do to reduce these errors. We find that the measurement error in the IMU signal is the biggest effector in the drift, therefore we analyze this signal. As final part of the research we look into filters; which can reduce noise in order to improve the signal.

We discuss the implementation of the system in following subsection. Then in the experiments section we explain the experiments we performed in order to be able to analyze the drift. The next subsection is used to present and discuss the results obtained by the experiments. Finally, we conclude this section by briefly reviewing the results with respect to our objectives.

3.1 Preliminaries

Subsection 3.1.1 explains dead reckoning, the main principle we use for positioning in this thesis. The sensor we use to estimate the orientation and translation, the IMU, is explained in Subsection 3.1.2. Finally, we discuss several topics on signal processing in Subsection 3.1.3.

3.1.1 Dead reckoning

Dead reckoning is the method we use for navigation in this thesis. It is the process where the current position and orientation are calculated by adding the measured translation and rotation to the last pose. A two dimensional example is illustrated in Figure 6. Consider a constant velocity of 1 m/s and a measurement interval of 20 sec. At position 1 the heading is to the east, resulting in 20m displacement to the east. If we consider position 1 as the origin (0 m, 0 m) then position 2, represented by the white circle, is calculated as:

$$\mathrm{s(t)} = \mathrm{s(t-1)} + \Delta\mathrm{s(t-1)} = (0\ \mathrm{m}, 0\ \mathrm{m}) + (20\ \mathrm{m}, 0\ \mathrm{m}) = (20\ \mathrm{m}, 0\ \mathrm{m}),$$

where s is the position and Δs is the displacement. At position 2 the heading is south east, a course change of 45 degrees. The displacement is now (14.14 m, -14.14m), with as result (34.14 m, -14.14m) being position 3, represented by the white circle in the figure. It is likely that position 3 as calculated in the example is incorrect as the behavior between two measurements is not taken into account. The error in position that accumulates each measurement is called drift. For example the problem in Figure 6 is that the time it took to change the course is not zero, resulting in an error compared to the true trajectory. The true trajectory is illustrated as the green line. The estimation of position 2 has an error due to the course change. Position 3 is calculated using the above formula, with as result the error in position 2 is added to any errors in the last calculation. In this thesis we try to minimize drift by reducing measurement and process errors.

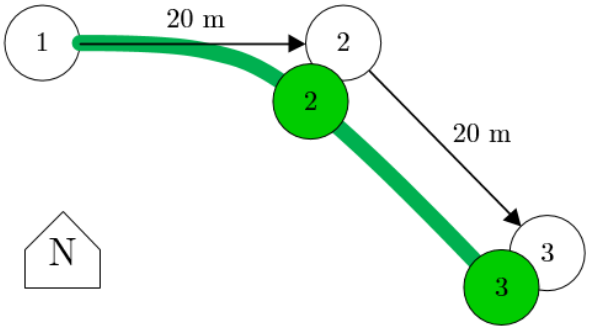


Figure 6. Dead reckoning example with true trajectory. Due to accumulated errors in calculating previous positions the currently estimated position is incorrect.

3.1.2 Inertial measurement unit

We use an inertial measurement unit that is a combination of a three-axis gyroscope, an orthogonal accelerometer setup and a three-axis magnetometer. We only focus on the accelerometers in this thesis and therefore we take a look into the accelerometer.

The accelerometer measures proper accelerations, which is the physical acceleration to which the object is subject. Practically it means that the accelerometer measures gravity if the object does not accelerate. The structure of a typical micro-electro-mechanical system (MEMS) accelerometer is illustrated in Figure 7. A MEMS is a mechanical device embedded in a chip. The structure of an accelerometer consists of a spring-mass system and a sensor system. The spring-mass system consists of a known mass which is fixed with four springs on the chip. When applying an acceleration the mass resists the change in motion, because the mass is not directly fixed onto the chip. The sensor system measures the inertia using the fixed outer plates and the mass. By charging the fixed outer plates and the mass the potential difference (Volts) can be measured. Using this potential difference the capacitance is calculated. The difference of the capacitances gives a mapped acceleration [30] [31] [32].

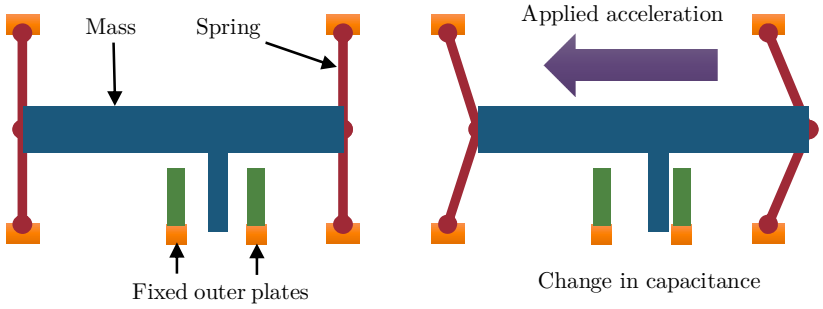


Figure 7. Schematic structure of a MEMS accelerometer. A mass attached to springs is subject to inertia. The inertia is determined by the fixed outer plates that measure the capacitance with the mass with respect to each outer plate.

3.1.3 Signal processing

We use the measurements of the accelerometers of Rift in this thesis. These measurements can be seen as three signals: x, y and z respectively. Every signal is in practice subject to noise. We distinguish two types of noise in a system: measurement noise and process noise. To understand the process we should take a look into control theory to explain open and closed loop systems. Control theory is used to actuate a system to reach a desired situation from the current situation. What we do here is the exact opposite, we want to find the current situation by observing the system with sensors.

Open-loop and closed-loop control systems

There are two conceptual systems: open-loop and closed-loop systems. An open-loop system is uses a controller and an actuator to control a process, without feedback. A closed-loop system is the same as an open-loop system but includes a feedback loop. The systems are schematically depicted in Figure 8.

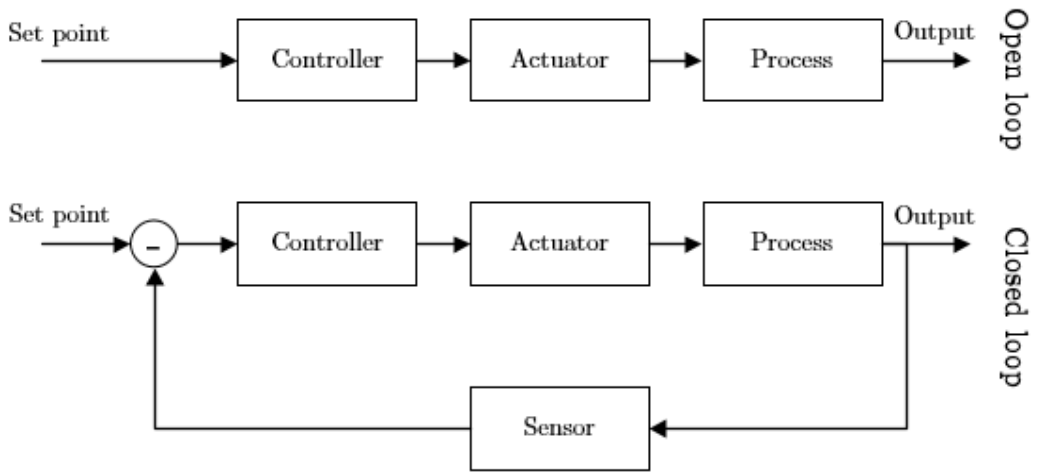


Figure 8. Schematic open-loop and closed-loop control systems. A closed-loop system includes a feedback loop where an open-loop system does not.

Noise

Disturbances on the system and measurement flaws are both considered noise. Disturbances on the system are called process errors and measurement flaws are called measurement noise. Now we use an IMU to measure the process of the boy. We expect the boy to walk in a straight line to the target location, therefore the disturbance by the boys friend can be seen as a process error. The accelerometers are subject to noise. When analyzing the noise in Subsection 3.2.4 we will see the noise has certain characteristics. The measurement noise we have to deal with is called white noise. The color of a noise signal says something about the characteristics of the power spectrum of the noise. To see the power spectrum the signal has to be converted from the time domain to the frequency domain. A power spectrum is a graph putting decibels on the y-axis and frequency on the x-axis. Then for each frequency the power in the signal is

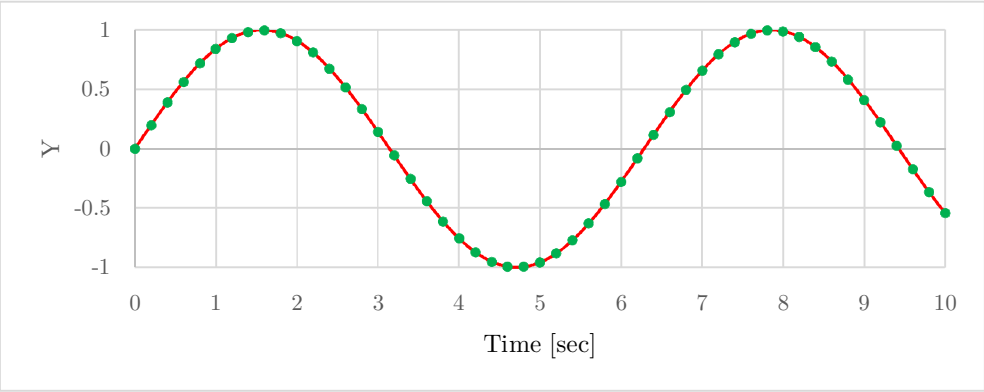


Figure 9. Continuous and discrete time signals. The line represents a continuous time signal and the dots represent a discrete time signal.

calculated. The typical characteristic for white noise is the power is more or less equal over the entire spectrum. What means there is an equal amount of noise over the entire signal. In this thesis we talk about process and measurement noise, but also about the true signal. The true signal is the signal that is not subjected to any kind of noise. There are approaches to reduce noise, like filtering.

Filtering

Filtering is used to improve the signal in such a way that the signal approaches the true signal. For example if it is impossible for a system to have signals with a frequency above 30 Hz then we can get rid of all signals with a higher frequency when we use a low pass filter. A low pass filter allows signals to "pass" which are below a given threshold frequency. We discuss filters in more detail in Subsection 3.2.5. Besides filtering we can improve the approximation of the true signal even more by increasing the sampling speed. The sampling speed is the speed we measure with, but to understand this we need to know a little more about time.

Continuous and discrete time

We model variables that change over time using two ways, continuous time and discrete time. When we look at the continuous time signal like depicted in Figure 9 as a line, we see the signal is defined over the whole timeline. While a discrete time signal is only defined at certain points in time, as illustrated by the dots in Figure 9. We measure the signal with the accelerometers which takes samples of the acceleration. Those samples form a discrete time signal. The sampling rate determines how close the discrete signal is to the continuous signal.

Sampling

The acceleration in Figure 9 is sampled at 5 Hz which seem to give an accurate estimate of the continuous signal. However when sampling a part of the continuous signal is ignored, this effect is exaggerated as shown in Figure 10, by sampling at 1 Hz. The discrete signal is measured each second and everything what happens in between is not measured. We see that there is space between the continuous and discrete signal especially when the slope of the signal changes. The absolute error is summed and also displayed. We can reduce the effect of

this sampling error by increasing the sampling speed, so we ignore less data. In Subsection 3.2.4 we learn there is a maximum frequency of 20 Hz at which a human can move. We can sample with a maximum of 500 Hz using the accelerometers of the Rift, which reduces the data loss massively. We will see that in the worst case we can measure 25 samples of a peak. The peaks in Figure 9 are sampled with approximately 15 samples.

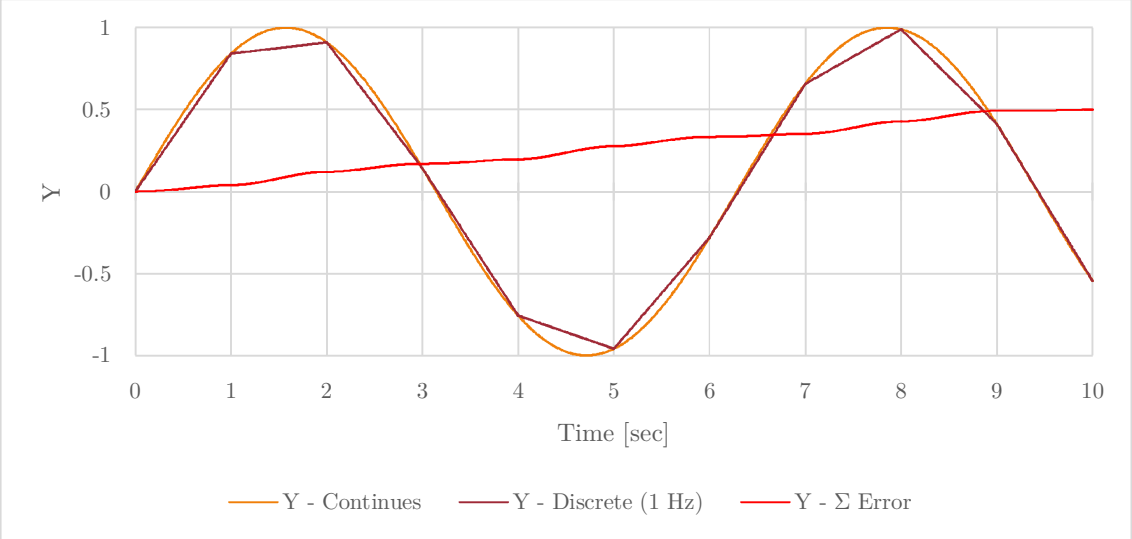


Figure 10. Sampled continuous signal. Sampling ignores a piece of data, which leads to errors.

3.2 Research

We discuss analyses, related work and relevant theory that are considered for an inertia based positioning system in this chapter. First the high level system is discussed in Subsection 3.2.1. In Subsection 3.2.2 we consider how positioning is achieved using the inertial measurements. Then we consider the effects of four potential errors in the system in Subsection 3.2.3. Signal processing, which is about signal and noise properties, is discussed in Subsection 3.2.4. We will see that filters can be used to improve signals in Subsection 3.2.5.

3.2.1 System

A high level system that uses a positioning system to have immersive augmented reality consists of two subsystems. One subsystem is responsible for the virtual world, the other for registering the real world using the positioning system as illustrated in Figure 11. The subsystem responsible for the virtual world is called the engine. The other subsystem is the positioning system. In this thesis we focus on the positioning system and we do not consider the engine.

The positioning system receives the measurements from the IMU. Those measurements are filtered and then used to update the current pose. The pose includes both position and

orientation in the real world. We do not consider the orientation in this thesis, since a filtered orientation is provided by the SDK for the Oculus Rift. The pose estimated is used to evaluate the new pose using the newly retrieved measurements.

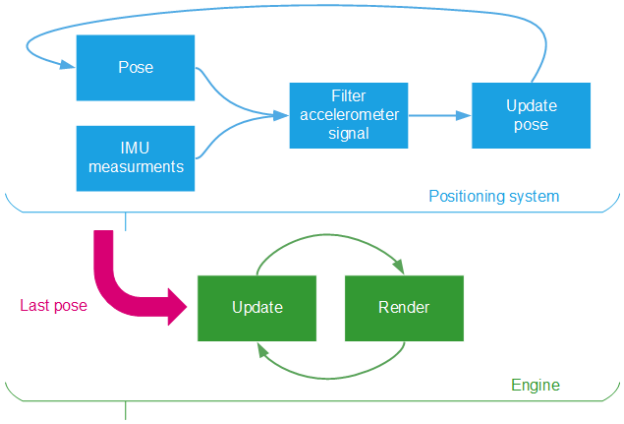


Figure 11. System - Software perspective

3.2.2 Inertial measurement unit positioning

First we look into the way we use the IMU measurements to determine the velocity and displacement. Assuming the measurements are perfectly correct the system as shown in Figure 12[1] is able to calculate the position after every measurement. The accelerometer provides a raw acceleration sample which is relative to the local coordinate frame of the Rift. The orientation is calculated by the SDK and describes the orientation of the global frame with respect to the local frame of the Rift. The global frame is defined as to the initial local coordinate frame of the Rift, therefore we call the orientation quasi global.

To calculate the new global position we need to transform the local acceleration to the quasi global frame first. This can be done using the following equation:

$$\mathbf{a_g} = \mathbf{O}\ \mathbf{a_l},$$

[1] In the perfect situation the part in the red square is not used.

where $\mathbf{a_l}$ is the local acceleration, $\mathbf{a_g}$ is the acceleration in the global frame and $\mathbf{O}$ is the rotation matrix of the orientation. The acceleration is a vector of three components which contains the measurements on the x, y and z-axis. We use basic physics to calculate the new

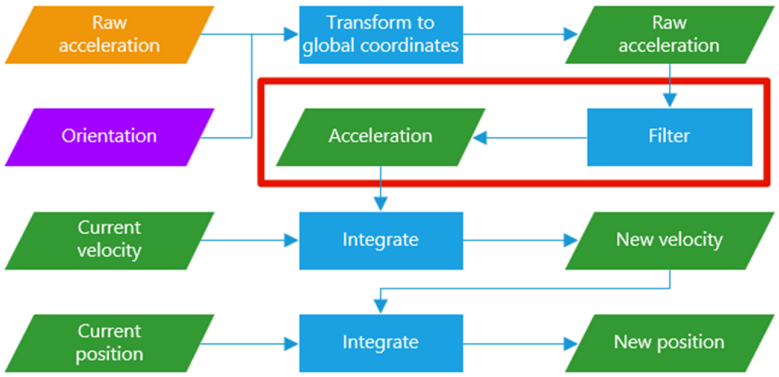


Figure 12. Simple positioning system using local acceleration and quasi-global orientation as provided by the Rift. A filter is used to reduce the error caused by measurement noise.

position by integrating the acceleration twice and adding the previous state.

$$\mathbf{v_t} = \mathbf{v_{t-1}} + \mathbf{a_t}\Delta t$$

$$\mathbf{p_t} = \mathbf{p_{t-1}} + \mathbf{v_t}\Delta t$$

We know however that there are always errors involved. The sensors are subject to measurement noise and the signal is discrete, therefore there is an error in the integration method. In Subsection 3.2.3 we look four potential errors in the system. We will see that the measurement noise causes significant error and that we need to integrate a filter in our system. Therefore the system requires a small modification (marked by the red rectangle) as shown in Figure 12. The task of the filter is to improve the measurement signal. We use the filtered acceleration to calculate the new velocity.

3.2.3 Errors in the system

We found that there are four potential errors in the system. The first error we look into is measurement noise. Then we will consider the effects of numerical integration on the system. We consider the effects of different sample rates on the system and finally we consider drift.

Measurement noise

The signals of the accelerometers are subject to a white noise. We can analyze the effects of the noise on the pose in the virtual world while we monitor and log the virtual position. To extract the noise from the real signal we place the Rift onto the ground. In this position the true position remains fixed and the true acceleration therefore equals zero if we subtract the measured gravity applied on the IMU.

Figure 13 depicts a plot of the raw acceleration measurements. We observe noticeable noise, since the true signal is a flat line at a constant acceleration of 0 m/s^2. The noise is not negligible in the system, because if those accelerations are integrated twice there would be displacement of at least centimeters after a few seconds. We should attempt to filter the signal to reduce the noise.

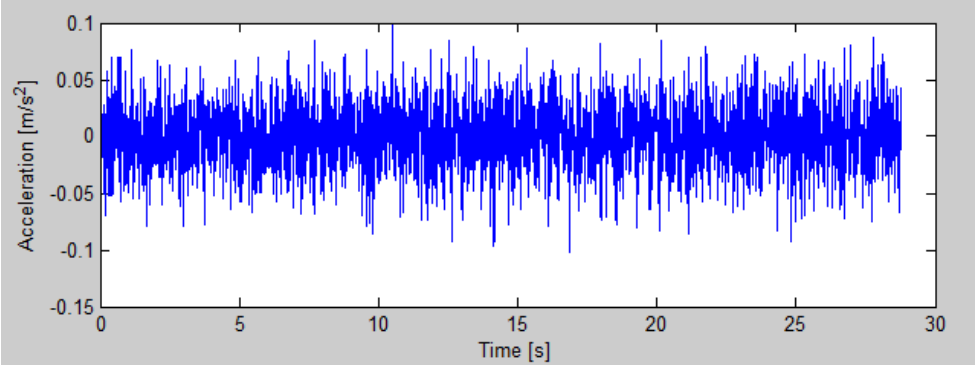


Figure 13. Plot of the raw acceleration measurements, showing noticeable measurement noise with considerable impact on the system

Numerical integration

We integrate the acceleration to acquire the velocity and displacement. The integration method introduces a new error, to which we shall refer to as the integration error. Integrating over a discrete signal results in an error, because we assume the value stays the same during the time step while the value can change. In Figure 14 we see an example of an integration error as result of the Euler integration method. The blue curved line in the figure represents a continuous time signal, for instance a path. The red dots marked by $A_{0..4}$ represent the samples of the discrete time signal. In between the samples the red lines are assumed to be straight resulting into an offset between the red and the blue line. This offset is referred to as integration drift. We analyze the impact of the integration error by comparing two integration

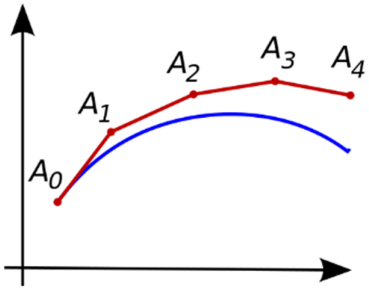


Figure 14. Euler integration method approximation error

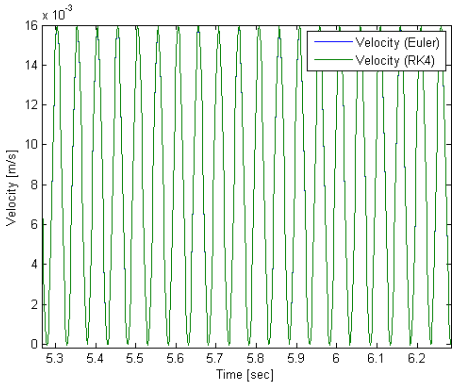


Figure 15. Velocity after integration using Euler and RK4 integration methods

methods. We compare the intuitive Euler integration and the more complex and theoretically more accurate Runga Kutta integration method (RK4). We generate a signal which is a sine with a frequency of about 20 Hz, an amplitude of 1 m/s^2 for 2π seconds. More specifically we use the following function to generate the signal:

$$y(t) = \sin(125t),$$

We sample the signal with 500 Hz and process it with both the Euler and RK4 integration methods, which results in two velocity signals. The amplitude of both signals is 0.008 m/s with a positive bias of 0.008 m/s. After 2π seconds the velocities should be exactly 0.0 m/s. The last second of both signals is illustrated in Figure 15. The velocity signals are not distinguishable using a sample rate of 500 Hz. At the end of the signal we see the velocity is practically zero. The integration error is cumulative, therefore we can assume that the error at the end is the largest error. The precise values at the end and thus errors are:

$$\varepsilon_{\text{Euler}} = -3.4098 \bullet 10^{-16}$$

$$\varepsilon_{\text{RK4}} = -3.6776 \bullet 10^{-16}$$

Those errors are negligible because the precise value of the acceleration signal is:

$$y(2\pi) = -8.0354 \bullet 10^{-14}$$

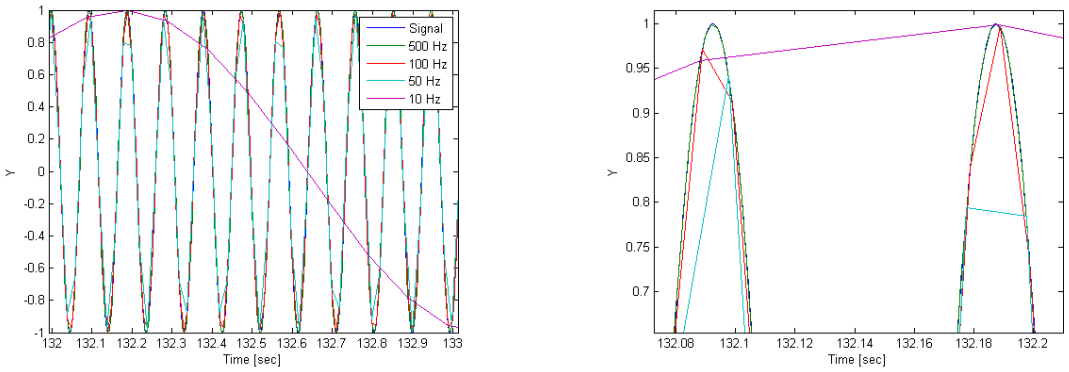
Therefore we do not consider numerical integration a problem in this thesis. The integration error is small due to the high sampling rate, this causes the curves to be estimated better. It is clear that the accuracy improves when the time step becomes smaller. Even if we consider the signal being the acceleration of the IMU, which is integrated to a velocity then the integration error has an insignificant effect compared to the measurement error.

Sampling rate

The time step is determined by the sampling rate. We found that the Rift has a maximum sample rate of 500 Hz. We can update the system at 500 Hz but that would require more computational power than when we sample at a rate of 100 Hz. Therefore if we could sample at sample rate of 100 Hz without significant losses in terms of accuracy then that would be better. However, we figured that a larger time step affects the integration error negatively. We might also find that due to a lower sample rate parts of the input signal are ignored. Therefore we analyze an input signal, a sine with a frequency of about 10 Hz, as illustrated in Figure 16. We sample this signal with sample rates of 500 Hz, 100 Hz, 50 Hz and 10 Hz to show how much of the signal is ignored.

At a sample rate of 10 Hz we see the measurements cannot keep up with the signal, therefore a large part of the signal is not taken into account. This phenomenon which makes the 10 Hz have a much lower frequency is called aliasing. This aliasing error occurs when the bandwidth of the sampled signal is larger than half the frequency at which we sample [33]. If we want to avoid this aliasing error then we need to sample faster than 40 Hz, since our signal

is bandwidth is 0-20 Hz. For sampling at 50 Hz and 100 Hz we see the measurements can keep up with the signal because we avoid aliasing. However, the minima and maxima of the signal are not sampled close enough to the extreme, which can lead to large errors. When we sample at 500 Hz the curve is approached properly in our case, since we expect the frequencies in our case are limited to 20 Hz as we will discuss in Subsection 3.2.4. We conclude that we have to sample as fast as possible to prevent data to be ignored.



(a) (b)

Figure 16. Signal sampled at different sample rates. (a) A sample rate of 10 Hz is not able to keep up with the signal and higher rates do keep up. (b) Sample rates of 50 Hz and 100 Hz ignore the tops of the peaks partially. The ignored part gets smaller as the sample rate increases.

Drift

Drift occurs due to the dead-reckoning. Dead-reckoning is used to determine the position. Drift is a cumulative error that always results in significant displacement. The effect of the drift on the system is that there is a bias introduced in the alignment of the virtual world to the real world. This bias grows over time and to the user it will look like the virtual world floats away.

In [34] a system is proposed where the drift is compensated for with are result a significant reduction of the drift. However, the researchers show that after a walk of 125 m the drift has accumulated to a positional error of 1.06% of the distance traveled. What means the error is larger than 1.25 m at the end. Such an error is considerable, because for about every step a user takes the virtual environment drifts approximately 1 cm away.

We have seen that the measurement noise contributes most to the drift. Therefore we need to reduce the measurement noise, which can be achieved by signal processing.

3.2.4 Signal analysis

We want to remove the noise to obtain the true signal. In order to do so we need to know as much as possible about the signal. Therefore we analyze the properties of the signal in this subsection. First we analyze the noise properties, then we analyze the signal properties in order

to see what kind of outrages we can expect. At last we show that there is a limit to the maximum movement frequency of the human body.

Noise properties

To determine the properties of the noise we place the Rift into a fixed straight position on the ground. Then we measure the acceleration as provided by the Rift itself. This acceleration is unmodified and provided from the IMU directly. Since the IMU is fixed we may assume an acceleration of 0 m/s^2 on both the horizontal axes and ~9.81 m/s^2 on the vertical axis. This means everything we measure in the horizontal axes is noise. We assume that all three accelerometers are the same and therefore are subject to the same noise. During the noise analysis we choose to analyze the acceleration of the z-axis, which is oriented forward on the Rift.

In Figure 13 we see the noise has a maximum amplitude of approximately 0.1 m/s^2 and the mean is 0 m/s^2. With the histogram plot as depicted in Figure 17 we show 75% of the measurements are smaller than 0.03 m/s^2 and 95% is smaller than 0.06 m/s^2. We analyze the frequency spectrum to identify the frequency properties of the noise. We transform the signal to the frequency domain first, using Fourier transformation:

$$Fy_k = \left(\sum_{n=0}^{N-1} y_n e^{-2\pi k \frac{n}{N}} \right) /L,$$

where Fy_k is the signal in the frequency domain, k is a point in the frequency domain, y_n is the discrete input signal, n is the number of samples that is considered and L is the number of samples of the signal. The frequency domain spawns half the signal frequency, this is to avoid aliasing effects [35]. As shown in Figure 18 the intensity of the noise is equally distributed over frequency spectrum which is the characteristic for white noise [36]. We conclude that the measurement noise is a zero-mean white noise with a maximum amplitude of 0.1 m/s^2.

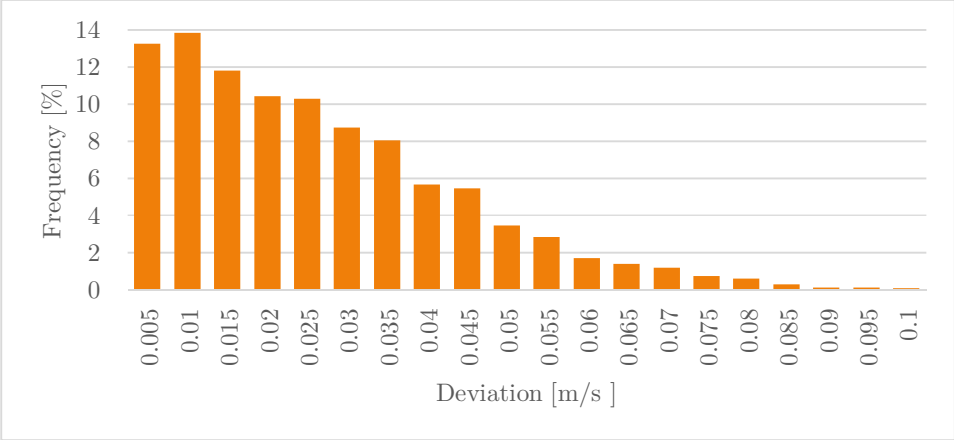


Figure 17. Histogram of the noise signal

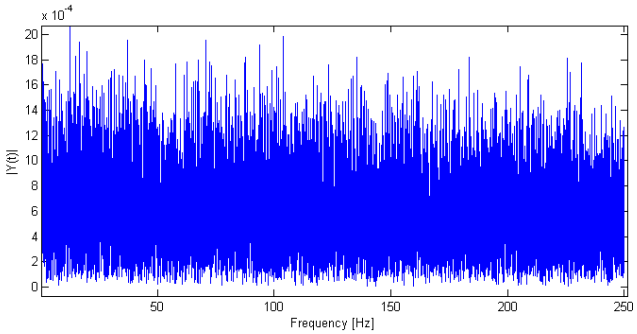


Figure 18. Frequency plot of the noise signal. The intensity along the frequencies is equal, which is a typical property of white noise.

Signal properties

Due to the noise we cannot extract the signal properties in a straightforward way. We need to retrieve the true signal. To retrieve the true signal we produce a repetitive mechanical sequence. The sequence is performed multiple times with the exact same interval for each sequence. The mechanical setup we use is illustrated in Figure 19.

To determine the true signal we split the signal on the intervals, with this we retrieve the sequences. Then twenty sequences are averaged, the result is considered the true signal. We consider the averaged signal of the Y-axis to be the true signal, which is the average of twenty sequences as depicted in Figure 20. The sequence is a tilt motion, where position 1 is the initial position, to reach position 2 a hydraulic actuator is used. We made the sequence interval exactly 4.1 seconds, consisting of two sub cycles up and down. The down cycle takes exactly 2.0 seconds and the up cycle takes exactly 2.1 seconds. The actuator needed approximately 1.0 sec to reach the other position. The grey box represents the Rift and arrows attached to box represent the IMU axis, where green is y and blue is z. The motion produced is curved around the rotation point. This results in an upward and forward motion where the upward displacement (5cm) is larger than the forward displacement (2.5mm).

Even when the motion is small it can occur that the acceleration outrages are larger than 1 m/s^2 as we see in Figure 20. We also see that the mean of the Y-axis is 9.81 m/s^2, which is the result of gravity, which means the accelerometers work as expected.

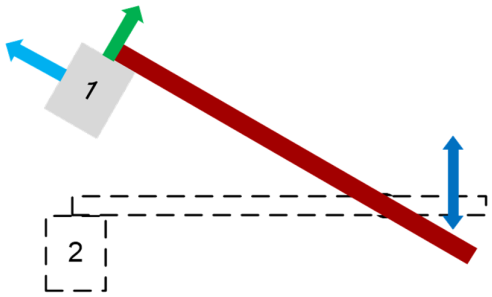


Figure 19. Mechanical setup, performing a repetitive tilt motion to extract the true signal produced by this motion.

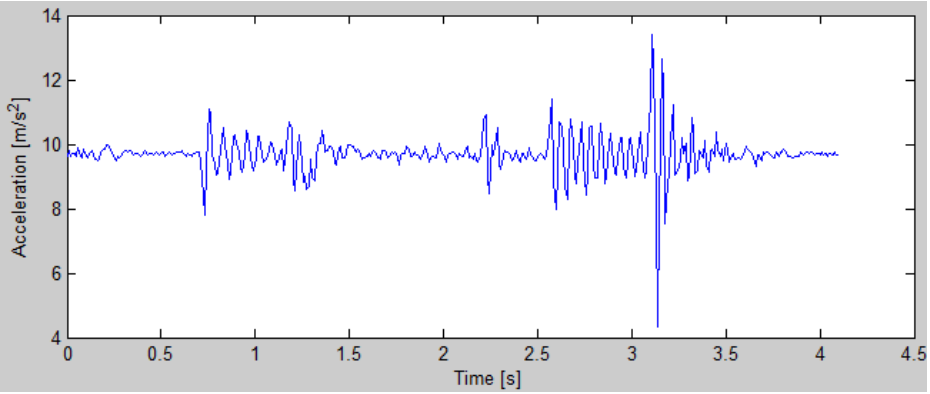


Figure 20. Averaged signal of the Y-axis acceleration during the mechanical sequence.

Human movement frequencies

The system we use is physically constrained by the limitations of a human body. What means we can apply constrains on the accelerations we measure. Research has proven that movements performed of the human body are below 20 Hz, 99% of the energy is contained below 15 Hz and 98% of the energy is contained below 10 Hz [37]. This energy is the captivated by the amplitude of the frequencies in the Fourier transformed function. It practically means we can ignore all signals with a frequency over 20 Hz.

Most important to know is which amplitudes occur when we move around. In Figure 21 we show a sequence which is recorded while performing minor movement actions. In the first section we walk slowly from point A to point B and back in a straight line. In the middle section we look around actively and in the third section we jump several times. The y and z axes show larger amplitudes than the x-axis, which can be explained by the orientation of the accelerometer axis within the Rift. The x-axis is oriented in sideways direction, the y-axis is oriented downwards and the z-axis is oriented frontward. We see that even when we walk slowly the accelerometers measure rather large peaks. In z direction the outrages are up to 10 m/s^2. When we look around the x and z axes receive noticeable acceleration up to 5 m/s^2. Finally in the jump process we see peaks larger than 10 m/s^2 on the y-axis.

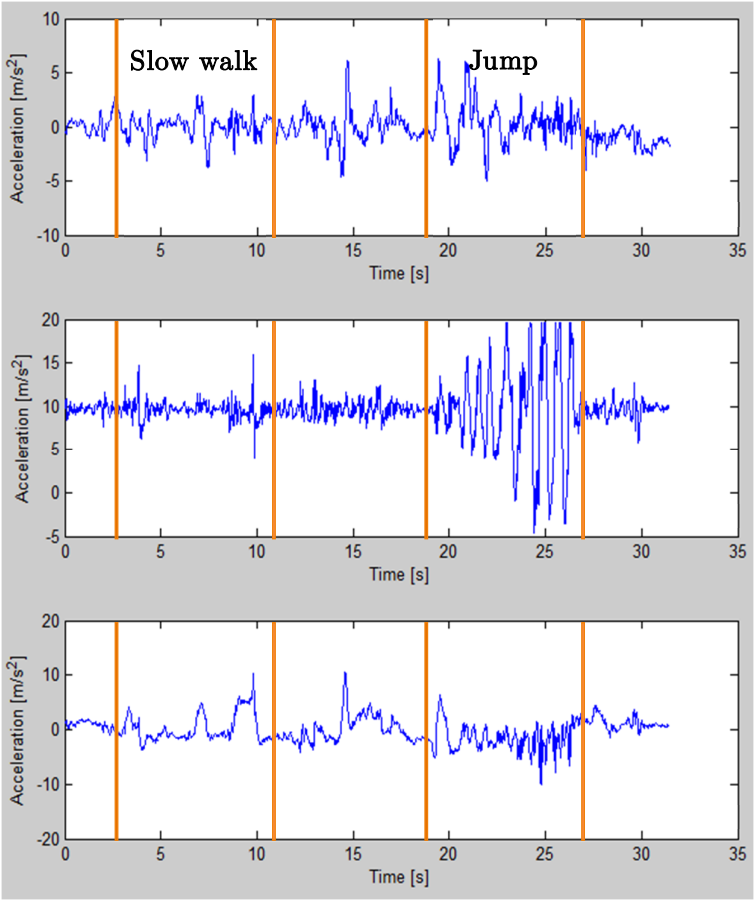


Figure 21. Plot of accelerometer readings during a test sequence behind a desk

3.2.5 Filters

We want to improve the signal to noise ratio and therefore the signal needs to pass a filter. In this subsection we will discuss filters that are used in related work. We will take a closer look at low pass filters and at the Kalman filter. We discuss the implementation details for each filter in Subsection 3.3.3.

Related work

In the literature we see that there are several filters commonly applied. The most used filters for positioning systems that involve an IMU are the complementary filter and the Kalman filter. However, particle filters, moving average filters and descendants of the Kalman filter are also used. The complementary and Kalman filters are both state estimators [38], which means the Kalman Filter is able to replace the complementary filter. This is because the complementary filter combines multiple filters, like a high and low pass filter to estimate a state. The Kalman Filter uses mathematical models to predict and estimate a state. These mathematical models are able to encapsulate such combinations of filters. The Kalman filter

is theoretically more accurate but more computationally expensive than the complementary filter as well [38] [39].

In the work of Fredrik Gustafsson et al. [40] a particle filtering approach is considered for several positioning, navigation and tracking applications. However, the focus is on aircraft positioning, navigation and tracking. The researchers claim that the particle filtering framework gives better results than the Kalman filter. The algorithm uses a recursive Bayesian estimator on the particles, which is a predecessor of the Kalman filter and also a state estimator. Despite the accuracy offered by this approach, the speed of the framework is not very promising for a real time solution. The real-time framework runs with 2Hz in MatLab with 2000 particles. Therefore the particle filter is not considered in this thesis.

The complementary filter is used to estimate orientation using an IMU [39]. It combines a low-pass to filter the acceleration signals and a high-pass filter to filter the gyroscope signals. Both sensors are "fused" to acquire a better estimate of the orientation. However, we would only use the low pass filter to filter the accelerometer signals, since human body movement never exceeds a frequency of 20 Hz.

Low pass filter

We use a low pass filter in this thesis because the system has to deal with a maximum frequency of 20 Hz. The maximum frequency is limited by frequencies the human body movements. A low pass filter has several parameters that can be configured and two major properties to consider. The parameters required by the low pass filter are the sample rate, the cut-off frequency and the order. The cut-off frequency is the maximum frequency we want to allow for the system, in our case this would be 20 Hz.

The order of a low pass filter determines the properties of the filter, the magnitude response and the phase response. Ideally we would filter every signal with a frequency higher than the cut-off frequency, but this cannot be done practically. The reason why this cannot be done is because the low pass filter has a transfer function to determine a magnitude response. This transfer function is a certain order, to cut off the frequency exactly we would need a transfer function with an infinity order. The magnitude response determines the effect of each frequency on the signal; a few examples are illustrated in Figure 22. In the figure we see several types of magnitude responses, which we discuss in detail in the next paragraph. We can see the frequency influence degrades as we progress from left to right over the horizontal axis. The graphs show the frequency on the horizontal axis. On the vertical axis we see the magnitude to which the frequency passed. If the magnitude is 1 for a certain frequency then the whole signal is passed, but if the magnitude is 0.2 only 20% of that frequency is passed. The steepness of the magnitude response is determined by the order and type of low pass filter we choose. Higher order transfer functions result in steeper responses, which means higher orders are better in terms of cutting off the frequencies. However, higher orders are worse terms of the phase response, because the slope of the phase response becomes steeper as the order gets higher causing an increase in phase shift. Phase shift is bad in our case because it would introduce lag to the system.

There are multiple implementations possible for a low pass filter. Those are divided into two categories, infinite impulse response (IIR) and finite impulse response (FIR) filters. A FIR is an impulse response that enters a rest state at some point in time. For example when you push a swing several times it will finally end-up at the rest position. An IIR does not come to rest after some point in time, like our system which is a human walking around. There are several types of IIR filters we consider. They are called the Butterworth, Chebyshev Type I and II, Elliptic and Bessel filter. Each filter has its own characteristic magnitude response, as shown in Figure 22.

The magnitude response of the Bessel filter is far from ideal, because the steepness of the response so poor that it partially filters frequencies lower than the cutoff frequency and it partially allows higher frequencies to pass. The Bessel filter tries to realize a flat phase response which explains the poor response. We want the low pass filter to consider all allowed frequencies to have the same influence on the filtered signal and the higher frequencies to have no effect. Therefore the Bessel filter is not used, since it does not consider all lower frequencies the same and it allows higher frequencies to pass through. Those higher frequencies are measurement noise by definition due to the human body movement frequency constraint.

The Chebyshev Type I and the Elliptic filter are not used for the same reason as the Bessel filter. Both filters are also likely to have large phase shifts as well. We can allow a little phaseshift, because off the refresh rate of the virtual environment. The time between each update is the maximum time a phase shift may cause, which is about 16 ms for a refresh rate of 60 frames per second. Chebyshev Type II has a flat pass band in the magnitude response. A pass band is the range of frequencies allowed to pass. With flat pass band we mean all allowed frequencies are passed through with a magnitude of 1. A stop band is the range of frequencies we want to be filtered. Unfortunately the Chebyshev Type II filter has a bounce in the stop band which allows higher frequencies to have influence on the filtered signal.

We see that the Butterworth filter has a flat pass band and a flat stop band and even the steepness is reasonable. Higher frequencies have minor impact on the filtered signal, therefore we use a Butterworth filter in this thesis.

To be more precise we use a second order Butterworth filter, because that gives us the best trade-off between the magnitude response steepness and the phase shift. For a cut-off frequency of 20 Hz at a sample rate of 500 Hz the phase shift is less than $\frac{1}{2}\pi$ for 20 Hz. What means in a worst case situation the delay would be 7 samples at 13 Hz. Therefore the maximum delay is 14 ms. The refresh rate of the screen of the Rift is 60 Hz, which means each frame takes $\frac{1}{60}$ sec and therefore the maximum delay is within this time and the user cannot notice any delay due to the filter. A third order Butterworth filter would exceed this limit, but in this case the second order provides decent steepness in the response after the cut-off frequency. It provides flat pass band, which we want because every low frequency should have the same magnitude. And it also provides a flat stop band, which we want because higher frequencies have a bad influence on the measurement since they are noise by definition.

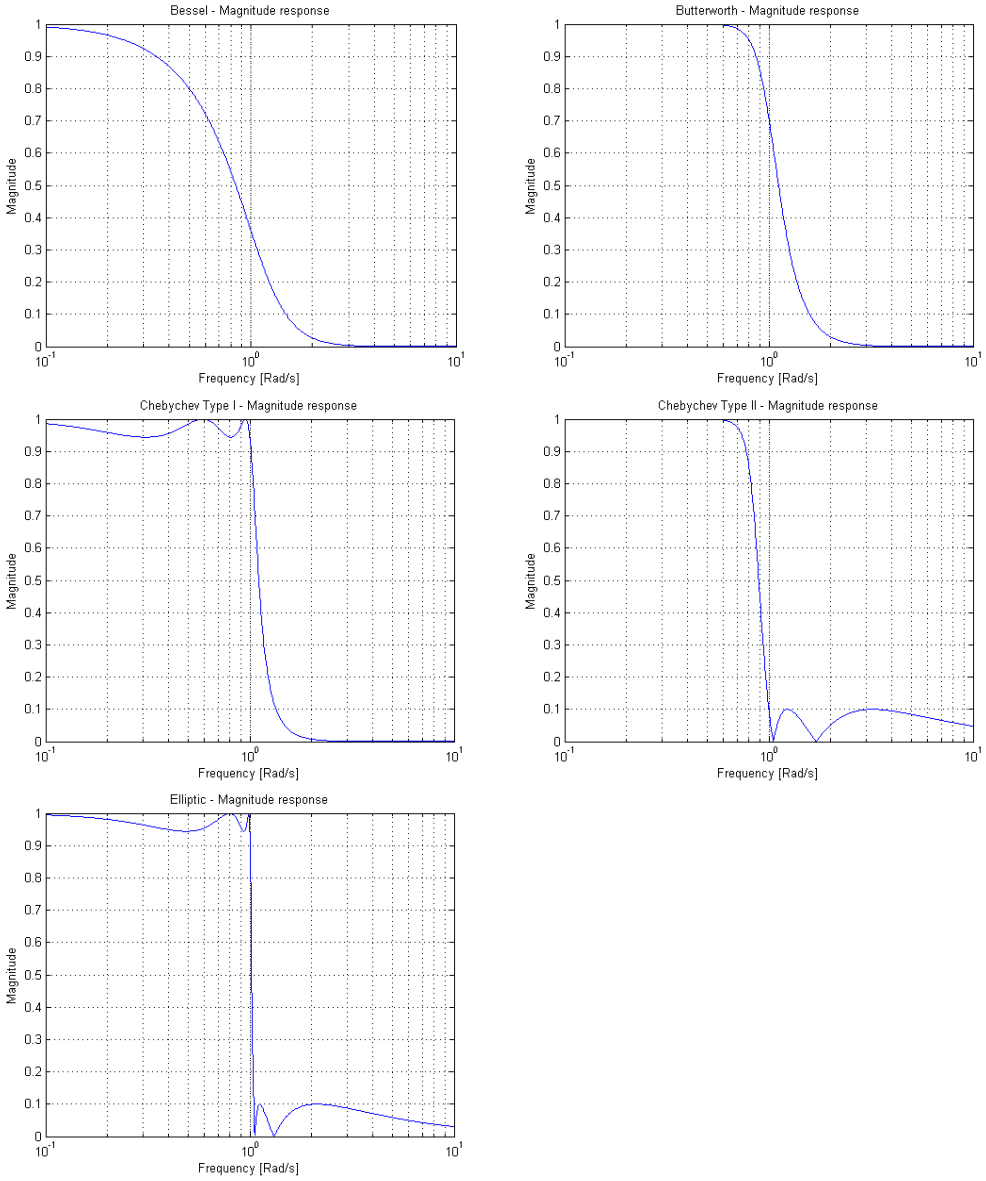


Figure 22. IIR filters magnitude responses. Each filter has certain characteristics, the Bessel filter concentrates on keeping the phase response as flat as possible. Butterworth keeps the pass band as flat as possible. The others focus on dropping the magnitude response as fast as possible on the cut-off frequency.

Kalman filter

The Kalman filter does not filter a range of values like a low pass filter, but instead it estimates the next system state. This estimation process consist of three steps; the prediction, observation and update step. In the prediction step the current state is estimated using the previous state. In the observation step the measured signal is compared to the predicted signal

and finally in the update step the estimate of the new system state is calculated. We will walk through the process to understand how the Kalman filter works.

State and model definitions

The Kalman filter is a system state estimator. Therefore it is required to know how the system is defined in terms of a state. As a system state we consider a vector to combine acceleration, velocity and position each time step. The state $\mathbf{x}_k$ is defined as:

$$\mathbf{x}_k = \begin{bmatrix} p_k \\ v_k \\ a_k \end{bmatrix},$$

where p_k is the position, v_k the velocity and a_k is the acceleration at time step k. Each next step can be determined by using dead reckoning. To do this we need a state transition matrix, so we know what to do with each variable the next step. In our case we know the following discrete formulas apply:

$$v_{k+1} = v_k + am_k \Delta t$$

$$p_{k+1} = p_k + v_{k+1}\Delta t = p_k + v_k \Delta t + am_k \Delta t^2,$$

where am_k is the last sampled acceleration by the accelerometers. To be able to calculate the next state in one formula we define a state transition matrix. In this case we will use state transition matrix to integrate the acceleration to the velocity and the position. If we remove all components which are included in a state from $\mathbf{v_{k+1}}$ and $\mathbf{p_{k+1}}$ then we are left with two times Δt and one time Δt^2. This we put in the state transition matrix defined as:

$$A = \begin{bmatrix} 1 & \Delta t & \Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

Then using the state and the state transition model we can model the system to calculate the next state as:

$$\mathbf{x_{k+1}} = A\mathbf{x_k} + w_k$$

$$w_k \sim N(0, Q),$$

where w_k is the process error and Q is the process error covariance matrix. $w_k \sim N(0, Q)$ means that w_k is an arbitrary value from the normal distribution with a mean of zero and an standard deviation of Q. Q defines the process error covariance matrix. Therefore the process error is a multivariate normal distribution which spawns a probability density function (PDF).

We can also verify that this state transition matrix is correct by applying it to the system model, as result we expect the formulas seen at v_{k+1} and p_{k+1} to appear in the result. We use a process error of zero, so $w_k = 0$ for clarity:

$$\mathbf{x_{k+1}} = \begin{bmatrix} p_{k+1} \\ v_{k+1} \\ a_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \\ a_k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} p_k + v_k \Delta t + a_k \Delta t^2 \\ v_k + a_k \Delta t \\ a_k \end{bmatrix},$$

We observe that the formulas are equivalent to v_{k+1} and p_{k+1} and thus the state transition matrix is verified. Besides a state transition matrix we need to model the measurement input as well, at this point several sensors can be combined. With the measurement model we tell the system how much each sensor affects the input. In our case the only measurement we provide at this point is the acceleration from the accelerometer and therefore it has full effect on the measurement. The measurement model is defined as:

$$z_k = H\mathbf{x_k} + u_k$$

$$u_k \sim N(0, R),$$

where z_k is the measurement, H is the measurement transition matrix, u_k is the measurement error and R is the measurement error covariance matrix. We know that accelerometers are subject to white noise, this noise is modeled by u_k which is another normal distribution with a mean of zero and a standard deviation of R. The matrix H allows us to determine the contribution of the measurement to the estimated signal. We only measure acceleration, therefore H is defined as:

$$H = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

The measurement model will now only provide the acceleration:

$$z_k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ v \\ a \end{bmatrix} + 0 = a$$

This suffices for the setup we use in this thesis, but if we would have two accelerometers oriented the same then we could simply average their measurements by using a measurement model like:

$$z_k = \sum_{m=1}^{n} H_{k_m} \mathbf{x_{k_m}} + v_k = \begin{bmatrix} 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} p_1 \\ v_1 \\ a_1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0.5 \end{bmatrix} \begin{bmatrix} p_2 \\ v_2 \\ a_2 \end{bmatrix} + 0$$

$$z_k = \frac{1}{2} a_1 + \frac{1}{2} a_2 = a$$

With this measurement model both sensors would have the same influence on the measurement.

Filter step

The Kalman filter uses both the measurement model and the system model to predict the next system state, through different steps. We will now consider the prediction step in detail. In the prediction step we predict two variables, the system state and the error covariance. Predicting the system state is straightforward, we use the previous system state and multiply

that with the state transition matrix. Note that we predict the next state using the previously measured acceleration. The formula to calculate the predicted state $x_{predicted}$ is:

$$\mathbf{x_{predicted}} = A\mathbf{x_{k-1}}$$

The predicted error covariance is calculated based on how the error is evolving over time, therefore it gives the Kalman filter information about the accuracy of the estimation. The formula to calculate the error covariance $P_{predicted}$ is:

$$P_{predicted} = AP_{k-1}A^T + Q$$

Then the next step is the observation step, where we consider the new input. We calculate the deviation between the measurement and the predicted measurement, this deviation is called innovation. Note innovation includes the measurement noise. The innovation $\tilde{y}$ is calculated as follows:

$$\tilde{y} = z_k - H\mathbf{x_{predicted}},$$

where z_k is the newly acquired input. We calculate the predicted measurement by substituting the predicted state in the measurement model. The next formula of the observation step is to calculate the innovation covariance matrix S. The innovation covariance matrix is used to determine the innovation of the error covariance matrix. This innovation describes how the error covariance matrix evolves.

$$S = HP_{predicted}H^T + R$$

The last step is the update step, which calculates the Kalman gain and the estimated system state and the estimated error covariance. The Kalman gain functions as weight factor for the influence the innovation has on the estimated system state. The formula for the Kalman gain K and the predicted state x_k are:

$$K = P_{predicted}H^T S^{-1}$$

$$\mathbf{x_k} = \mathbf{x_{predicted}} + K\tilde{y}$$

We see that the predicted system state is corrected by the innovation using the gain. That means if the gain is one the new state is equal to the measured state. Finally we update the error covariance matrix P_k using the gain by:

$$P_k = (I - KH)P_{predicted}$$

The influence of the measurement has effect on the potential error by involving the gain.

Error covariance matrices

To determine the measurement error covariance and the process error covariance is not straightforward, however, there are models available describing our problem. We use the

Wiener process acceleration model (WPAM) for the Kalman filter. This model has the same transition models as we described above, but also describes the error covariance matrices [41]. The process error covariance matrix Q is defined as:

$$Q = q \begin{bmatrix} \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix},$$

where q is a scalar value we have to determine. We can see that the matrix consist of nine elements. This is to apply a distribution over the elements in the state we defined earlier, which also contains nine elements. The measurement error covariance matrix R is defined as:

$$R = [r^2],$$

where r is a scalar value we have to determine. This matrix consists of only one element, which represents the error in the measurement value, which is a scalar value.

Before we determine r and q we will analyze the effects on the signal of both parameters in order to make an initial guess for the values. We expect a low value for q since the process error, apart from the drift, is small. For a low value for q we expect the signal would filtered a little but the responsiveness of the signal is not harmed. For r we expect a higher value in order to reduce the measurement noise. Our expectation is that the signal will be filtered fine but at the cost of responsiveness since we will filter a part of the true signal as well.

In order to do an initial guess on the parameters we analyze their effects. We created three signals subject to a white noise, as illustrated in the top left plot of Figure 23. As first initial guess for the error covariance matrices we choose values of $r = 1$ and $q = 1$. The result is shown in the top right plot of Figure 23, where we see a significant reduction of noise.

In the middle row of Figure 23 we set the value of r to 0.1 and 10. The result of $r = 0.1$ is that the error is considered smaller and thus the signal more likely, which means more noise appears. Using $r = 10$ results in smoother curve because changes in the measurement value are considered less likely than our default case. However, we see the noise is reduced almost entirely. The only side effect is that the signal is delayed with respect to the true signal, and thus reacts slowly on any change.

We consider the same values for q as we did for r in the bottom row of Figure 23. We see the changes in q have a more subtle effect on the signal and the reverse effect of r as we would suspect. If the process covariance error grows larger then the average error in the system is growing larger as well. Therefore we see a reduction of noise when q is small, because the system has a small average error and is confident of the prediction. However, using a large q the system is less confident of the prediction made and is affected more by the measurement.

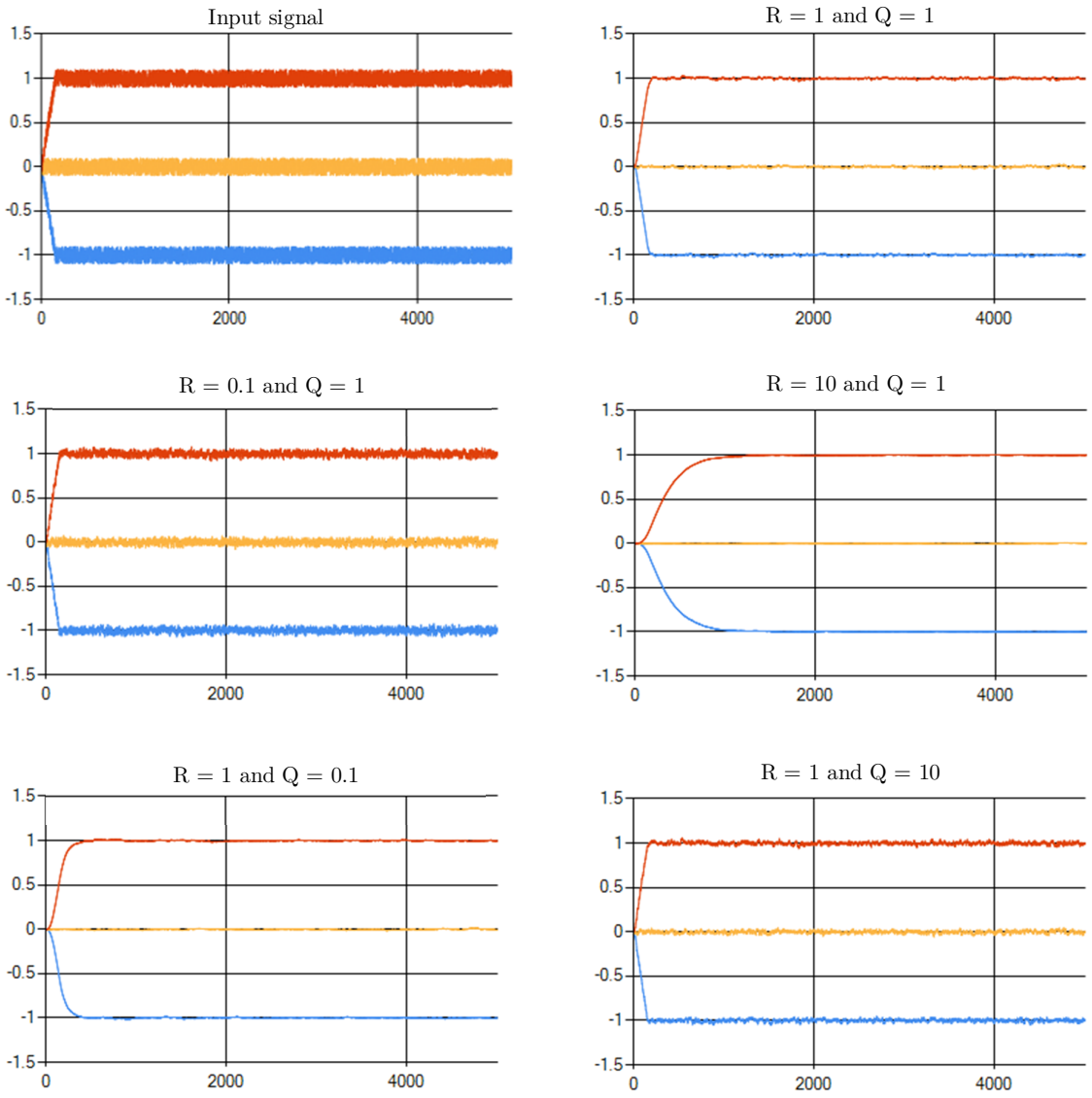


Figure 23. Effects of error covariance matrices on the filtered signal. The top row shows the input signal without filtering and the initial guess which is reducing the noise a lot. In the second row the values for the measurement error covariance matrix is shown with values of 0.1 and 10. On the third row the process error covariance matrix multiplier is given a value of 0.1 and 10.

3.3 Implementation

We discuss the implementation of the inertial positioning system in this subsection. First we consider the programming environment in Subsection 3.3.1. Then we discuss the software architecture in Subsection 3.3.2. In Subsection 3.3.3 we discuss the implementation of the Butterworth low pass filter and the Kalman filter. Then in Subsection 3.3.4 we will take a brief note about the implementation of the integration methods and how they are integrated in the system.

3.3.1 Programming environment

We use C# .NET as programming framework, because it has several beneficial aspects for prototyping. This includes native support for threading and graphical user interfaces using WPF. Another benefit is the garbage collector, which deletes unreferenced managed objects and prevents memory leaks.

We use a wrapper which is a modified version of RiftSharp[2] to access the C++ Oculus SDK 2.5.0 as downloadable from the Oculus website. To be able to give insight into the positioning system during the research, we use SlimDX[3] which is a wrapper for the DirectX APIs. We use the mathematic framework of SlimDX to implement linear algebra in the system, like the Kalman filter for example.

3.3.2 System

The positioning system consists of several components: the sensor, the filter and the black box. The sensor component is responsible for the measurements and the filter is responsible for the filtering the signal. A black box combines the components and keeps track of the current state. Because there will be more sensors than only the IMU in the final implementation we choose a modular architecture for the system. This allows us to run each sensor module in a separate thread.

Overview

In the schematic overview of the system as illustrated in Figure 24 we see the separation between the components within the system. The inertial measurement unit module runs in a thread where it samples the acceleration and orientation with a frequency of 500 Hz. The components communicate using events. This simplifies cross thread calls and eliminate the possibility of a deadlock. Because of this the black box is event driven. However, we do not have a loop in the system we consider the fastest input loop as our main loop, which is the IMU loop.

The event from the IMU to the black box contains the raw acceleration to which the global frame is subject with respect to the sensor and the orientation in pitch, roll and yaw

[2] https://github.com/davyloots/riftsharp
[3] http://slimdx.org/

angles to which the global frame is subject with respect to the sensor. First we transform the acceleration to the global frame and then the acceleration is filtered. We apply dead reckoning on the filtered acceleration and store the position and orientation in an object. Any application using the positioning system can access the singleton instance and retrieve the latest known position and orientation. We do not use an event because it would require the higher level application to update at least 500 times per second. In this way the sample rate remains optimal and the higher level application can get the required information "just-in-time".

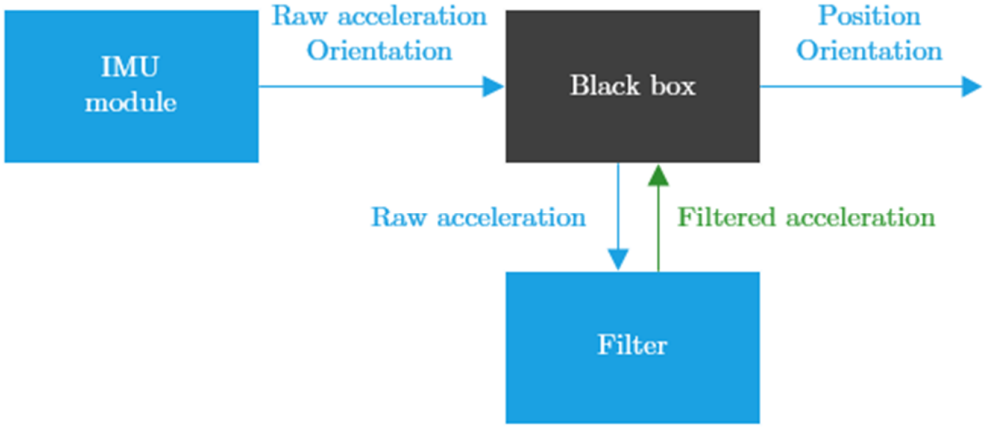


Figure 24. Inertial positioning system overview

Module

We use the RiftSharp wrapper to initialize a singleton instance of the Rift object in a separate thread. All calls via the wrapper are unmanaged code, therefore we delete the Rift object when we are done with it to avoid memory leaks. When the initialization is complete we start an infinite loop that samples the sensor at a frequency of 500 Hz. We fire an event with the sample each loop, so the positioning system can use it. Note that the module fires an event to update the positioning system at 500 Hz, but the positioning system does not fire an event every update.

Black box

When we receive the IMU sample in the black box we first put change the perspective, because we want the acceleration and orientation the sensor is subject to with respect to the global frame. We change the perspective by inverting the values. In the same step we transform the acceleration to the global frame. With the acceleration with respect to the global frame we measure a constant negative acceleration on the y-axis, which is gravity. We subtract gravity from the oriented acceleration as well. The following steps are performed to calculate the acceleration and orientation to which the sensor is subject:

$$\mathbf{a_g} = (\mathbf{O_l}(-\mathbf{a_l})) - \mathbf{g},$$

$$\mathbf{O_g} = -\mathbf{O_l},$$

where $\mathbf{a_l}$ and $\mathbf{O_l}$ are the values provided by module and $\mathbf{g}$ is gravity.

Next we filter the acceleration, but we have two cases here; the low-pass filter and the Kalman filter. The low-pass filter requires the raw acceleration as input and returns the filtered acceleration as output as depicted in Figure 24. The acceleration is processed by the dead reckoning algorithm where we assemble a complete state including filtered acceleration, velocity, position and orientation. The Kalman filter requires a measured state, which includes velocity and position. Since we only have acceleration we integrate the acceleration to obtain the velocity, then we integrate the velocity to obtain the position using the Runga-Kutta integration method. As output we get a partial state including the filtered acceleration, estimated velocity and estimated position. The orientation is added to the partial state to complete the state. The complete state is saved so it can be requested by the higher level application.

3.3.3 Filters

We have selected two filters to implement, the Butterworth filter, which is a low-pass filter and the Kalman filter. We know for low pass filters there is a tradeoff between the steepness of the magnitude response and the phase shift. The best tradeoff seems to be when using a second order Butterworth filter, since the phase shift is than smaller than 16 ms. We assume a typical higher-level application runs with 60 frames per second, which means the time step is about 16.67 ms.

Low-pass filter

The Butterworth filter implementation is retrieved from the internet. The implementation itself is almost an exact copy of [42], which is based on the [43]. The filter requires several input parameters. The cut-off frequency which is freely adjustable, the sample rate which is 500 Hz in our case and the resonance. The resonance for the second order filter is $\sqrt{2}$ which can be retrieved from the system function:

$$B_2(s) = \frac{1}{s^2} + \sqrt{2}s + 1$$

We have implemented the low-pass filter and found that the filter did not have much effect since we could not filter the noise in the lower frequency domain. The explanation for this is that the impact of lower frequency noise on the position is much higher than high frequency noise due to the double integration. Which makes the filter practically useless, therefore we do not consider the low-pass filter in the system implementation.

Kalman filter

First we discuss the structure and the process of the Kalman filter. Then we consider the implementation of the WPAM we discussed in Subsection 3.2.5.

We provide the raw acceleration in a matrix to the Kalman filter, this matrix is the measurement matrix z_k. We pass the measured state each filter step. The measured state is relative to the global frame so the change of orientation does not interfere in such a way the

direction of the velocity is changed. The absence of this interference results in a more constant acceleration.

The measured state is divided into three one dimensional states, which we provide to a Kalman filter. Each axis is filtered by a separate Kalman filter to simplify the mathematics. We use the SlimDX matrix implementation to set the definitions of the model and to do the matrix calculations. A disadvantage of this matrix implementation is that we are forced to use a 4x4 matrix and does not allow larger matrices for more complex systems. However, for our implementation it suffices. To deal with this constraint we made most matrices homogeneous, except for the measurement transition matrix H and the process error covariance matrix Q, which are defined as:

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q = q \begin{bmatrix} \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} & 0 \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} & 0 \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The mathematical equations we discussed in Subsection 3.2.5 regarding the Kalman filter are implemented one on one in the code. We have compared the outcome of each calculation to an existing MatLab implementation to validate our filter. At last we return the new one-dimensional state of each Kalman filter. Those states are assembled to one three dimensional state, which we return to the black box.

3.3.4 Integration methods

We have concluded that the error caused by the integration methods is negligible in our system in Subsection 3.2.3. However, we discuss their implementation because we use them both.

Runge-Kutta integration method

The RK4 method is used for integrating the acceleration when we use the Butterworth low-pass filter. We implemented the method in such a way it integrates a three state vector. We did not modify any of the calculation steps of the original method. However, we changed the input and output from a scalar to a three state vector, which is straightforward.

Euler integration

The Euler integration method is used by the state transition model of the Kalman filter. The Kalman filter is responsible for all the integration, which is directly calculated using matrix multiplication as we have discussed in Subsection 3.2.5. The Euler integration method is embedded in the Kalman filter step at the equation where the predicted state is calculated by:

$$x_{predicted} = Ax_{k-1} = \begin{bmatrix} 1 & \Delta t & \Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{k-1} \\ v_{k-1} \\ a_{k-1} \end{bmatrix} = \begin{bmatrix} p_{k-1} + v_{k-1}\Delta t + a_{k-1}\Delta t^2 \\ v_{k-1} + a_{k-1}\Delta t \\ a_{k-1} \end{bmatrix}$$

3.4 Experiments

Two of the three objectives we have are to reduce the measurement noise and to analyze the drift. In order to reduce the measurement noise we have to determine the value of r and q of the Kalman filter error covariance matrices. We determine r and q by performing a series of experiments using a range of values. Then we analyze the outcome of each experiment to find the best overall set of parameter values in order to use those values for the drift analysis. We also compare a few experiments which are ran using the filter and without, to determine if the filter reduced the measurement noise and improves the outcome.

After the Kalman filter parameters are found we analyze the drift. In this analysis we consider the results over 15 seconds, which is a long time since recent applications perform 3D SLAM at a speed of about 30 Hz to update the position and 1 Hz to update the map [44]. We want to know whether the accuracy of the system is good enough to be within the limits of map based positioning systems. Therefore we consider an error margin of 10 cm and 20 cm as we evaluate the results of the experiments, since those are the accuracies reported for several common used SLAM applications [45].

We perform the experiments during several cases using the experimentation setup as discussed in Subsection 3.4.1. In Subsection 3.4.2 we discuss the cases we use. The first experiment we discuss in Subsection 3.4.3 where we estimate the most optimal parameters for the Kalman filter. In Subsection 3.5.2 we analyze the accuracy of the inertial based positioning system, to gain insight in the integration drift caused by dead reckoning.

3.4.1 Setup

The experimentation setup we use consists of two components. The first component is the winch supported by a plank and wooden blocks as illustrated on the left side of Figure 25. The winch is attached to the second component, which is a sled as illustrated on the right side of the figure. The sled is equipped with the Rift and a laptop to do the measurements using the IMU. We use the winch to pull in sled in a straight line causing the Rift to move in a controlled way. We need to measure the distance travelled using a laser range sensor, since we are unable to pull in the sled at a constant velocity. The laser range sensor measures the distance to the winch which we use to determine the true displacement of the Rift sensor.

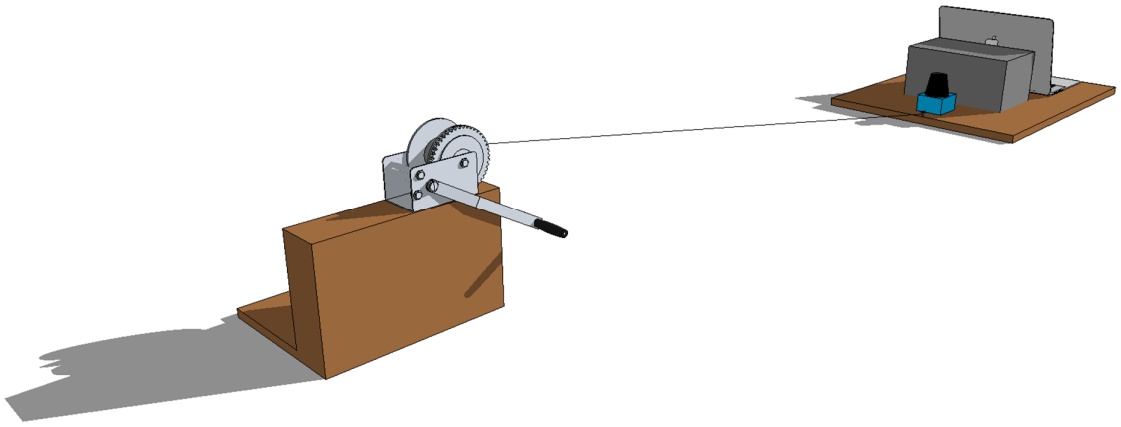


Figure 25. The experimentation setup. The left component with the winch is used to pull the sled forward. The sled is equipped with the Rift, the laser range scanner and the laptop.

3.4.2 Cases

We consider three cases for the experiments. During the first case we do not move the Rift, therefore the displacement is considered drift when noteworthy. For the second case we use the setup to pull in the Rift in such a way the acceleration is applied on one axis only. The third case is similar to the second case, but the Rift is place under an angle of 45° in order to use two axis of the IMU to determine the displacement. The three cases are illustrated in Figure 26.

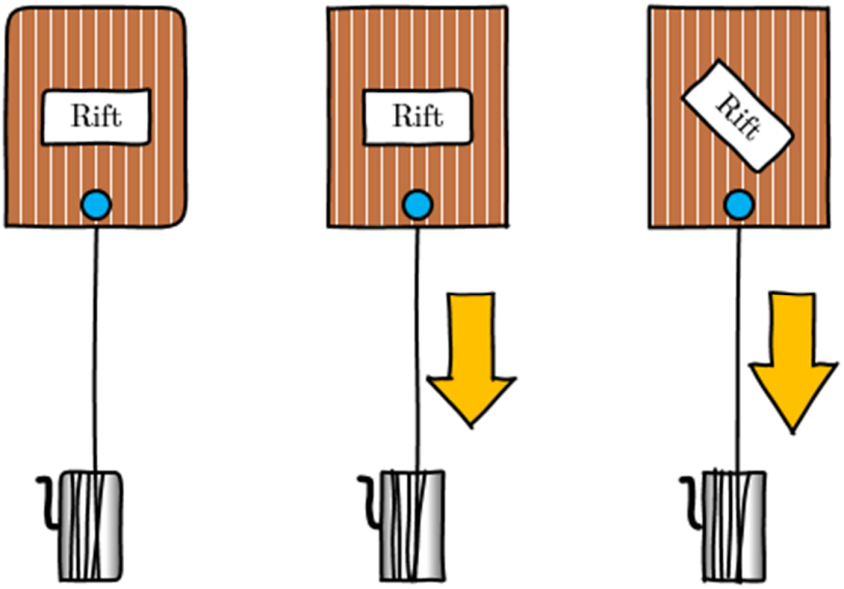


Figure 26. Sketch of the three cases. Left is the first case where the Rift does not move. In the middle is the one axis movement case and at the right the two axis movement case.

Case 1: No movement

During the no movement case we place the Rift on the sled and we let the algorithm run for about a minute. We expect the position to be off several meters after the sequence, due to the measurement errors that causes the drift. This case is used as reference to compare with the other two cases.

Case 2: One axis movement

When moving over one axis we measure how the drift behaves while moving. We expect that the error is worse than during the no movement case. The expectation is that due to the movement the measurement error has a larger effect on the velocity because due to the drift it becomes harder to compensate the estimated movement direction. We use this case to gain insight in the behavior of the drift while moving compared to when not moving.

Case 3: Two axis movement

The MEMS IMU as we explained in Subsection 3.2.2 is a chip, which means the x and y-axis have the same amount of space on the axis, but the z-axis has less space. This might result in other measurement behavior. To be able to gain insight of the effect we put the Rift on the sled in a 45° position on the sled in order to measure the x and z axes. We can then compare the measurements which should be more or less the same. And we can compare the drift to the case with one axis movement, where we expect to have similar behavior.

3.4.3 Kalman filter parameters estimation

To estimate the best settings for the parameters of the error covariance matrices we run the second case five times for a wide range of values for both parameters. For r we consider the range from 1 to 101.00 with steps of 10 and for q we take a range from 0.01 to 1.01 with a step size of 0.01. Then we search the best overall parameters and the individual best scores. The error is measured in meters as distance on the z-axis of the estimated displacement to the displacement measured by the laser range sensor of the first ten seconds. Which means an error of zero is perfect and that the larger the score the worse the estimation is. We consider only the z-axis because the other axis are both zero all the time. Therefore a very large value for r and a very small error for q is always the best since that results in a flat line.

3.4.4 Drift analysis

Using the best overall parameters we analyze the absolute error of the estimated displacement over time. Then we can see how the drift evolves over time for fifteen seconds. We can gain insight about how long the drift is within usable proportions to be able to combine this technique with a mapped based system. Therefore we specifically look for the error exceed these proportions of 10 and 20 centimeter.

3.5 Results

We present the measurement results of the Kalman filter parameters estimation, which we perform to reduce the measurement error in Subsection 3.5.1. Then we present the results of the accuracy experiment in Subsection 3.5.2 in order to analyze the drift over time.

3.5.1 Kalman filter parameter estimation

We performed a series of runs using different values for both r and q in order to determine the parameters for the error covariance matrices of the Kalman filter. We present all the results from the experiments in Appendix I, however, we discuss a part that interests us here. We used the five runs from the one direction case for this experiment. The runs have been executed a number of times and for each the absolute error for the first ten seconds is summed up. This results in a table as shown in Table 1. We see that for the r a range from 1 to 101 is taken with a step size of 10; resulting in all r values ending on a one. The Kalman filter does not accept a value of zero, therefore we started with a one. The numbers in the colored cells represent the summed absolute error in meters for this specific run and the numbers in the uncolored cells on the left represent the q values. If we take a look at the color pattern in the sample, as shown in Table 1, we see a green band which represents the "better" r and q combinations. This band can be observed in every run we did as is shown in the appendix.

For every run we selected the best three results and we marked those with a star, as shown in the table as well. In Table 2 we present an overview of the top three of each run.

q\r	1	11	21	31	41	51	61	71	81	91	101
0.01	163.8029	41.81261	41.36466	54.16035	60.66774	63.99295	65.89876	67.08601	67.87337	68.42133	68.81762
0.02	174.7415	56.01239	36.90023	42.4685	52.12795	58.04443	61.5575	63.79403	65.29833	66.35555	67.12545
0.03	179.72	66.29566	39.30125	37.91481	44.8883	52.6171	57.49632	60.6641	62.82289	64.35405	65.47649
0.04	182.6054	74.52482	43.33695	36.86509	40.8023	47.63728	53.68565	57.68301	60.44065	62.41351	63.86895
0.05	184.4781	81.01523	47.62273	37.30595	38.35579	43.67271	50.09985	54.83887	58.14558	60.53069	62.30094
0.06	185.7905	86.31796	51.56142	38.54819	37.15548	41.16419	46.72974	52.12161	55.93264	58.70295	60.77102
0.07	186.7592	90.86647	55.06534	40.07742	36.86446	39.37438	44.0008	49.52158	53.79676	56.92745	59.27758
0.08	187.5028	94.79576	58.17787	41.92229	36.993	38.04441	42.03204	47.03011	51.73327	55.20155	57.81899
0.09	188.0903	98.27038	61.07532	43.88261	37.44694	37.27365	40.60865	44.88115	49.73805	53.5229	56.39386
0.10	188.5666	101.4128	63.8837	45.88452	38.16096	36.96677	39.38921	43.18758	47.8077	51.88963	55.00119
0.11	188.9592	104.253	66.46145	47.82134	38.91758	36.846	38.41214	41.83775	45.99441	50.29955	53.63964
0.12	189.2894	106.8218	68.89639	49.66948	39.84118	36.92064	37.65521	40.79763	44.45683	48.75066	52.30789
0.13	189.5702	109.1716	71.18182	51.43353	40.8778	37.13689	37.23751	39.85608	43.20611	47.24125	51.00503
0.14	189.8114	111.3253	73.3473	53.07083	41.9292	37.47639	37.01032	39.03737	42.13068	45.84077	49.72971
0.15	190.0216	113.3323	75.35131	54.67072	43.05707	37.91131	36.90929	38.35225	41.26717	44.62479	48.48118
0.16	190.2067	115.2016	77.23428	56.18131	44.16488	38.42996	36.847	37.77372	40.5106	43.5952	47.25824
0.17	190.3703	116.9354	79.0068	57.58696	45.32045	38.90835	36.90915	37.38792	39.80218	42.66837	46.10922
0.18	190.5163	118.5604	80.66093	58.93456	46.46026	39.49438	37.04204	37.15394	39.16912	41.88765	45.08753
0.19	190.6483	120.0845	82.2095	60.26155	47.55798	40.11151	37.21743	37.00663	38.60663	41.22621	44.15788
0.20	190.7675	121.5312	83.68887	61.59956	48.64	40.80087	37.46868	36.92973	38.12834	40.62607	43.36982
0.21	190.8743	122.8983	85.11837	62.89944	49.67904	41.46613	37.76047	36.86378	37.70782	40.04541	42.62715
0.22	190.9733	124.1904	86.4977	64.13933	50.69511	42.17311	38.11111	36.85247	37.4201	39.51497	41.99117
0.23	191.0634	125.4189	87.81563	65.32795	51.68665	42.90189	38.46861	36.90765	37.22471	39.03637	41.43552
0.24	191.1469	126.5844	89.06814	66.49519	52.62769	43.62965	38.80466	36.99423	37.06971	38.59867	40.93538
0.25	191.2242	127.6984	90.26949	67.63035	53.52763	44.34314	39.16327	37.11472	36.98414	38.21861	40.45114
0.26	191.2961	128.7715	91.42184	68.73214	54.45542	45.08972	39.60211	37.26485	36.93113	37.86621	39.9887
0.27	191.3628	129.7953	92.52779	69.8075	55.34309	45.83646	40.03169	37.44915	36.88133	37.58351	39.55812
0.28	191.4252	130.7727	93.59483	70.84695	56.1917	46.56821	40.50788	37.65895	36.843	37.37577	39.16404
0.29	191.4836	131.7127	94.63528	71.86185	57.00487	47.28036	40.98251	37.89363	36.86759	37.22847	38.79143
0.30	191.5383	132.6151	95.64109	72.85661	57.79424	47.97987	41.44592	38.164	36.91009	37.09805	38.45958

Table 1. Sample of the parameter estimation experiment results

Here we see that for run one to four values between 51 and 91 are considered best for the r parameter, but in the last run much lower values are in favor. For the q value we see that a small value below 0.5 is considered best for the first three and the last run. However, the fourth run has much higher q values. This means there is not a set of parameters which works for every run.

Run	Rank	r	q	Error (m)
1	1st best	81	0.29	36.843
	2nd best	51	0.11	36.846
	3rd best	61	0.16	36.847
2	1st best	91	0.06	11.792
	2nd best	81	0.05	11.848
	3rd best	51	0.02	11.874
3	1st best	81	0.37	8.4265
	2nd best	61	0.21	8.4267
	3rd best	71	0.28	8.4277
4	1st best	61	0.94	9.4754
	2nd best	51	0.66	9.4757
	3rd best	61	0.95	9.4763
5	1st best	21	0.48	13.4515
	2nd best	11	0.13	13.4518
	3rd best	21	0.49	13.4526

Table 2. Overview of the top three results of each run

In order to determine a set of parameters we have to combine all the results and find the best overall combination of r and q. The results are combined by normalizing all the errors for each run. This normalization is performed by summing all the error values and then divide each error by the total sum. The normalized values of the five runs are summed per r and q combination, with as result a top three as presented in Table 3. The entire overview of the results is available in Appendix II.

Rank	r	q	Normalized error ($\times 10^{-3}$)
1st best	81	0.52	2.90423
2nd best	71	0.40	2.90424
3rd best	91	0.81	2.90426

Table 3. Overview of the overall top three results

We use the parameter set with r = 81 and q = 0.52. We compare a run of each case with and without using the Kalman filter. The comparison of the first fifteen seconds of case one are depicted in Figure 27. In this case there is no movement, therefore any movement is considered as an error. We see that the unfiltered signal becomes significantly larger than the filtered signal, thus filtering the signal leads to an improvement in this case. In the second case, as illustrated in Figure 28, the filtered signal is performing worse over the time span of fifteen seconds. However, if we compare the first two and half seconds then the filtered signal is better than the unfiltered signal. Then for the last, the two direction movement case, we see that the unfiltered signal is moving in the wrong direction as shown in Figure 29. Therefore the filtered signal is performing better in this case as well.

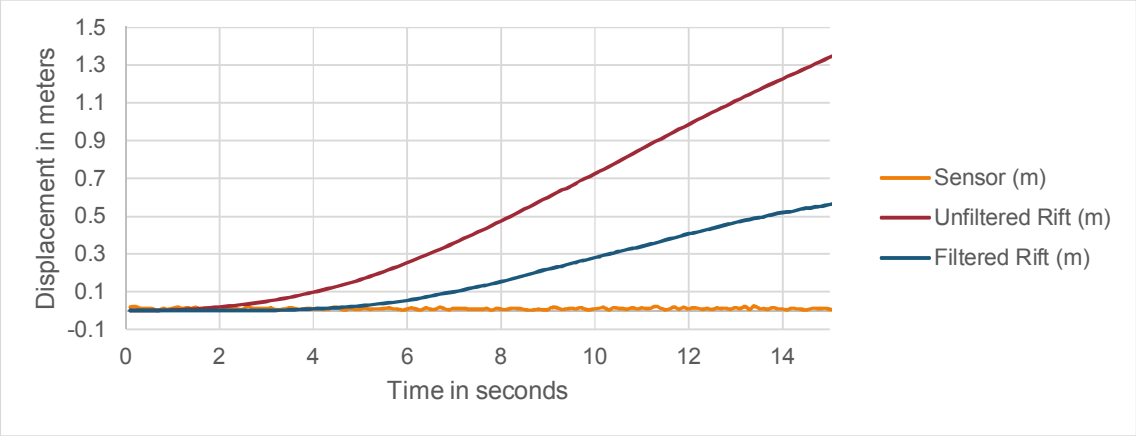


Figure 27. Case 1: No movement - Filtered compared to unfiltered signal

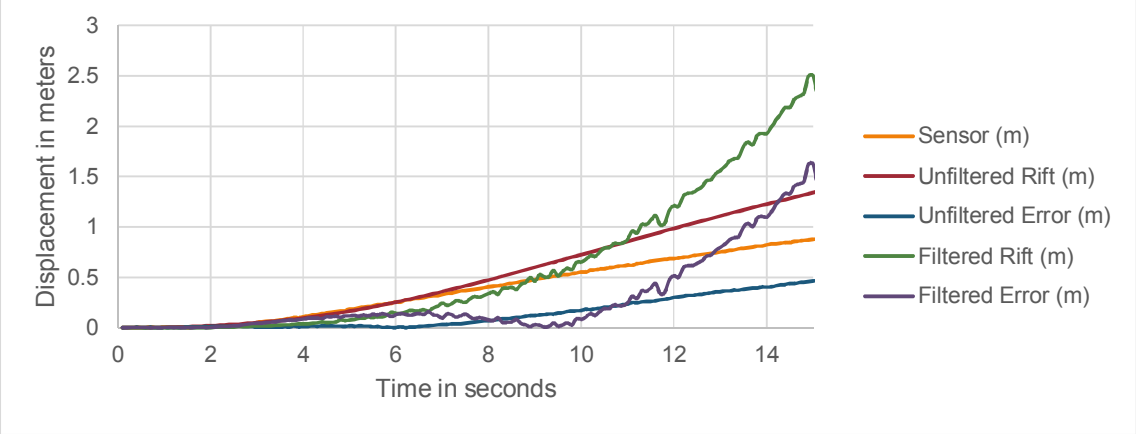


Figure 28. Case 2. One direction movement - Filtered compared to unfiltered signal

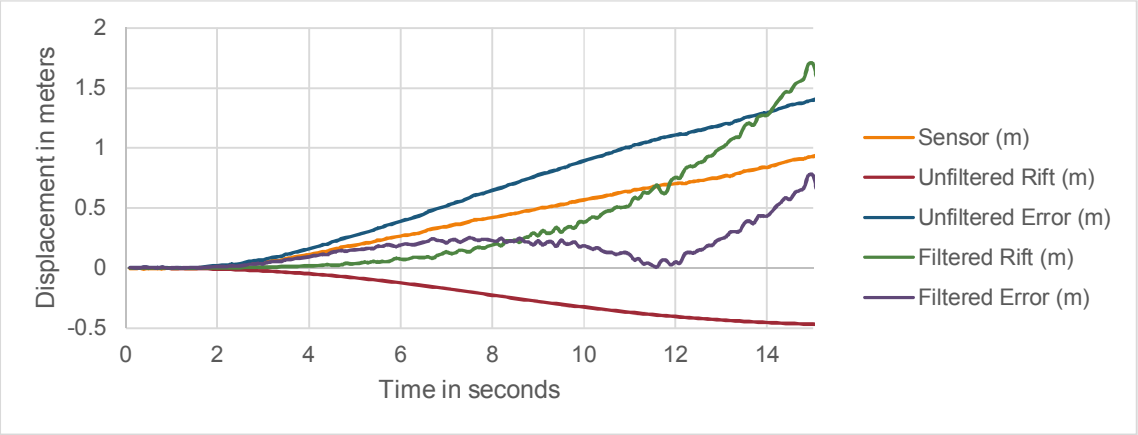


Figure 29. Case 3: Two direction movement - Filtered compared to unfiltered signal

3.5.2 Drift analysis

We analyze the evolution of the drift over time for the three cases as discussed in Subsection 3.4.2. We discuss the performed experiments per case in order to determine if the drift stays within the 10 and 20 centimeter error margins for fifteen seconds.

We present the graphs of the measured displacements on the z-axis of both the estimated displacement by the positioning system and the laser range sensor. We have added the absolute error to the graphs as well, except for the first case because there was no movement and thus is any displacement considered as an error. The data on which the graphs are based are available in the appendices. The data of the first case are found in Appendix III, for the second case the data is included in Appendix IV and the data of the third case are available in the Appendix V.

Case 1: No movement

In the first case where we do not move the estimated displacement by the system is almost the same for all five runs, as depicted in Figure 30. When we zoom in on the estimated displacement we see that the deviation between the runs is present, however, it is very small as shown in Figure 31. The first thing we see in the data is that after 3 seconds the

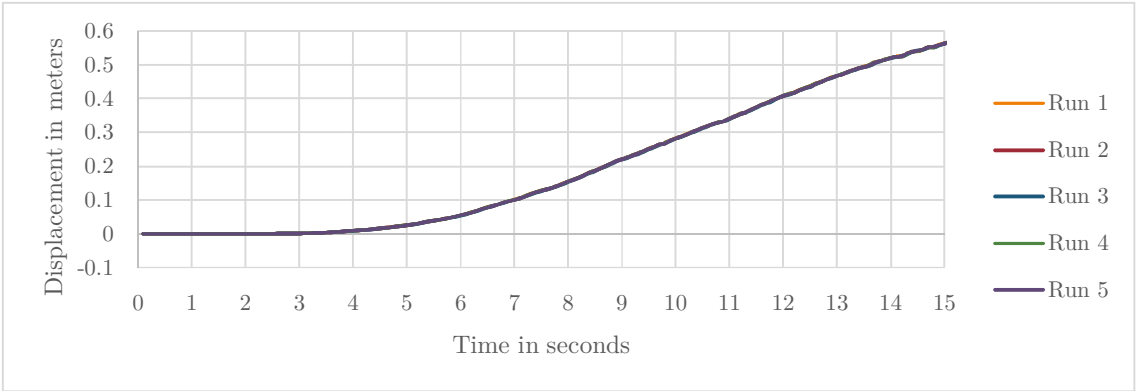


Figure 30. Case 1 – Combined overview of five runs

displacement becomes larger than 1 mm. We observe that after 7 seconds the error exceeds 10 cm and that after 8.5 seconds the error exceeds 20 cm. We do not have a proven explanation for the effect that the deviation between the runs is very small occur. However, as we manipulate the signal using the Kalman filter it is most like that this effect is enforced by the filter. Then the course of the signal is enforced by the prediction step of the Kalman filter since the process error is considered small and the measurement error is considered large. Therefore it seems that due to the high value of r the evolution of the process error forces all

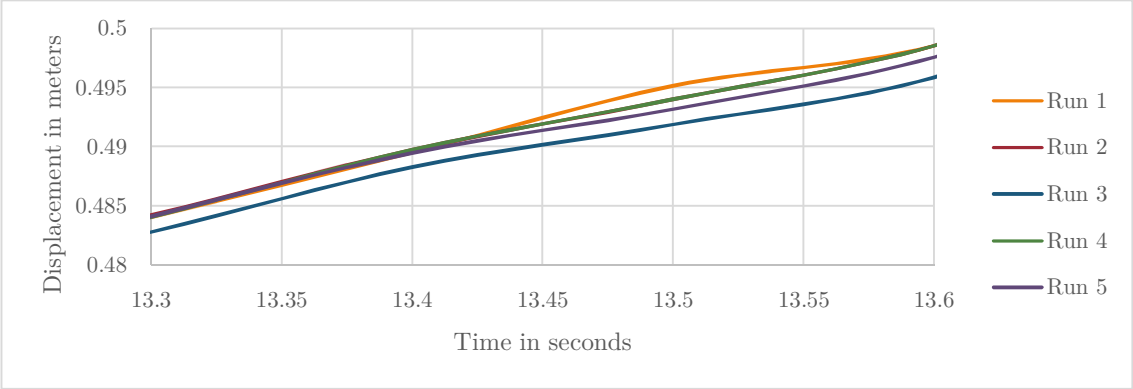


Figure 31. Case 1 – Zoom on the combined overview of five runs

the runs to be almost the same. Further research is required to find out if this hypothesis is true.

Case 2: One direction movement

We present the runs of the one direction movement case in this subsection. The first run differs from the other four runs, therefore we discuss this run individually. The four remaining runs are considered all together since they are very similar to each other.

In Figure 32 we see the first run displayed. We observe that the estimated displacement named Rift cannot follow the sudden displacement at 2 seconds, which is due to the high value of the measurement error covariance matrix. This high value results in that the Kalman filter ignores large parts of the measurements. We observe that after 1.7 seconds the error exceeds 10 cm and after 2.0 seconds the error exceeds 20 cm. It is clear that the estimated displacement

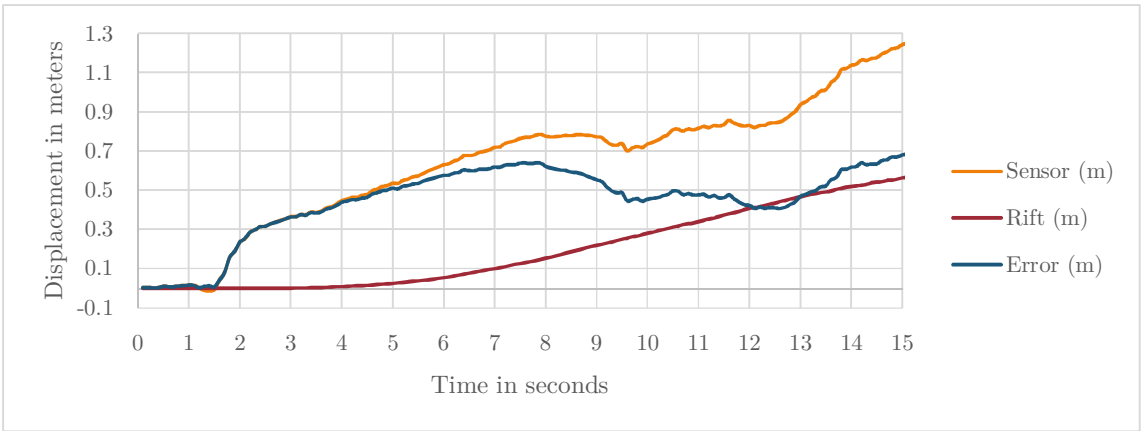


Figure 32. Case 2 – Run 1

is not able to follow the sudden movements during the run. Because when the displacement moves back a little after 9 seconds then no effect is observed in the estimation.

We present the results of the second to fifth runs combined. First we consider the displacement as measured by the laser range sensor in Figure 34. We observe that the runs are similar, however, we see that the second run starts to move earlier than the other runs. We can also see that the second and third run overlap after five seconds. We expect the estimations to be similar to each other, in such a way that the drift over time is the same for all runs.

The displacement estimated by the system is presented in Figure 33. We observe that the second run drifts faster due to the earlier movement as reported by the sensor. The third to fifth run show what we did expect; they look very similar. We see that the sensor reports a displacement of less than 1.1 meters while the estimated displacements is at least the double after 15 seconds. If we consider the error we see that all runs are within the same margin the first five seconds, as illustrated in Figure 35. The second run exceeds the error margin of 10 cm after 5 seconds and after 5.6 seconds it exceeds the 20 cm margin, which is faster than the other three runs. However, the error margin of the other runs exceeds 10 cm after 5.4 seconds which is quite close to the second run. This in contrast with exceeding the 20 cm margin which happens after 10.7 seconds for the third run and after 11.1 seconds for the fourth and fifth run.

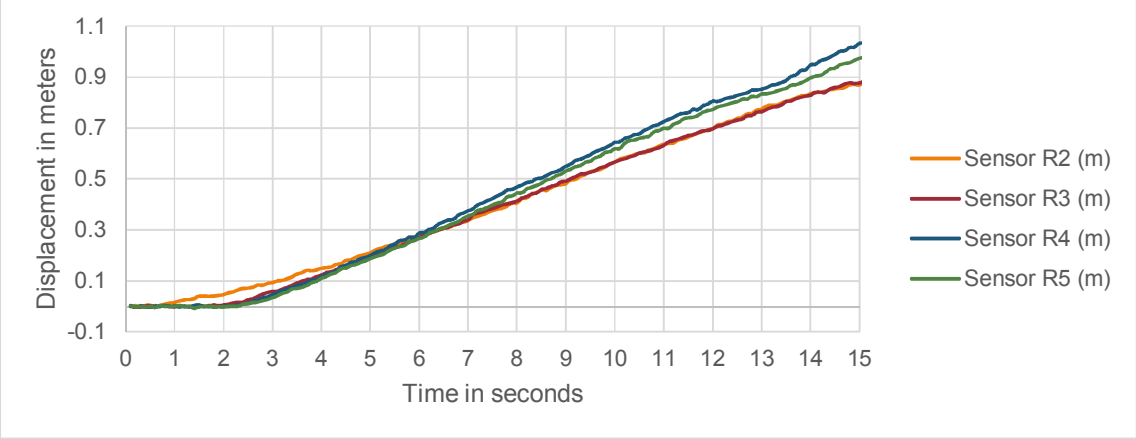


Figure 34. Case 2 - Run 2 to 5 – Measured displacement by the sensor

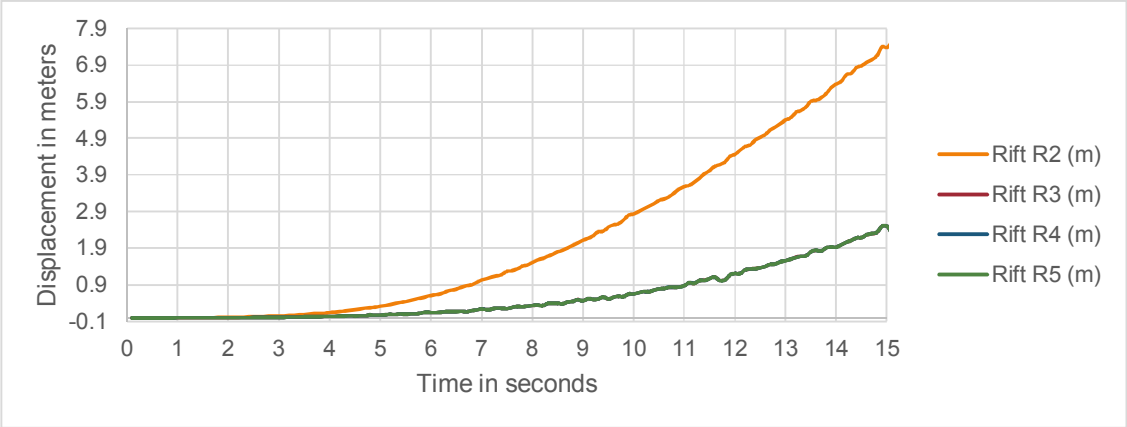


Figure 33. Case 2 - Run 2 to 5 - Estimated displacement by the system

We observe that the error grows and shrinks during the first ten seconds of the third to fifth run. This is due to the estimation which cannot keep up with the change in the signal, but after some time the drift grows larger causing the estimation to intersect with the sensor signal. The intersection takes place at approximately 10 sec, which is most likely due to the Kalman filter parameter estimation. In this estimation we chose to pick the values that resulted in the smallest error over a time period of 10 seconds.

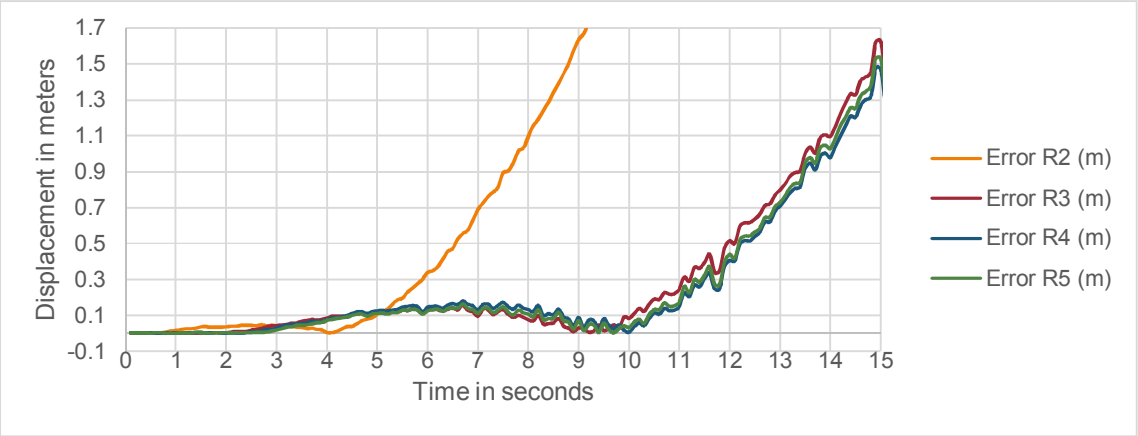


Figure 35. Case 2 - Run 2 to 5 - Error caused by drift in the system

Case 3: Two direction movement

The results of the two direction movement case are presented in this subsection. The goal of this experiment is to determine if using two axis of the IMU results into different results then when using one axis as in the second case. We consider all five runs together, like the second to fifth run of the second case. The measured displacement as retrieved by the laser range sensor are shown in Figure 36. We see that all runs are similar, however, the fifth run seems to have more hick-ups in the displacement than the other runs and it moves a bit faster as well. We expect to see a larger error for the fifth run due to the hick-ups.

The estimated displacements by the system are illustrated in Figure 37. We see that the first to fourth run are similar to each other, but in contrast with the second case here the displacement after 15 seconds is smaller. The displacement for the fifth run approaches the measured displacement quite well after fifteen seconds, however, we observe the hick-up errors during the run as we expected. If we consider the errors as depicted in Figure 38 then we observe that the hick-ups cause the fifth run to have the worst error for the first thirteen seconds, but the error is significantly smaller after fourteen seconds compared to the other runs. For the first to fourth run we see that the intersection between the estimated and the measured displacement happens at approximately 11 seconds. This indicates that there might be a significant difference between the axes of the accelerometer, however more extensive research is required to prove this. However, if the axes differ from each other it might be resolved by using three IMUs to determine the acceleration. If we consider the error margins then we see that the first run exceeds the 10 cm margin after 4.9 seconds and the 20 cm margin

is exceeded after 12.5 seconds, which is the best result of all five runs. The time after which the margins are exceeded is more varied for this case than for the previous case, therefore we present an overview of when the margins are reached for each run in Table 4. We see that the fifth run is the worst run, which is as we expected as it is the result of the hick-ups while performing the experiments.

We observe that the results of the second case are better than the results of the third case. This is probably due to the hick-ups which are more present in the third case. The hick-ups might be the result of combining the two axes to retrieve an acceleration signal. Such a hick-up can be the result of the axes to have the same error and the next moment have an opposing error.

Run	Exceeded 10 cm margin	Exceeded 20 cm margin
1	4.9 seconds	12.5 seconds
2	3.9 seconds	6.4 seconds
3	4.1 seconds	6.1 seconds
4	4.4 seconds	7.1 seconds
5	3.7 seconds	5.0 seconds

Table 4. Case 3 - Overview of the times after which the error margins are exceeded

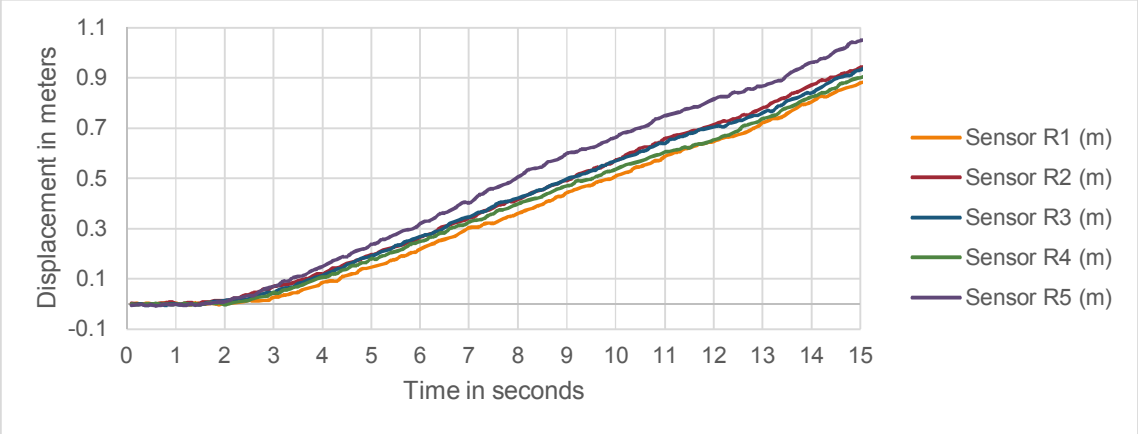


Figure 36. Case 3 – Measured displacement by the sensor for all runs

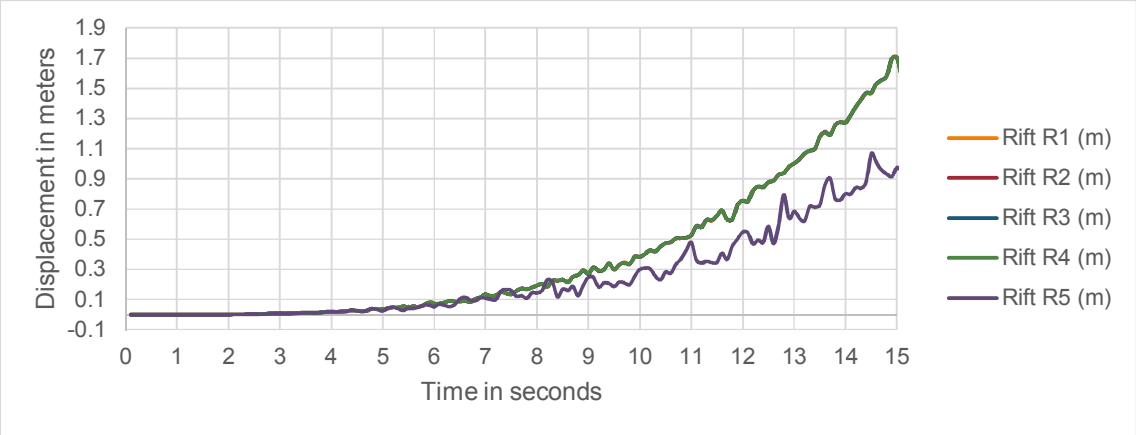


Figure 37. Case 3 - Estimated displacement by the system for all runs

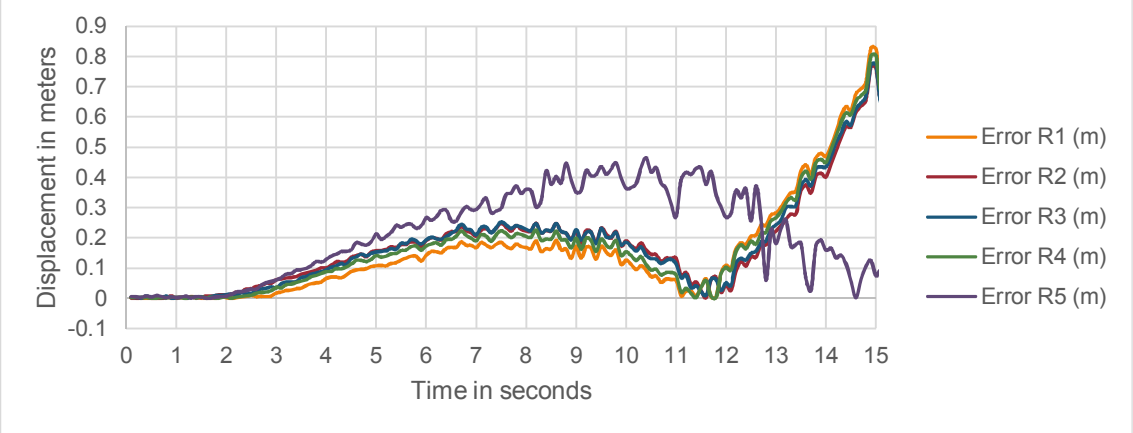


Figure 38. Case 3 - Error in the estimated displacement for all runs

3.6 Conclusions

In this subsection we will discuss the results and conclude this section. We consider the results of the Kalman filter parameter estimation first, then we discuss the results from the drift analysis. At last we consider whether the objectives are reached and what can be done to improve the system further.

3.6.1 Kalman filter parameter estimation

We have observed that the estimated displacement by the system reacts very slowly on changes in the accuracy and velocity. This is most likely the result of the high value for the measurement covariance matrix parameter r. We also see that essential data is ignored by the Kalman filter, therefore we can conclude that the parameter set is not chosen well. We have seen that there is a range of good values, therefore it would be wiser to consider smaller values for both parameters. In case where we want to use this inertia based system in combination

with a map based system then we have to know the time step between the updates of the map based system. If for example this is 1 second then we should consider the best results after 1 second instead of 10 seconds. It might be helpful to extend the error formula in such a way that the unfiltered estimation is considered as well. This would resolve the problem in which the filter causes the estimation to be worse than the unfiltered estimations.

3.6.2 Drift analysis

From the drift Kalman filter parameter estimation results we have learned that even while the second and third case are fairly similar that there is no specific set of parameters which works best for all runs. When we analyzed the drift we have observed that the system is not able to keep up with changes in the signal. However, due to the small velocity we still manage to keep the error within the margins of 10 cm and 20 cm for several seconds. The worst case, if we do not consider the first run of the second case, is the fifth run of the third case where we manage to keep the error within a margin of 10 cm for 3.7 seconds. This would be good enough if we combine it with a map based system, however, if we consider the first case of the second run we see that due to the larger velocity the error exceeds the margins as soon as the displacement starts at about 1.7 seconds. This means that for a more fundamental solution just filtering the signal is not enough as every situation requires another set of parameters for the Kalman filter.

3.6.3 Objectives

The objectives of the experiments were to reduce the measurement noise and to analyze the drift. We can state that we reduced the noise, however, not in an effective way since we consider parts of the signal as noise as well. This is the result of the high value for the measurement covariance matrix of the Kalman filter. We should perform another experiment as explained earlier this subsection to estimate better results, but this might still not lead to a result in which all situations benefit by being filtered. An approach we would do in future research is keeping the filter using much smaller values for the Kalman filter parameters, so a little noise is allowed. Then we would combine it with computer vision to reduce the drift and make the measurement noise insignificant. Choosing computer vision techniques is a logic choice since an augmented reality system as we consider it contains a stereo camera system already.

The second objective we consider in this section is analyzing the drift. Due to the chosen Kalman filter parameters we have seen that for the cases we considered that the drift stays within the error margins of a map based system for several seconds. However, the situation is that when a human moves around then the dispositioning is rarely linear. Therefore we will consider reducing the drift using computer vision as we discuss in the next sections.

4 Monocular vision

We use computer vision in order to reduce the drift in a more fundamental way than reducing the noise of the measurements performed by the IMU. Computer vision is a logic approach since in an augmented reality system, as we consider it, is equipped with a stereo camera system. Stereoscopic vision is superior compared to monocular vision if we consider the accuracy and amount of data which can be obtained. However, monocular vision in the way we propose it requires less computational power than stereoscopic vision.

We use a single camera to retrieve images of the environment in front of the positioning system. These images are used to retrieve an optical flow in order to determine a direction in which we move. We choose this technique because other techniques such as scene reconstruction are more efficient when a stereo camera system is used. Using monocular vision to detect objects as landmarks in the environment in order to determine the position of the user relative to the landmarks is computational expensive. Detecting objects is a challenge on its own and the calculations required to determine the locations of the landmarks is considered as a SLAM algorithm. An application in which recognizable points in the environment are used to assemble a map is called MonoSLAM [46]. The optical flow technique we use is used for optimizing a visual SLAM system which is localizing a vehicle in a suburban environment [47]. We use three much more simple and intuitive methods in order to assist the positioning system.

In this section we discuss computer vision and optical flow in the preliminaries first. Then we discuss the methods we use in the research section. The implementation of these methods is discussed in Subsection 4.3. We present and discuss the experiments we perform in order to see if we can reduce the drift in Subsection 4.4. The results of the experiments are presented in Subsection 4.5 and discussed in the following subsection.

4.1 Preliminaries

4.1.1 Computer vision

We use cameras to analyze and process images of the real world. There are several tasks where computer vision is often used, for example motion analysis, scene reconstruction and recognition tasks. Recognition tasks can involve finding objects, facial recognition and content-based image retrieval. All those are techniques applied in our daily life. Scene reconstruction, where a 3D object or environment is reconstructed using vision, is an extension to recognition. Using multiple images we are able to reconstruct the real world environment by recognizing the same things in the images.

Motion analysis is the part on which we focus with monocular vision. Motion analysis is less complex than scene reconstruction and less expensive in terms of computation costs. Nevertheless, scene reconstruction might contain more valuable information for the purpose of drift reduction. In this thesis we focus on a method called optical flow.

4.1.2 Optical flow

An optical flow algorithm determines how preselected points move on the image plane. We use two images to determine the movement. There is an efficient implementation available which is performing this task to retrieve the optical flow from sequential images; therefore we are interested in the outcome only. The output is the optical flow, which is the movement of the camera relative to the environment and objects registered by the camera in an image plane. If we inverse the movement then we have the movement of the camera. We can only determine whether there is movement in horizontal or vertical direction directly, since we only use one camera. Assuming we only move the camera and the environment is stationary we can extract more information about the direction we move in.

We can analyze the optical flow and retrieve several properties. We could extract the movement direction using the focus of expansion (FOE) as depicted in Figure 39 [48]. The FOE is the point from which all points expand if moving in a forward direction. At the left side of the figure we see a forward movement the FOE is in the middle with a positive expansion, walking backward would have a negative expansion. At the right side of the figure we see diagonal movement leads (i.e. forward and sideways) to a FOE moved horizontally in the image plane.

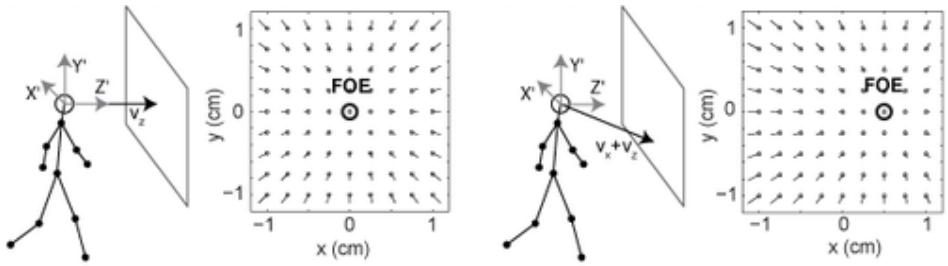


Figure 39. Optical flow used to determine the focus of expansion (FOE) which is used to determine the movement direction in 3D

4.2 Research

We use optical flow to improve the inertia-based positioning system. In this section we discuss three methods. In Subsection 4.2.1 we discuss the no movement method, where we recalibrate the system when no movement is detected by the optical flow algorithm. We discuss the second method in Subsection 4.2.2. The second method uses the no movement principle but on each axis independently, in order to compensate errors in the acceleration. This error in the acceleration is the result of the error in the orientation of the gravity. The third method is a more complex approach in which we analyze the focus of expansion to estimate a movement direction in three dimensions as explained in Subsection 4.2.3.

4.2.1 No movement method

The drift is always present even when we are not moving. To deal with scenarios in which there is no movement, for example when we are looking at something, we can detect there is no optical flow. When the average flow is nearly zero we set the acceleration, velocity and

displacement to zero. We treat all measurements as result of the gravity and therefore we compensate this gravity with the measurements.

However, when there is movement in the environment this method is not useful. Therefore filter the optical flow. We calculate the average movement of the flow. Then the length of every flow line is compared to the average length times a manually controllable factor, if the length of the flow line is larger then it is ignored. The result of this filtering is that small movements on the foreground are ignored. An example in which there is movement is depicted in Figure 40, here we see the lines move horizontally as the camera moves from right to left.

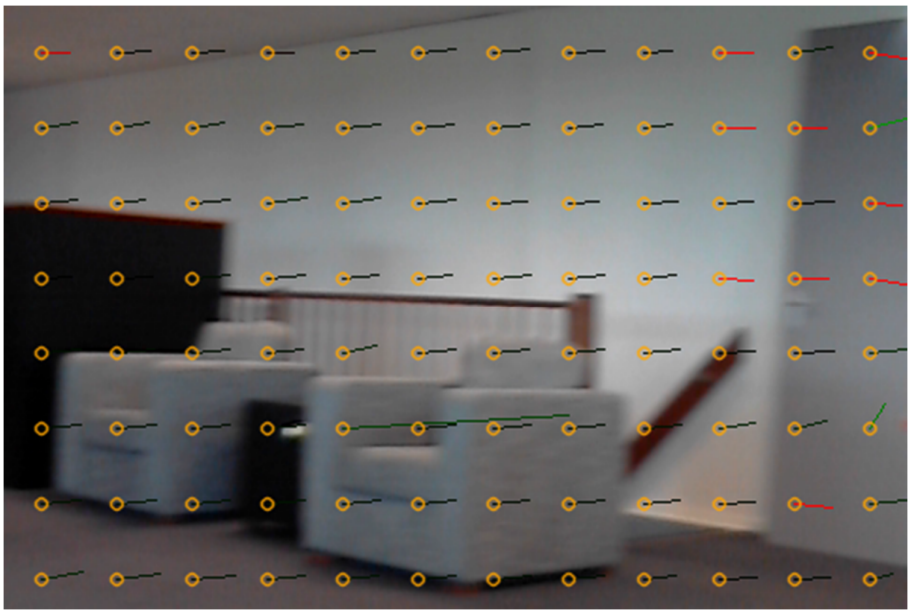


Figure 40. Screenshot of optical flow as obtained by the algorithm

4.2.2 Correct gravity method

When we have a scenario in which we walk in a straight line then the no movement method does not do anything. Therefore we analyze the x- and y-axis movement separately. When we have a no movement state in an axis then the gravity, acceleration, velocity and displacement is fixed. We do this the same way as with the no movement method, including the filtering.

Besides the separation of the axis we reduce the impact false measured acceleration and reset the drift when we correct the gravity. This correction is performed by applying the next formula:

$$\mathbf{g_c} = \mathbf{g_u} - \mathbf{c},$$

where $\mathbf{g_c}$ is the corrected gravity, $\mathbf{g_u}$ is the current gravity and $\mathbf{c}$ the correction. The correction is the average measured acceleration on the not moving axis over the last four samples in which there was no movement detected. The correction of the gravity has a delay of four image samples, which is about half a second. Using more frames would lead to an unresponsive system and thus the method would have no contribution to the signal.

4.2.3 Direction estimation method

The disadvantage of the correct gravity method is that we cannot correct false accelerations in the z direction. To be able to correct for the z direction we need an estimation of the movement direction in three dimensions.
We could try to determine the focus of expansion. The points we track are distributed in a grid formation allowing search to the focus of expansion. With the FOE known we can extract the direction. We can cast a line from the origin of the camera to the FOE on the image plane. The line is the distance between camera and the FOE, in order to retrieve the direction this distance is normalized. We can use the estimated direction to guide the acceleration and velocity in the appropriate direction. However, the vision module is too slow to do adjust the acceleration like that. Therefore we have to stick with correcting the gravity and then a much less complex method does suffice. In essence we can add an extra axis to the gravity correction method for we consider the z-axis.

We take the source image points from the optical flow and divide the image in four quarters. We do the same for the second image points. Then we analyze the flow direction in every quarter. Based on those four directions we can determine the direction we move in. The direction is limited to combinations of movement on the axes that can be combined by using values of -1, 0 or 1 for each axis.

4.3 Implementation

First we consider the libraries we used for computer vision in this thesis in Subsection 4.3.1. In Subsection 4.3.2 we discuss how the monocular vision module is integrated in the positioning system and how the module itself is implemented. We discuss the implementation of each method in Subsection 4.3.3.

4.3.1 Programming environment

There is a huge and widely known open source computer vision library available called OpenCV. We use the functions of this library, but the library is written in C and C++. Therefore we need a wrapper to be able to call the functions, which is provided by OpenCVSharp. OpenCVSharp is a wrapper that is actively maintained and the largest of the .NET wrappers available. The wrapper does not include the newest GPU module available in OpenCV yet, thus we cannot use this module. Even if the GPU module would be available it requires a NVIDIA GPU with CUDA support, which we do not have.

4.3.2 System

The system is expanded with the additional monocular vision module, which we named the MonoVision module. We implement the MonoVision module ourselves so it contains all three monocular vision methods. The module runs at a speed lower than 60 Hz, which is significantly slower than the IMU module. To synchronize the modules we let the last update from the slower module affect the faster module updates until there is a new update available from the slower module.

Overview

The module provides the sensor value, which is the optical flow data. The methods to improve the system are integrated in the black box. The methods modify the measured state, but the system reacts too slowly in cases where the estimated state is far off. Examples of such cases is when the drift is large and when there is no movement. To deal with this slow reaction we reinitialize the Kalman filter and provide the appropriate initial state. Reinitializing the filter has a benefit; the potential errors are reset which results to better estimations by the filter.

Like the IMU module the MonoVision module runs in a separate thread and we fire an event every loop to report to the black box. The event contains the retrieved optical flow data, leading to the overview as illustrated in Figure 41.

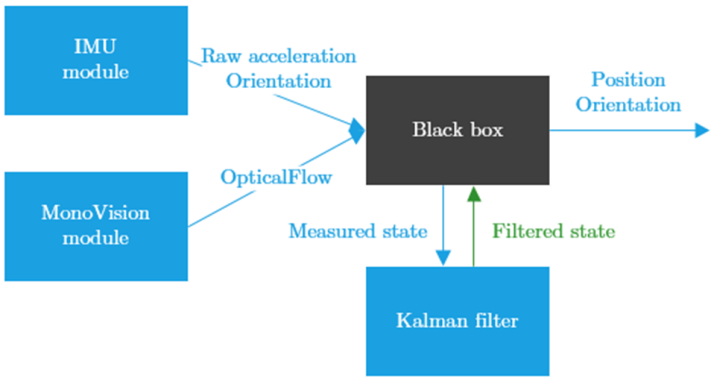


Figure 41. Inertial positioning system overview including the MonoVision module. The additional MonoVision module provides the optical flow which is used by the black box to improve the measured state.

Module

The module depends on the OpenCVSharp library. The overview as depicted in Figure 42 contains four parts that are handled by library. First we initialize the module during the initialization of the positioning system. The initialization of the module consists of the initialization of the camera instance. This initialization is provided by the library.

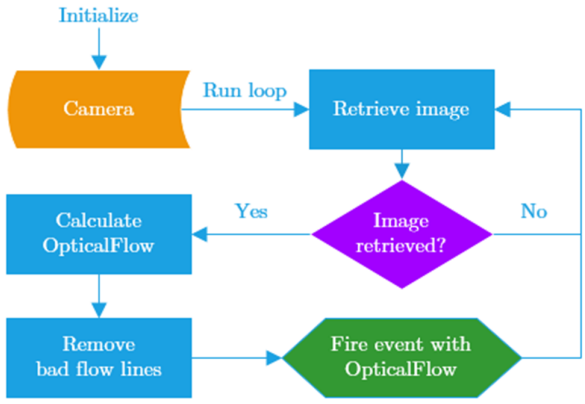


Figure 42. MonoVision module overview. Each loop a camera frame is retrieved and the optical flow relative to the previous frame is calculated.

When the loop starts we retrieve a video frame, which is an image. If a frame is retrieved successfully then we use a function from the library to calculate the optical flow. We remove the bad flow lines by calculating the average flow line length and eliminating flow lines with a length larger than the average length times a configurable factor.

4.3.3 Methods

No movement method

The optical flow direction is provided to the positioning system by the module. If there is no direction detected in the optical flow then we correct the measurement. The average measurement of four sequentially no direction frames is added to the gravity to compensate for the measurement error. We do not reinitialize the Kalman filter in this implementation because we cannot foresee the effect of such an intervention and it is beyond the scope of this thesis to research the effect.

Correct gravity method

The correct gravity method works exactly like the no movement method except that the flow direction does not have to be zero in this method to work. In this method the measurement is corrected on each axis individually. In a case where we are moving horizontally and not vertically then we can correct the measurement in vertical direction. If the measurement moves in the opposite direction with respect to the optical flow direction then the measurement is not corrected, since the user can make sudden movements.

Direction estimation method

We divide the optical flow in quarters based on the position of the point in the source image. Then we determine the two dimensional movement in each quarter as shown in Figure 43. Based on those four direction we determine whether there is movement in forward or backward direction or not. If we detect that there is movement then we determine the direction of the movement by the combinations of each quarter. For instance if the we have all quarters point to the middle then the direction is backward but if three quarters point to one quarter and that one quarter does not have a direction then the direction is in that one quarter.

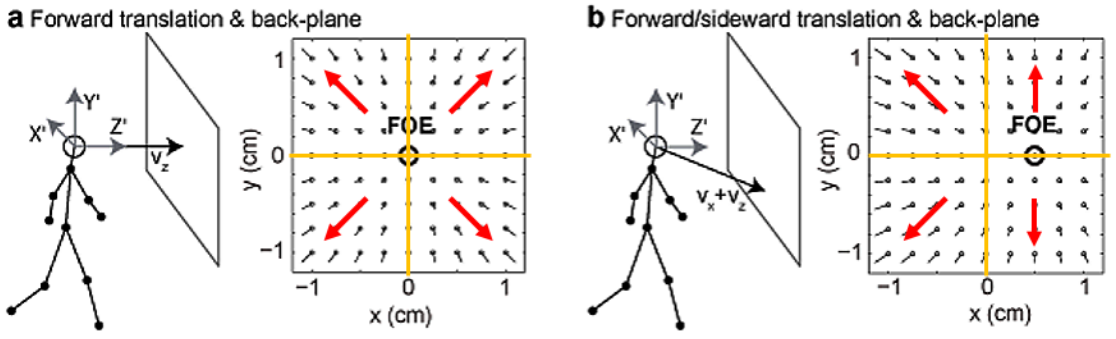


Figure 43. Schematic illustration of the implementation of the direction estimation method

4.4 Experiments

We want to see whether the methods improve the accuracy of the cases as described in Subsection 3.4.2. Therefore we use the same recordings as used in Subsection 3.5.2, but due to the similarities in runs we only use one run per case. To be specific we use the first run of the first case and the third run of the second and third run. Those three runs are ran for the three methods. The results of the tests will be compared to the results presented at Subsection 3.5.2. Notice that we do not estimate the parameters for the Kalman filter again. However, better results might be achieved when we use a specific set of parameters for each method. We do not consider a different set of parameters due to the lack of time.

4.5 Results

We present the results per case. Each case is executed three times, once for each method. The first case we discuss is the no movement case. The outcome of all three methods look exactly the same in a full scale overview. This overview is shown in Figure 44. We see that the measurement error of the sensor is worse than the estimation of the Rift, since we know that the displacement is zero. However, if we zoom in on the three methods then we will see there is a difference in the error of the displacement as depicted in Figure 45. Note that any displacement is considered as an error for this case.

As we analyze the differences in error we notice that after 15 seconds the errors have evolved in an almost linear way. We can also see that the error decreases as the method gets more advanced. The error of the correct gravity method is about sixteen times as large as the error of the direction estimation method after one minute. The error of the no movement method is on its turn 1.77 times larger than the error of the correct gravity method.

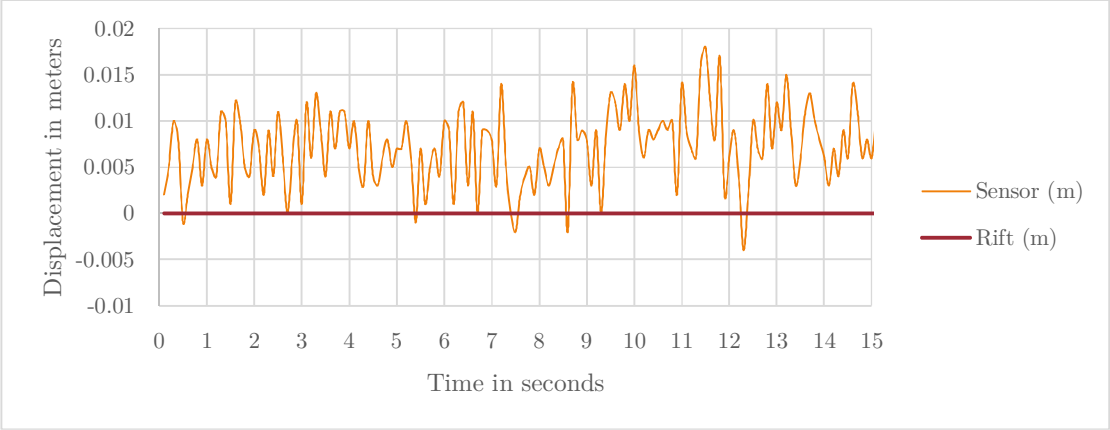


Figure 44. Case 1 – MonoVision runs

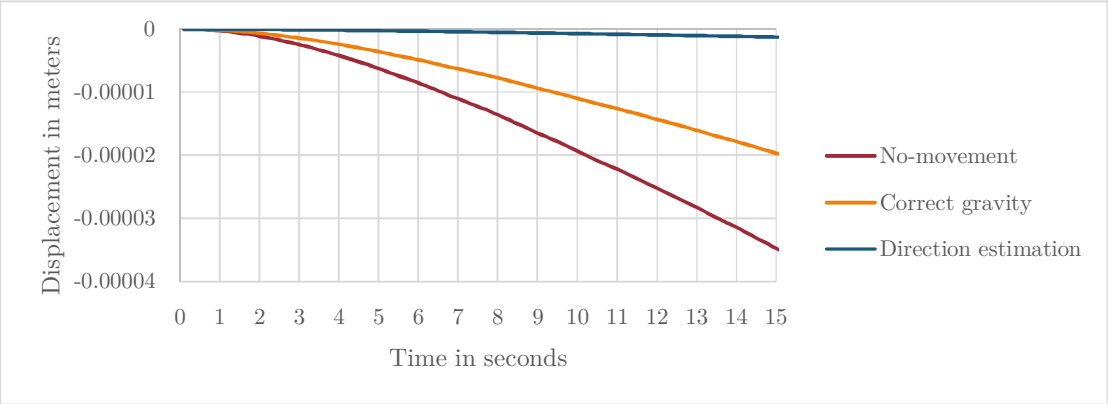


Figure 45. Zoom in on the displacements as estimated by the positioning system using the three different monocular vision methods.

The second case is moving in forward direction. We expect that the no movement method and the gravity method do not change anything compared to results as presented in Subsection 3.5.2. This expectation is based on the fact that movement is detected in both horizontal and vertical direction in the optical flow which results in that the measurements are fully considered.

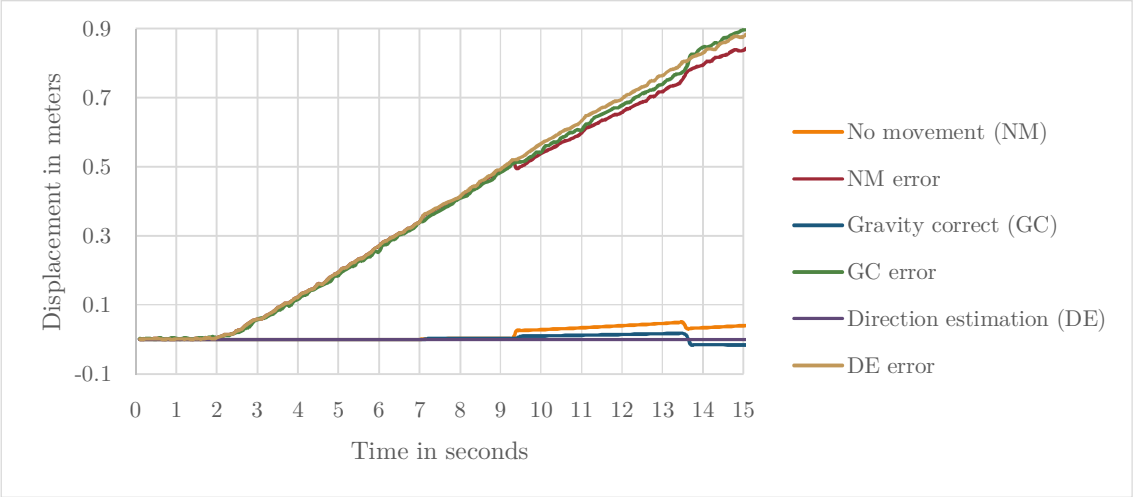


Figure 46. Displacement by monocular vision methods for the one direction case.

As we can see in Figure 46 the displacements advance totally different than we expected. After analyzing the video data it became clear that there is no movement detected except for a few frames. Which means we move too slow to be able to detect movement. This results in that almost all data is ignored for the no movement and gravity correct method. For the direction estimation all data are ignored.

The third case where two directions of the sensor are used shows similar results as the second case. The movement is too small to be able to detect a direction or any kind of motion. The plots of this case are depicted in Figure 47.

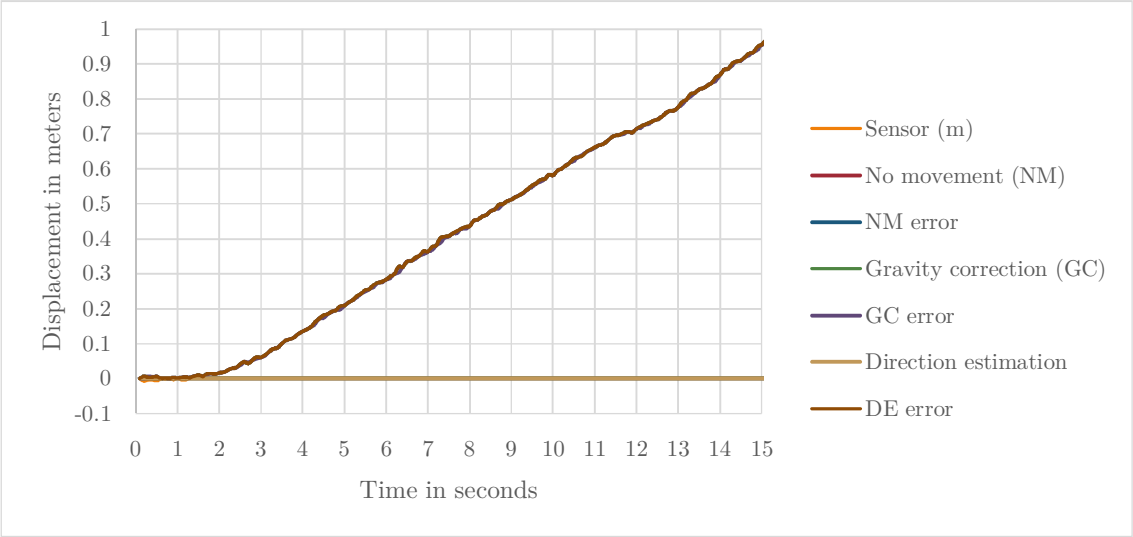


Figure 47. Displacement by monocular vision methods for the two direction case

4.6 Conclusions

Based on the results as presented in Subsection 4.5 we can conclude that in this setting, without modifying the Kalman filter parameters and slow movement that monocular vision does not improve the measurements at all. However, in cases where the velocity is higher the optical flow might work. We do not consider higher velocities or other Kalman filter parameters because the time limit does not allow us to research this any further.

In order to be able to reduce the drift we can use stereoscopic vision, so we can see depth as well. This gives us more information about the environment at the cost of computational power. The stereoscopic vision method we implemented is discussed in the next section.

5 Stereoscopic vision

In this section we discuss how stereoscopic vision can be used to retrieve depth and how this information can be used for positioning purposes. Stereoscopic vision is performed by analyzing the images of two parallel cameras. The depth can be retrieved from the images using several approaches. The approaches are divided into local and global methods, where the feature matching method we use is considered a local method [49]. We choose this feature matching method since it is a very intuitive approach to look at a certain point and estimate the depth of that point. In this research stereoscopic vision can be used in order to guide and correct estimated displacements and velocities by analyzing the displacement and determining a velocity based on the displacement. We can construct and analyze a point cloud based on the matched features in the images in order to retrieve the visual displacement. The methods how we can construct a point cloud are presented in this section, however, the stereo camera system we built is not aligned in a proper way resulting in unreliable depth estimations. Therefore we use the point clouds to estimate the direction in which we move.

5.1 Preliminaries

5.1.1 Stereo camera system

The most important aspect in stereoscopic vision is the stereo camera system. The system contains two cameras which should be exactly parallel in the view direction. This parallelism we is used to calibrate the stereo camera system. The images are used to create a three dimensional representation of the world. A photograph of the stereo camera system we created is shown in Figure 48. The bracket of steel is place over the Rift, with as result in that the two cameras are mounted in front of the Rift.

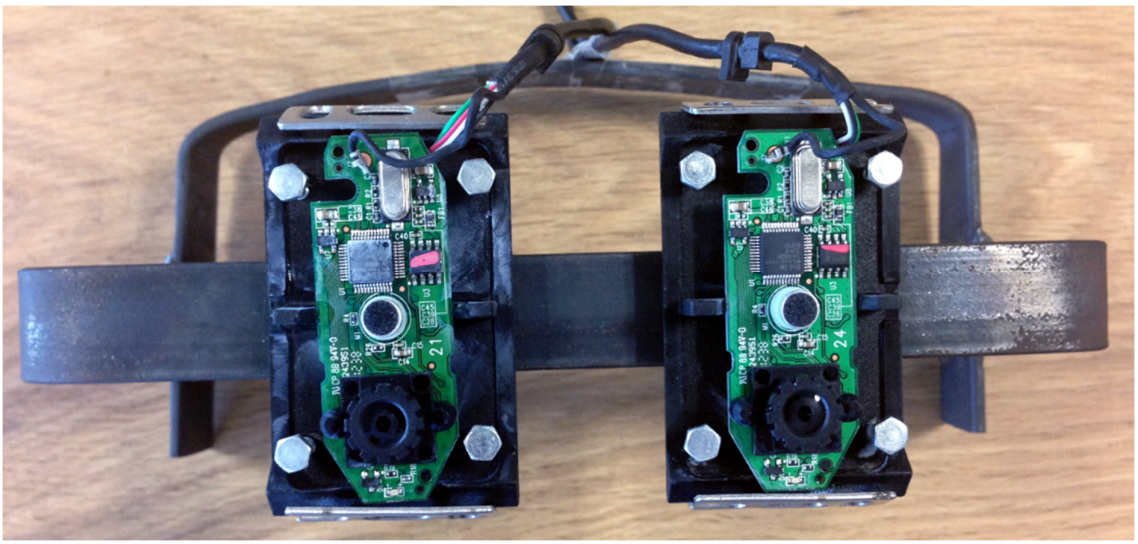
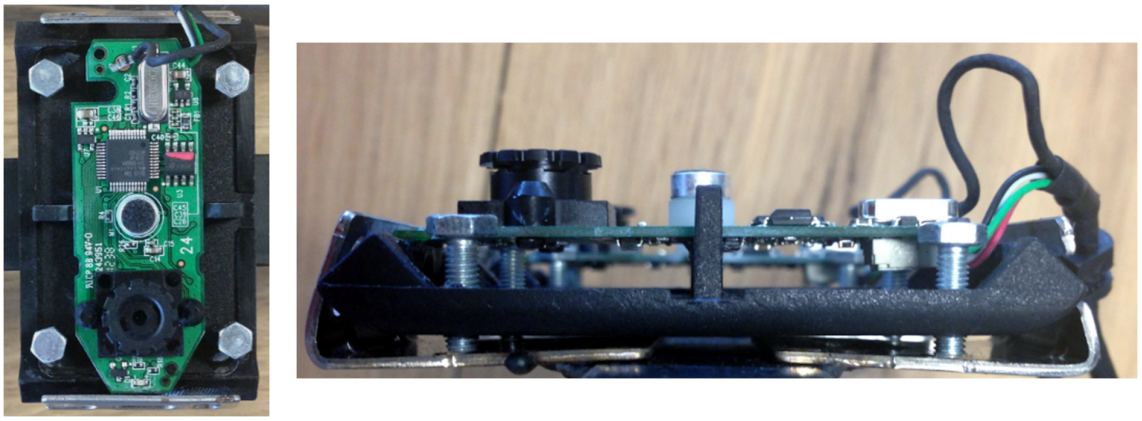


Figure 48. The stereo camera system. The bracket is placed over the Rift placing the cameras in front of the Rift. Two cameras are attached to the bracket.

The cameras are stripped down Logitech C310 webcams. We glued the cameras onto plastic panels and we tapped screw-threads in the corners of the panels. We use the bolts that fit the threaded holes to mount and align the cameras to the shackles, as depicted in the Figure 49.

To create the representation of the real world we need to find corresponding points in the stereo images. From there we are able to determine where the points are in the real world relative to the stereo camera system. Computer vision is used to detect those points called features.



(a) (b)

Figure 49. The camera attachment. (a) The camera is glued to a small panel, equipped with a bolt in every corner. (b) The panel is placed in the shackle and the bolts are used to mount and align the camera.

5.1.2 Features

Features are points in an image which are recognizable. The features in an image are selected by feature detectors. Over the years several feature detectors are developed, each with specific advantages and disadvantages. Examples of characteristics of the features detected by the detector are corners, edges or blobs. We use a feature descriptor to extract the descriptors of a feature. Those descriptors define certain characteristics of a feature. The characteristics which are considered depends on the used descriptor.

We match the detected features with a feature matcher, which calculates the distance between two features. This distance is not the distance in pixels on the image plane, but the distance in the descriptors. If a descriptor is found in both images then there is a match.

5.2 Research

The features in the stereo images are used to we construct a point cloud as we explain in Subsection 5.2.1. To be able to determine the velocity of the stereo camera system, we need

to match point clouds. In Subsection 5.2.2 we look at how the velocity can be estimated after the point clouds are matched.

5.2.1 Point cloud retrieval

We build a point cloud in two steps. First we extract the features from the stereo images. Then we estimate the position of the feature relative to the camera in three dimensions.

Stereo feature extraction

A feature detector finds recognizable points in both stereo images as depicted in Figure 50. Then we use a feature descriptor which extracts feature-specific information. The descriptors are shown by the size of the circles in the figure, and the orientation of the feature is illustrated by the line which is drawn from the center of the circle. We compare the descriptors of each feature of both images with each other with a feature matcher. This feature matcher calculates the distance between the descriptors, which is similar to histogram matching. In the figure the matched features are marked by a green color and a line is drawn between the matched features. The red features are unmatched features, these features are detected in only one of the two stereo images.

We know the location of each three dimensional point on the image plane of the left and right image. Because of the parallelism of the cameras we can assume the features are always on the same horizontal line and the difference in the vertical axis is zero. For every feature we check the position on the left and right image planes, if the difference in vertical axis is larger than zero we mark the feature as false positive match. The false positive matches are deleted, resulting in a more robust feature set.

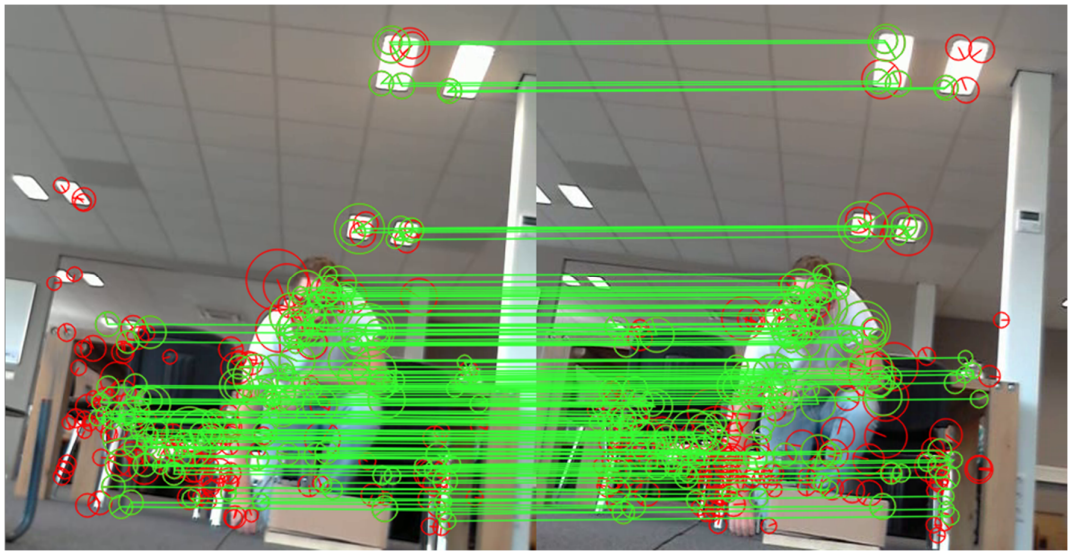


Figure 50. Stereo feature extraction. The red circles are extracted features in an image which cannot be matched to a feature of the other image. The green features with a line are the features which are matched.

Position retrieval

We need to estimate the location of each point in three dimensions, which is possible with several methods. First we discuss an intuitive method where we cast lines to find the intersection of the line, which is the location of the point in three dimensions. Then we discuss a method in which we use the focal length to calculate the position of the feature. And at last we use prior knowledge about the stereo camera system to estimate the position of the points. For all methods prior knowledge is required about the cameras and the stereo camera system. Therefore we need to calibrate the cameras and the stereo camera system. The details are explained in Subsection 5.3.2. For now we assume to have perfect calibration parameters.

Reprojection method

We know the position of the features on the image planes and we know the distance between the cameras. Besides that we also know the center of the image and the focal length of the camera. This knowledge is used to calculate the image plane corners in the camera coordinate system. We calculate the three dimensional position of the upper left image plane corner defined as **ipc** by:

$$\mathbf{ipc} = \begin{bmatrix} -\mathbf{c_x} \\ \mathbf{c_y} \\ \mathbf{f_x} + \mathbf{f_y}/2 \end{bmatrix},$$

where $\mathbf{c}$ is the image center and $\mathbf{f}$ is the focal length. These mathematics are based on the pinhole camera model. Next we add the position of the feature to the image plane corner. Note that the origin of the matrix is at the upper left corner, but the origin of the image is at the bottom left corner. Therefore we add the position of the feature using the next equation:

$$\mathbf{pos3D} = \mathbf{ipc} + \begin{bmatrix} \mathbf{pos2D_x} \\ -\mathbf{pos2D_y} \\ 0 \end{bmatrix},$$

where **pos2D** is the position of the feature on the image plane. From here we need to convert the three dimensional position in the camera coordinate system to the global coordinate system, which we call reprojection. We use the inverse transformation matrix to do the reprojection. The inversed transformation matrix $\mathbf{T}'$ is calculated by the following formula:

$$\mathbf{T}' = [\mathbf{tR}]',$$

where $\mathbf{t}$ is the translation vector from the camera calibration and $\mathbf{R}$ is the rotation matrix extracted from the stereo calibration. Then to do the reprojection we need to retrieve the homogeneous form of **pos3D**, which achieved by adding a fourth element to the vector with value 1. This homogeneous vector is multiplied by the inverse transformation matrix to retrieve the global position $\mathbf{p}$ of the feature by:

$$\mathbf{p} = \mathbf{T}' \begin{bmatrix} \mathbf{pos3D_x} \\ \mathbf{pos3D_y} \\ \mathbf{pos3D_z} \\ 1 \end{bmatrix}$$

The depth of this point is not correct, therefore we use cast a line from the cameras through the corresponding points. We can perform an intersection on those lines in the xz-plane, which is possible because the points always have the same value on the vertical axis. This intersection point is the lines is the estimated position of the point in the real world.

Triangulation method

The depth can be calculated directly from the two points on the image plane as well. When we lay the images on top of each other we will see the points have different values on the x-axis. The difference between those x-axis values is called disparity. From the calibration we know the focal length and the baseline, which is the distance between the cameras. We can calculate the depth by triangulation, this is performed by the next calculations:

$$\mathrm{d} = \mathbf{left_x} - \mathbf{right_x}$$

$$\mathrm{z} = \mathrm{f} * \mathrm{b}/\mathrm{d}$$

$$\mathrm{x} = \mathbf{left_x} * \mathrm{z}/\mathrm{f}$$

$$\mathrm{y} = \mathbf{left_y} * \mathrm{z}/\mathrm{f}$$

$$\mathbf{p} = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{bmatrix},$$

where d is the disparity, b is the baseline and f is the focal length. This method is less computationally expensive than the reprojection method.

Disparity-to-depth mapping method

The calibration retrieves a disparity-to-depth mapping matrix of the stereo camera system we refer to as **M**. The matrix allows us to do the reprojection method more efficient. Instead of several matrix calculations to find the point per camera in the global frame we can do this with two calculations. Those calculations are:

$$\begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ \mathrm{W} \end{bmatrix} = \mathbf{M} * \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{d} \\ 1 \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} \mathrm{X}/\mathrm{W} \\ \mathrm{Y}/\mathrm{W} \\ \mathrm{Z}/\mathrm{W} \end{bmatrix},$$

where x and y are the coordinates of the feature on the image plane. We reproject both the left and right point and cast lines like with the reprojection method. The point for the point cloud is calculated by intersecting the lines, where the intersection is the position of the point.

5.2.2 Velocity estimation

We match the point clouds by the same principle as we use for matching features. We obtain the descriptors of each feature while the features are matched. Then we use a feature matcher to find corresponding points in both point clouds. To calculate the velocity of the points we need to estimate the average displacement of all the points and divide it by the difference in time. The average velocity $\mathbf{v}$ is calculated by:

$$\mathbf{v} = \frac{\mathbf{s}}{\Delta t} = \begin{bmatrix} \Delta x/\Delta t \\ \Delta y/\Delta t \\ \Delta z/\Delta t \end{bmatrix},$$

where $\mathbf{s}$ is the average displacement and Δt is the difference in time. In order to calculate the displacement we need to correct for any change in orientation first. We retrieve the orientation of the Rift from the last IMU sample. Then we reverse the effect of the orientation of every point in the cloud by:

$$\mathbf{p_n} = -\mathbf{O}\mathbf{p_o},$$

where $\mathbf{p_o}$ is the oriented point, $\mathbf{p_n}$ is the not oriented point and $\mathbf{O}$ is the orientation from the IMU sample, which is the orientation of the Rift relative to the global frame. We calculate the displacement of each matched point in the point cloud. Then we take an average of all points, which leaves us with the direction we move into. We divide this direction with the elapsed time to calculate the velocity of the point cloud relative to the camera. To retrieve the velocity of the camera we reverse the velocity of the point cloud.

5.3 Implementation

5.3.1 System

We expand the system with another additional vision module. We named the module the StereoVision module. The module runs at a speed lower than 10 Hz at all times, which is significantly slower than the IMU module. The slower module affects the faster module to synchronize them.

Overview

Like the previous two modules we attach the StereoVision module to the black box as well. The StereoVision module provides the point cloud calculated by firing an event to the black box. The black box then calculates the direction we move in. In Figure 51 we illustrate the modified system which includes the StereoVision module.

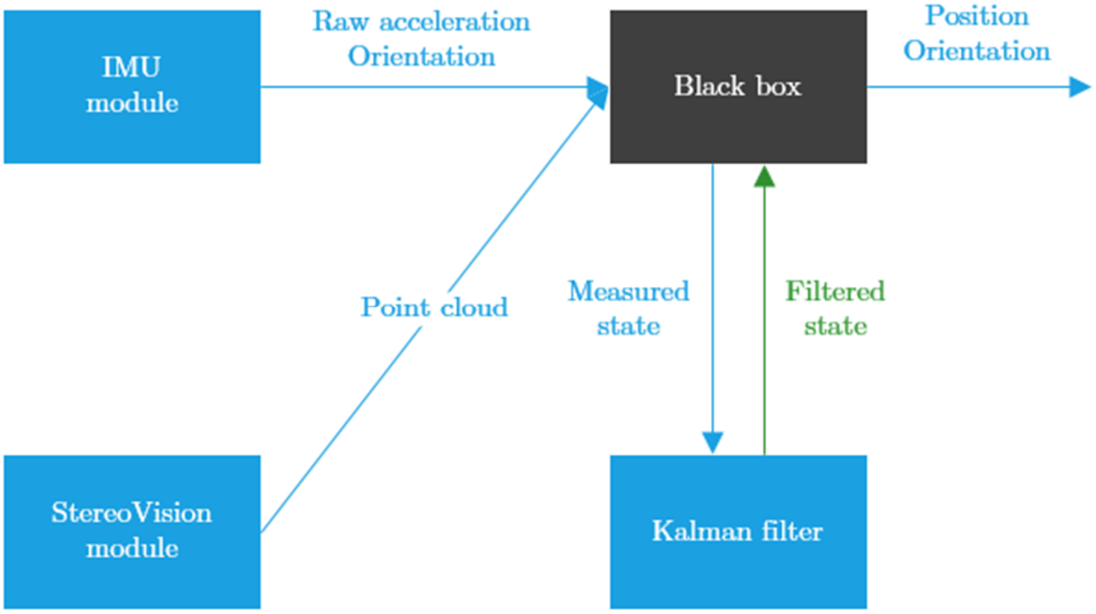


Figure 51. System overview with the additional StereoVision module. The module provides the newly calculated point cloud to the black box. The black box calculates the movement direction by comparing the two point clouds.

Module

The StereoVision module depends on the OpenCVSharp library like the MonoVision module. The module uses several functions from the MonoVision module as well, e.g. the initialization of the camera, the record and playback methods. As illustrated in Figure 52 the module looks similar to the MonoVision module as well. First both cameras are initialized, then the calibration file is retrieved. We discuss the calibration in detail in Subsection 5.3.2. At this point we enter the loop that retrieves an image from both cameras.

When both images are retrieved we use a feature detector as provided by OpenCV to search for corresponding points. We use the SURF feature detector in this implementation, since this detector in combination with the FREAK descriptor gives us decent results within a relative short time. We tried the FAST feature detector which is incredibly fast, with detection times of less than 3 ms. However, we had too many mismatches leading to a very instable point cloud. We also tried the ORB feature detector, which shows similar results as the SURF detector but the ORB feature detector is slower. We do not consider further comparison between the feature detectors and descriptors.

Based on the physical limitation that corresponding points are always on the same height in both images, we eliminate all matches that do not apply to this constraint. This elimination of matches result in a set that contains a minor amount of mismatches. Less mismatches lead to a more stable point cloud. From this point we use the reprojection method to calculate the 3D positions of the matched points. An event that contains the point cloud is fired when the construction of the point cloud is complete.

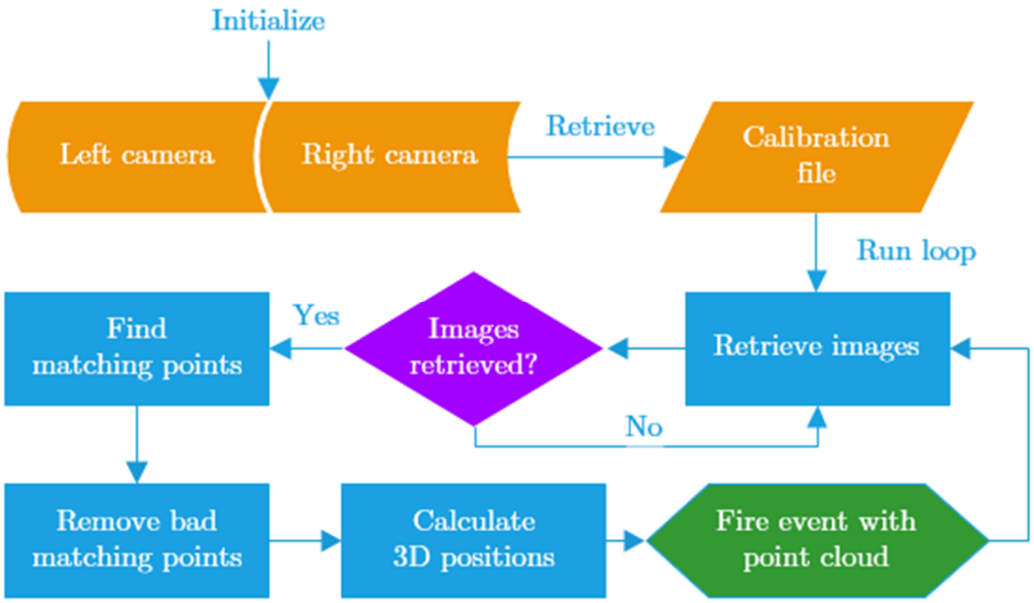


Figure 52 The StereoVision module overview. Each loop retrieves images and finds the corresponding features in the images. Based on the physical constrains mismatches are removed. Then using the calibration file the 3D positions of the points are calculated.

5.3.2 Calibration

The calibration of the cameras consists of two parts. First we calibrate the each camera independently to retrieve the camera parameters. A known pattern is presented in front of the camera to execute the calibration routine. In our case the known pattern is a chessboard. The corners within the board are recognized. We measured the size of the squares and delivered that as input to the calibration algorithm. Once the calibration is complete for each camera we still need to know how the cameras are place relative to each other.

The calibration of the stereo camera uses the chessboard pattern which is shown in front of the cameras as well. Based on the disparity of the corners of each camera the algorithm calculates the required parameters. Those parameters we use for the depth calculations which are performed at the point cloud retrieval.

5.3.3 Method

The velocity estimation method is divided in three parts. The first part is the point cloud retrieval, where we use the camera parameters to estimate the depth of matched points. Here we will see the depth estimation is too unreliable to estimate a velocity from. Then we discuss how point clouds are compared. At last we show how we implement the direction estimation.

Point cloud retrieval

As we have seen in Figure 52, the point cloud retrieval is performed by the stereo vision module. We divide this process three parts. First we find features in the images with the SURF

feature detector. Then we describe the features with the FREAK descriptor. We use a brute force matcher to compare every feature from the left image against every feature in right image and visa versa. This is called cross checking the matches, it delivers more reliable matches and thus a more robust point cloud. We also eliminate all matches which do not have the same height in the image, because those are mismatches according to the assumption that the y-value of the position of the feature is in both images are the same.

Then the reprojection method is used to calculate the depth, as we discussed in Subsection 5.2.1. Unfortunately we created our own stereo camera system and therefore we cannot place the cameras in an exactly parallel configuration. Therefore all methods for depth estimation are by definition wrong, since this parallelism is required. The reason why we cannot place the cameras in parallel alignment is because the focal length of the cameras is 4.4 mm. Such small focal length means that we can have a large error in the alignment even with the slightest affine rotation between the cameras.

Point cloud matching

To be able to determine the displacement of the point cloud we need to match each point of the current point cloud with the points of previous point cloud. We achieve the point cloud matching in the same way as we match the points in the images. Every point in a point cloud contains the descriptors of the matched features. We match the descriptors with the same matcher as we used for the features. We do not consider other methods to match the point clouds since the depth estimation is not good. Therefore this method suffices while there might be better methods in terms of match accuracy.

Direction estimation

We can calculate the displacement of the point clouds after they are matched. However, we need to compensate for the displacement caused by rotation. To compensate for rotation we transform the point cloud to the last local frame. Then we calculate a weighted average of all points in the point cloud. The weight of a point in the point cloud is determined by the disparity in the image. We took the disparity in the image as weight because the displacement of closer points are more reliable in accuracy than points with a larger disparity.

5.4 Experiments

We implemented the point cloud method to see whether it could improve the accuracy of the positioning system. However, we have seen with the monocular vision methods that the movement speed was too low to detect any movement. With this method we can only process about two or three frames per second which might result in changes that are large enough.

The original idea was to estimate a velocity based on the displacement of the point cloud, but as it turned out that our custom stereo system is not constructed well enough. The alignment of the cameras is not near perfect which causes large errors in the depth estimation. Therefore we have to stick with an estimation of the direction. We expect similar results as we expected in Subsection 4.4. This time the processing speed is much lower, so it is more

likely the stereoscopic experiments gives us a slightly better result than the monocular experiments.

5.5 Results

We presents the results for each case individually. The first case is the no movement case. As we do not move we expect that the point cloud does not move. However, if we look at the plot as shown in Figure 53 then we see movement is detected and thus we have a similar outcome compared to the first case in Subsection 3.5.2.

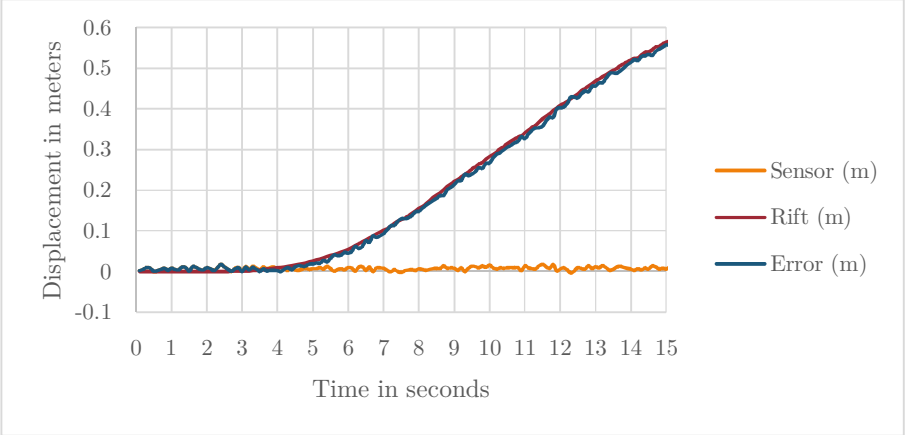


Figure 53. Displacement by the point cloud method for the no movement case

In Figure 54 we see the displacement for the second case. Here we see that movement is detected, therefore all measurements are allowed. Even in the part where the displacement is equal for a second the measurements are considered, instead we would like them to be ignored. Now the run looks very similar to the second case in Subsection 3.5.2.

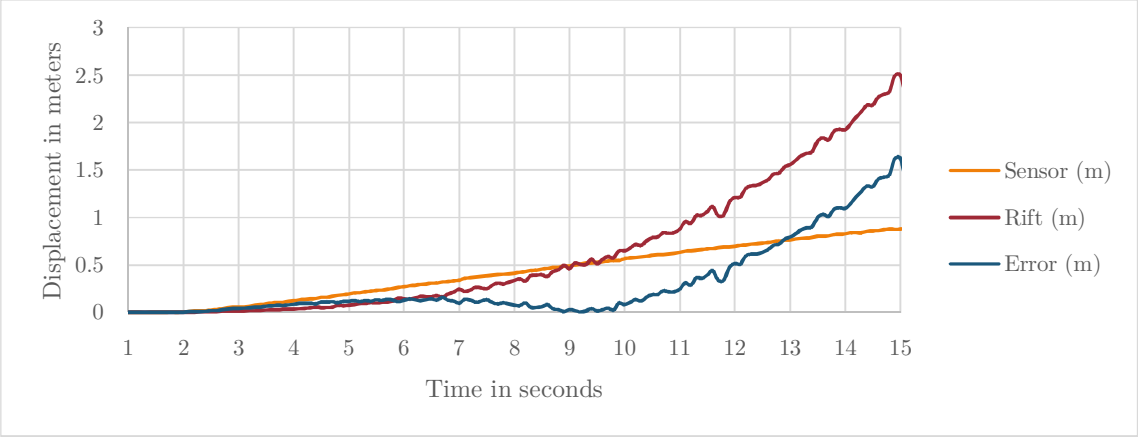


Figure 54 Displacement by the point cloud method for the second case

In Figure 55 we see the displacement for the third run. Here we witness a totally different outcome than we expected. The estimation of the system moves backward instead for forward. The result can be explained after analyzing the video and the point cloud movement. Due to the sideways movement in a 45 degree clockwise rotated orientation the features seem to move away from the cameras. This means the system registers the movement of the point cloud, but that the analysis is wrong.

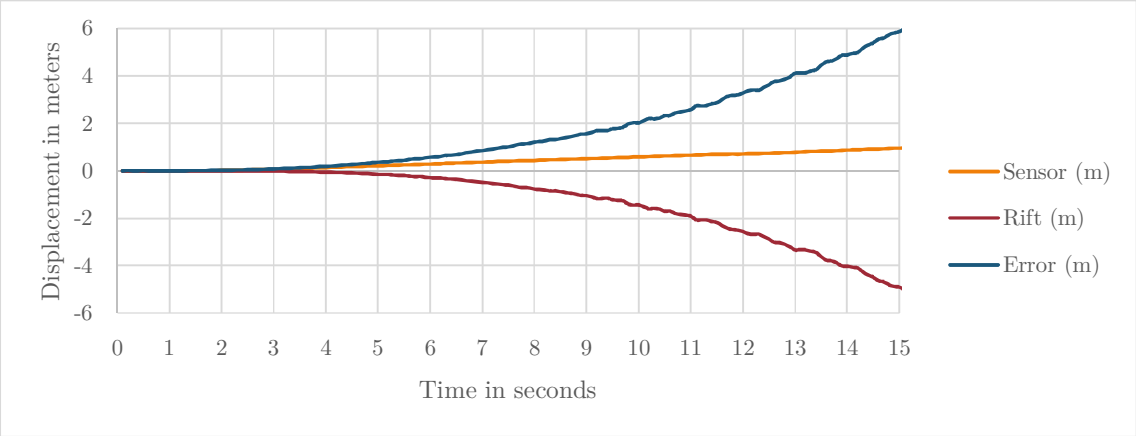


Figure 55 Displacement by the point cloud method for the third case

5.6 Conclusions

We can conclude that in this setting the stereoscopic vision method does not contribute to the system in a good way. It either performs the same as the default setup or it performs worse. In this case not only the movement speed but also the processing speed and the stereo camera system contributes to the failure of this method. We did not have time to further improve the processing speed nor the stereo camera system. Further research is needed in order to determine whether stereoscopic vision using a point cloud build from features might for a positioning system.

6 Discussion

In this section we discuss any thoughts on how to improve the inertia based system further. We also discuss the Kalman filter parameters and what would be advisable to do when we want to use the system for short term periods. Thoughts on how to resolve the issues with the monocular vision approach are discussed. Finally, we will discuss how the velocity estimation of the stereoscopic vision can be realized with better hardware.

In Section 3 we have discussed and determined the parameters for the Kalman filter. Those values are considered for the experiments we did. The outcome results in a rather high parameter values. We can also see that in most runs the measured and the estimated position intersect at about ten seconds. This is most likely due to the fact that we determined the values based on the first ten seconds of each run. We have seen in Figure 23 that higher values for the measurement parameter r results in a less adaptive estimation. Therefore it is advisable to experiment with lower values for both parameters when we want to use the inertia based system for short term measurements of less than 2 seconds.

An application where the inertia based system would operate on short terms could be a combination of the system with another positioning system. For instance we can use a map based positioning system which updates at a rate of 1 to 5 Hz and add an inertia based system to deal with all the movements in between. In this way the drift would be canceled every update of the other system. This might result in a system that is capable to give reliable positioning information at a rate of 500 Hz. There have been studies whom focused their research on combining inertia based systems with other systems.

The drift and measurement errors can be reduced when better equipment is used. In this thesis a MEMS IMU was the main sensor. These are not suited to do navigation tasks, however there are several grades in the quality of sensors. When a practical solution is considered it is advisable to use either a tactical or a navigation grade IMU. It might also be a solution to use multiple IMUs and combine their measurements to get a more reliable measurement.

We tried to improve the measurements with computer vision techniques. We use monocular vision to detect an optical flow within an image. However, in our case the movement of the system in the experiments is too slow. This might be solved when we use cameras with higher resolutions. The optical flow precision would increase almost seven times when we would use 1080p cameras instead of the VGA cameras. This means we can move much slower and then the movement would be detected. We use a threshold to determine when movement is detected in the optical flow. This threshold can be lower, but then there will be much more noise that would influence the system.

If we would be able to reduce the drift as it occurs to the velocity estimation then we would have a much more accurate positioning system. Therefore we introduced stereoscopic vision which we use to construct point clouds. Those point clouds can be used to estimate a velocity, however the stereo camera system we build is not suitable to use to build correct point clouds. Therefore we had to stick to direction estimation. Velocity can be estimated in

the same manner as we presented if better hardware is available. A professional stereo camera system would suffice, but a structured light sensor like the Kinect can be used as well. Those systems and sensors have integrated microsystems which compute the point cloud. Those sensors are also more reliable than any experimental setup and can truly estimate a velocity at a decent sample rate.

References

[1] (2014, August) Oxford Dictionaries. [Online]. http://www.oxforddictionaries.com/definition/english/augmented-reality

[2] Borko Furht, *Handbook of Augmented Reality*. New York: Springer, 2011.

[3] David Goldman. (2012, April) CNN Money - Google unveils 'Project Glass' virtual-reality glasses. [Online]. http://money.cnn.com/2012/04/04/technology/google-project-glass/?source=cnn_bin

[4] (2012, August) BBC News technology - Oculus Rift virtual reality headset gets Kickstarter cash. [Online]. http://www.bbc.com/news/technology-19085967

[5] Hugh Durrant-Whyte and Tim Bailey, "Simultaneous localization and mapping: part I," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99-110, 2006.

[6] Michael Montemerlo and Sebastian Thrun, "FastSLAM 2.0," in *FastSLAM.*: Springer, 2007, pp. 63-90.

[7] Jiaqing Wu, "Three-Dimensional Indoor RFID Localization System," University of Nebraska - Lincoln, Lincoln, Dissertation 2012.

[8] Lionel M. Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Networks*, vol. 10, pp. 701-710, 2004.

[9] Sinan Gezici, Zhi Tian, Georgios B. Biannakis, Hisashi Koboyashi, Andreas F. Molisch, H. Vincent Poor, and Zafer Sahinoglu, "Localization via Ultra-Wideband radios: a look at positioning aspects for future sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 70-84, 2005.

[10] Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Guillaume Marceau, Lydia E. Kavraki, and Dan S. Wallach, "Robotics-based location sensing using wireless ethernet," *Wireless Networks*, vol. 11, no. 1-2, pp. 189-204, 2005.

[11] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert, "Robust Monte Carlo Localization for Mobile Robots," *Artificial intellegence*, vol. 128, no. 1, pp. 99-141, April 2001.

[12] Gamini Dissanayake, Hugh Durrant-Whyte, and Tim Bailey, "A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem," *Robotics and Automation. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2, pp. 1009-1014, 2000.

[13] Andreas Nüchter, Hartmut Surmann, Kai Lingemann, and Joachim Hertzberg, "6D SLAM with an application in autonomous mine mapping," *Robotics and Automation. Proceedings. ICRA'04. IEEE International Conference on*, vol. 2, pp. 1998-2003, 2004.

[14] C. S. Draper, W. Wrigley, G. Hoag, R. H. Battin, J. E. Miller, A. Koso, A. L. Hopkins, and W. E. Vander Velde, "Apollo on-board guidance, navigation, and control system," *International space hall of fame dedication conference in Alamogordo, N.M.*, vol. 1, October 1976.

[15] Eric Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *Computer Graphics and Applications, IEEE*, vol. 26, no. 6, pp. 38-46, 2005.

[16] Antonio R. Jiménez, Fernando Seco, Francisco Zampella, José C. Prieto, and Jorge Guevara, "PDR with a foot-mounted IMU and ramp detection," *Sensors (Basel, Switzerland)*, vol. 10, no. 10, pp. 9393-9410, 2011.

[17] Eric M. Foxlin, Michael Harrington, and Yury Altshuler, "Miniature six-DOF inertial systems for tracking HMDs," *Aerospace/Defense Sensing and Controls. International Society for Optics and Photonics*, pp. 214-228, August 1998.

[18] Nirupama Bulusu, John Heidemann, and Deborah Estrin, "GPS-less low cost outdoor localization for very small devices," *Personal Communications, IEEE*, pp. 28-32, 2000.

[19] Sinan Gezici, Zhi Tian, Georgios B. Biannakis, Hisashi Kobayashi, Andreas F. Molisch, H. Vincent Poor, and Zafer Sahinoglu, "Localization via Ultra-Wideband radios," *Signal Processing Magazine, IEEE*, pp. 70-84, 2005.

[20] V. Pierlot, M. Urbin-Choffray, and M. Van Droogenbroeck. (2011, June) A new three object triangulation algorithm based on the power center of three circles. [Online]. http://www2.ulg.ac.be/telecom/publi/publications/pierlot/Pierlot2011ANewThreeObject/index.html

[21] Nan Wu and Xiangdong Li. (2011) RFID Applications in Cyber-Physical System. [Online]. http://www.intechopen.com/books/deploying-rfid-challenges-solutions-and-open-issues/rfid-applications-in-cyber-physical-system

[22] Hugh Durrant-Whyte and Tim Bailey, "Simultaneous localization and mapping: Part I," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99-110, 2006.

[23] GMV. (2011) GNSS Augmentation - Navipedia. [Online]. http://www.navipedia.net/index.php/GNSS_Augmentation

[24] (2013, December) Novatel. [Online]. http://www.novatel.com/an-introduction-to-gnss/chapter-4-advanced-gnss-concepts/real-time-kinematic-rtk/

[25] M. Norremark, H.W. Griepentrog, J. Nielsen, and H.T. Sogaard, "The development and assessment of the accuracy of an autonomous GPS-based system for intra-row mechanical weed control in row crops," *Biosystems engineering*, pp. 396-410, 2008.

[26] (2013, December) University of Maryland - Ranger NBV. [Online]. http://www.ssl.umd.edu/projects/RangerNBV/thesis/2-4-1.htm

[27] Walter Galan, Andrew Optimus Goldberg, Sam Lionheart, and David Hodson. (2013, April) Oculus Rift Teardown. [Online]. https://www.ifixit.com/Teardown/Oculus+Rift+Teardown/13682

[28] (2013, January) Oculus VR Blog. [Online]. http://www.oculusvr.com/blog/building-a-sensor-for-low-latency-vr/

[29] (2014, March) Oculus Rift Teardown - iFixit. [Online]. http://www.ifixit.com/Teardown/Oculus+Rift+Teardown/13682/

[30] Rob O'Reilly, Kieran Harney, Analog Devices Inc., and Alex Khenkin. (2009, June) Sonic Nirvana: MEMS Accelerometers as acoustic pickups in musical instruments. [Online]. http://www.sensorsmag.com/sensors/acceleration-vibration/sonic-nirvana-mems-accelerometers-acoustic-pickups-musical-i-5852

[31] Rob Candler. (2008, April) Capacitive sensors and accelerometer fundamentals. [Online]. http://www.stanford.edu/class/me220/data/lectures/lect08/lect_4.html

[32] Rob Candler. (2008, May) Accelerometer. [Online]. http://www.stanford.edu/class/me220/data/lectures/lect09/lect_5.html

[33] Rik Pintelon and Johan Schoukens, *System identification: a frequency domain approch.*: John Willey & Sons, 2012.

[34] Antonio R. Jiménez, Fernando Seco, Francisco Zampella, José C. Prieto, and Jorge Guevara, "PDR with a foot-mounted IMU and ramp detection," *Sensors*, vol. 11, no. 10, pp. 9393-9410, 2011.

[35] Clizia Welker and Hugo Quené. (2006) University of Utrecht - Phonetics - Basics of acoustics 1. [Online]. http://www.hum.uu.nl/uilots/lab/courseware/phonetics/basics_of_acoustics_1/sampling_rate.html

[36] Zhaohua Wu and Norden E. Huang, "A study of the characteristics of white noise using the empirical mode decomposition method," in *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 460(2046)*, London, 2004, pp. 1597-1611.

[37] Erik K. Antonsson and Robert W. Mann, "The frequency content of gait," *Journal of biomechanics*, vol. 18, no. 1, pp. 39-47, 1985.

[38] W. T. Higgings, "A comparison of complementary and Kalman filtering," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 3, pp. 321-325, 1975.

[39] Shane Colton and F. Mentor. (2007) Massachusetts Institute of Technology. [Online]. http://web.mit.edu/scolton/www/filter.pdf

[40] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and Per-Johan Nordlund, "Particle filters for positioning, navigation and tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 425-437, 2002.

[41] X. Rong Li and Vesselin P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," *Aerospace and Electronic Systems, IEEE, Transactions on*, vol. 39, no. 4, pp. 1333-1364, 2003.

[42] Wouter L. de Vries. (2013, October) Stackoverflow - signal processing - lowpass and high pass filter in C#. [Online]. http://stackoverflow.com/a/19155926

[43] Patrice Tarrabia. (2002, March) Music-DSP Source Code Archive. [Online]. http://www.musicdsp.org/archive.php?classid=3#38

[44] Frank Steinbrücker, Jürgen Strum, and Daniel Cremers, "Volumetric 3D Mapping in Real-Time on a CPU," 2014.

[45] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss, and Alexander Kleiner, "On measuring the accuracy of SLAM algorithms," *Autonomous Robots*, vol. 27, no. 4, pp. 387-407, November 2009.

[46] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *Pattern Analysis and Machine Intelligence, IEEE Transaction on*, vol. 29, no. 6, pp. 1052-1067, 2007.

[47] Wen Li and Dezhen Song, "Featureless motion vector-based simultaneous localization, planar surface extraction, and moving obstacle tracking," July 2014.

[48] Florian Raudies. (2013) Optical flow - Scholarpedia. [Online]. http://www.scholarpedia.org/article/Optic_flow

[49] Myron Z. Brown, Darius Burschka, and Gregory D. Hager, "Advances in computational stereo," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 8, pp. 933-1008, 2003.

Appendix I

-

Kalman filter experiment

Individual run data

Run 1

q\r	1	11	21	31	41	51	61	71	81	91	101
0.01	163.80292	41.812611	41.364662	54.160355	60.667744	63.99295	65.898758	67.086014	67.873367	68.421326	68.817619
0.02	174.7415	56.012386	36.900227	42.468502	52.127949	58.044426	61.557495	63.794033	65.298332	66.355553	67.12545
0.03	179.72005	66.295662	39.30125	37.91481	44.888298	52.6171	57.496315	60.664097	62.822887	64.35405	65.476494
0.04	182.60541	74.524818	43.336948	36.865093	40.802303	47.637276	53.685654	57.683014	60.440651	62.413506	63.868946
0.05	184.47815	81.015228	47.62273	37.30595	38.355793	43.67271	50.09985	54.838871	58.14558	60.530689	62.300938
0.06	185.79047	86.317955	51.561417	38.548187	37.155479	41.164188	46.72974	52.121613	55.932644	58.70295	60.771023
0.07	186.75923	90.86647	55.065338	40.077423	36.86446	39.374378	44.000801	49.521584	53.796761	56.927452	59.27758
0.08	187.50281	94.795761	58.177868	41.922295	36.992996	38.044407	42.03204	47.030113	51.733273	55.201553	57.818989
0.09	188.09033	98.270378	61.075317	43.882614	37.446941	37.273647	40.60865	44.881145	49.738049	53.5229	56.393864
0.10	188.5666	101.41275	63.883698	45.884518	38.160961	36.96677	39.389206	43.187576	47.807697	51.889626	55.001186
0.11	188.95924	104.25295	66.461449	47.821339	38.917576	⭐ 36.8463	38.412144	41.837753	45.994408	50.299545	53.639645
0.12	189.2894	106.82181	68.896393	49.669479	39.841183	36.920635	37.655212	40.79763	44.456825	48.750664	52.307892
0.13	189.57021	109.17163	71.181824	51.433533	40.877804	37.136887	37.237507	39.856083	43.206108	47.241249	51.005028
0.14	189.81143	111.32527	73.347298	53.070831	41.929203	37.476387	37.010323	39.037373	42.13068	45.840771	49.729713
0.15	190.02156	113.33234	75.351311	54.670715	43.057068	37.911308	36.909294	38.352253	41.267166	44.62479	48.481178
0.16	190.2067	115.20164	77.234283	56.181313	44.164883	38.429962	⭐ 36.8466	37.773724	40.510601	43.595203	47.258236
0.17	190.37032	116.93543	79.006798	57.586964	45.32045	38.908348	36.909145	37.387924	39.802181	42.668369	46.109222
0.18	190.51625	118.56045	80.660934	58.934563	46.460258	39.494381	37.042038	37.153942	39.169117	41.887653	45.087532
0.19	190.64833	120.08449	82.209496	60.261551	47.55798	40.111507	37.21743	37.006626	38.606628	41.226212	44.157875
0.20	190.76752	121.53118	83.688866	61.599556	48.639996	40.800865	37.468681	36.929726	38.128342	40.626072	43.369823
0.21	190.87434	122.89832	85.11837	62.899437	49.679039	41.466125	37.760468	36.863781	37.707821	40.04541	42.627151
0.22	190.97328	124.19039	86.497704	64.139328	50.695114	42.173107	38.111115	36.85247	37.420097	39.514969	41.991169
0.23	191.0634	125.41895	87.815628	65.32795	51.686653	42.90189	38.468605	36.907654	37.224709	39.036366	41.43552
0.24	191.14693	126.58442	89.068138	66.495186	52.627689	43.629654	38.804665	36.994228	37.069714	38.598667	40.935379
0.25	191.22415	127.6984	90.269485	67.630348	53.527626	44.34314	39.163273	37.114719	36.984138	38.218609	40.451138
0.26	191.29605	128.7715	91.421844	68.73214	54.455418	45.089722	39.602112	37.264847	36.931129	37.866211	39.988701
0.27	191.36284	129.79532	92.527794	69.807495	55.343086	45.836464	40.031693	37.44915	36.881332	37.583508	39.558117
0.28	191.42517	130.77269	93.594833	70.846947	56.1917	46.568214	40.507881	37.658951	⭐ 36.8433	37.375771	39.164043
0.29	191.4836	131.71266	94.635284	71.861855	57.004868	47.280357	40.982506	37.893627	36.867588	37.228466	38.791431
0.30	191.53835	132.61513	95.64109	72.856606	57.794235	47.979874	41.445923	38.163998	36.910091	37.098053	38.459579
0.31	191.59062	133.48085	96.607956	73.813683	58.562752	48.677322	41.934761	38.427559	36.971252	37.020531	38.157242
0.32	191.63933	134.31291	97.540047	74.735138	59.329323	49.354156	42.434616	38.679516	37.066162	36.961121	37.873692
0.33	191.6859	135.11272	98.448441	75.630104	60.081181	50.017902	42.951836	38.920555	37.154827	36.924126	37.634888
0.34	191.72974	135.88217	99.333313	76.500824	60.841793	50.66869	43.463081	39.204735	37.287167	36.884499	37.454128
0.35	191.77185	136.62448	100.19658	77.346733	61.61124	51.316002	43.956329	39.530884	37.426701	36.846596	37.306934
0.36	191.81133	137.33997	101.04318	78.172447	62.363144	51.9422	44.464035	39.845772	37.582371	36.851048	37.199192
0.37	191.85065	138.03453	101.86288	78.971107	63.090561	52.547443	44.985001	40.169724	37.755844	36.881725	37.094673
0.38	191.88646	138.70654	102.65733	79.745209	63.800137	53.133682	45.513645	40.532143	37.943481	36.914284	37.030346

0.39	191.92218	139.35678	103.43128	80.495155	64.497322	53.720303	46.026115	40.883183	38.154472	36.961224	36.977062
0.40	191.95502	139.98648	104.18579	81.223511	65.174225	54.32082	46.539833	41.224026	38.358536	37.037659	36.944046
0.41	191.98672	140.59639	104.91842	81.930153	65.843765	54.902458	47.040802	41.573009	38.555397	37.109951	36.913258
0.42	192.01802	141.18727	105.63006	82.617157	66.507805	55.467987	47.52919	41.93573	38.745552	37.185364	36.880539
0.43	192.04749	141.76378	106.32247	83.297035	67.161476	56.01704	48.019463	42.305485	38.92942	37.296532	36.847889
0.44	192.07724	142.32414	107.00027	83.967316	67.80146	56.551189	48.508194	42.678864	39.144146	37.40453	36.847874
0.45	192.10439	142.86876	107.6623	84.626961	68.428757	57.070683	48.98838	43.063999	39.393612	37.529156	36.865009
0.46	192.13106	143.39803	108.30757	85.269928	69.056938	57.58065	49.456692	43.440231	39.640327	37.659634	36.893341
0.47	192.15701	143.91248	108.93668	85.90686	69.669731	58.086636	49.920406	43.806633	39.880013	37.796349	36.919041
0.48	192.18269	144.41315	109.55028	86.53421	70.26709	58.579159	50.379646	44.171963	40.129494	37.949844	36.958618
0.49	192.20679	144.90044	110.14825	87.146652	70.860214	59.074883	50.833462	44.55035	40.406109	38.117699	37.020985
0.50	192.23016	145.37485	110.73594	87.744698	71.442337	59.56778	51.284279	44.935936	40.679756	38.280777	37.080708
0.51	192.2533	145.83728	111.31182	88.328857	72.019707	60.051247	51.724949	45.328152	40.946949	38.439304	37.137711
0.52	192.27559	146.2885	111.87719	88.899849	72.589355	60.543335	52.155125	45.712025	41.207924	38.593716	37.209385
0.53	192.29715	146.72969	112.4312	89.457687	73.147079	61.035259	52.575733	46.087635	41.473629	38.744076	37.299335
0.54	192.31764	147.16014	112.97355	90.009163	73.692131	61.534084	52.986549	46.468216	41.751156	38.890472	37.386307
0.55	192.33846	147.58282	113.50806	90.550339	74.225891	62.024906	53.389168	46.840977	42.031963	39.043102	37.485661
0.56	192.35878	147.99719	114.03201	91.079796	74.747688	62.504654	53.808006	47.205791	42.314919	39.234131	37.584904
0.57	192.37813	148.40369	114.5444	91.598289	75.260277	62.974819	54.227493	47.566135	42.60183	39.43372	37.697193
0.58	192.3968	148.80087	115.04613	92.106422	75.769501	63.435276	54.638344	47.929123	42.896763	39.628948	37.808498
0.59	192.41554	149.18935	115.53771	92.609497	76.267891	63.892616	55.040478	48.288357	43.189869	39.819714	37.93243
0.60	192.4332	149.56915	116.01937	93.103302	76.757195	64.343956	55.434639	48.648384	43.477036	40.014114	38.068863
0.61	192.45117	149.94098	116.49157	93.587166	77.23983	64.786591	55.82074	49.001278	43.758583	40.218594	38.202652
0.62	192.46852	150.30429	116.95435	94.066147	77.717224	65.220535	56.199642	49.34779	44.037479	40.439743	38.333775
0.63	192.48627	150.66116	117.40842	94.543022	78.18528	65.652878	56.570892	49.687954	44.325096	40.656723	38.461761
0.64	192.50186	151.0098	117.8574	95.01136	78.645149	66.0868	56.935265	50.032166	44.616547	40.869282	38.586708
0.65	192.51843	151.35165	118.29786	95.470894	79.096291	66.513031	57.292435	50.370365	44.913189	41.077965	38.709183
0.66	192.53403	151.68674	118.73116	95.92247	79.539581	66.936974	57.649578	50.704063	45.215019	41.282948	38.828957
0.67	192.55003	152.01448	119.15624	96.365913	79.974823	67.355957	58.003216	51.040428	45.512856	41.49604	38.946308
0.68	192.56503	152.33627	119.57324	96.801865	80.402489	67.768227	58.350571	51.370972	45.805618	41.716743	39.081482
0.69	192.57988	152.65132	119.98338	97.230637	80.822906	68.173767	58.694202	51.695393	46.093079	41.936981	39.236904
0.70	192.59451	152.96175	120.38931	97.653114	81.235992	68.581055	59.039906	52.014435	46.38623	42.163265	39.399036
0.71	192.60905	153.26501	120.78877	98.072357	81.642456	68.986832	59.386482	52.3284	46.675163	42.385952	39.558472
0.72	192.62273	153.56339	121.18387	98.484116	82.042076	69.385498	59.727238	52.636646	46.958763	42.61512	39.714943
0.73	192.63696	153.85576	121.57307	98.893326	82.435646	69.777664	60.064465	52.939541	47.237747	42.848148	39.868622
0.74	192.65034	154.14316	121.95535	99.296837	82.824196	70.163956	60.409805	53.237717	47.513359	43.081818	40.028591
0.75	192.66356	154.4252	122.33196	99.694252	83.213379	70.548805	60.75214	53.536793	47.793552	43.31176	40.192646
0.76	192.67657	154.70328	122.70174	100.08987	83.598564	70.929138	61.096794	53.850361	48.069149	43.538033	40.372837
0.77	192.68983	154.97786	123.06654	100.48369	83.980354	71.305077	61.446613	54.159943	48.346558	43.760471	40.550083

0.78	192.70279	155.24869	123.42467	100.8712	84.359299	71.679527	61.79184	54.464664	48.623047	43.979809	40.724621
0.79	192.71469	155.51591	123.77782	101.25324	84.732941	72.051971	62.132084	54.765217	48.895561	44.205582	40.896851
0.80	192.72731	155.77776	124.1268	101.62981	85.100937	72.421844	62.467449	55.061291	49.163902	44.437065	41.06641
0.81	192.7395	156.03555	124.47173	102.0003	85.465927	72.785927	62.797318	55.35244	49.428196	44.669449	41.233276
0.82	192.75143	156.28894	124.81152	102.36597	85.828941	73.144539	63.122265	55.639217	49.688774	44.90398	41.403847
0.83	192.76291	156.53825	125.14621	102.7265	86.190536	73.498398	63.442974	55.92226	49.953747	45.142891	41.578758
0.84	192.77469	156.7838	125.4759	103.0825	86.547638	73.847282	63.763256	56.201145	50.215588	45.380413	41.758236
0.85	192.7858	157.02542	125.80091	103.43634	86.899239	74.191383	64.081459	56.476151	50.47345	45.614433	41.937321
0.86	192.79715	157.26262	126.12168	103.78499	87.246452	74.530518	64.394966	56.746799	50.729641	45.845039	42.121178
0.87	192.80804	157.49654	126.43681	104.12916	87.588509	74.864975	64.704453	57.013954	50.988274	46.072868	42.302784
0.88	192.81903	157.72688	126.74828	104.46895	87.926277	75.195518	65.009941	57.2775	51.243637	46.305279	42.485962
0.89	192.82956	157.95343	127.05692	104.80373	88.259262	75.526321	65.311203	57.540424	51.495323	46.535393	42.67168
0.90	192.84067	158.17731	127.36191	105.13437	88.58783	75.852669	65.613876	57.803249	51.743313	46.762074	42.862526
0.91	192.85089	158.39688	127.66341	105.46089	88.912697	76.175285	65.9179	58.063278	51.988529	46.986462	43.052349
0.92	192.86104	158.61375	127.96313	105.78272	89.23291	76.493439	66.219208	58.319332	52.230263	47.20779	43.239628
0.93	192.87155	158.82738	128.25928	106.10094	89.549492	76.808174	66.516632	58.572224	52.46896	47.42617	43.424404
0.94	192.88126	159.0381	128.55193	106.41518	89.865555	77.119774	66.812477	58.827827	52.704464	47.647324	43.606895
0.95	192.8914	159.2455	128.83978	106.72697	90.177856	77.430588	67.107666	59.083042	52.936939	47.867561	43.786823
0.96	192.90109	159.45024	129.12531	107.03611	90.48645	77.737663	67.399582	59.338814	53.16658	48.085556	43.964836
0.97	192.91048	159.65114	129.40576	107.3418	90.791626	78.041283	67.68795	59.591988	53.394497	48.305779	44.147949
0.98	192.9201	159.85043	129.68349	107.64374	91.092834	78.34079	67.972855	59.841179	53.629635	48.525753	44.334129
0.99	192.92964	160.04659	129.95792	107.94241	91.390251	78.636864	68.255569	60.091103	53.8703	48.74268	44.521187
1.00	192.93913	160.23976	130.22806	108.23718	91.684349	78.929535	68.540085	60.346184	54.108421	48.957325	44.711506
1.01	192.94774	160.43036	130.49596	108.52893	91.975311	79.218994	68.824982	60.598099	54.343842	49.169277	44.901592

Run 2

q\r	1	11	21	31	41	51	61	71	81	91	101
0.01	469.61838	123.76527	38.083599	14.5415	12.7878	19.2611	23.2415	25.7196	27.3624	28.5054	29.3319
0.02	492.76965	184.06284	77.857277	34.355125	16.9316	11.8742	14.18	18.8455	21.9878	24.195	25.8019
0.03	503.39917	220.3755	107.97887	54.41011	28.4031	16.3094	12.0571	12.783	16.8159	20.0155	22.3598
0.04	509.78	245.89285	131.53635	72.079254	40.28968	23.4626	15.0602	11.9915	12.5352	15.9601	19.002
0.05	514.12781	265.36172	150.67831	87.591095	51.694313	31.160643	19.6899	13.8741	11.8476	12.623	15.7248
0.06	517.31805	281.00479	166.71716	101.26058	62.3106	38.861622	24.9374	16.9399	12.9648	11.7921	12.9235
0.07	519.78265	293.98804	180.43993	113.43868	72.108215	46.344852	30.335041	20.5312	14.9357	12.3287	11.8981
0.08	521.75616	305.02631	192.39478	124.36696	81.212593	53.568352	35.726707	24.4366	17.4442	13.553	11.9471
0.09	523.37903	314.58679	202.95061	134.23912	89.684341	60.424908	41.062538	28.4178	20.2137	15.2426	12.6542
0.10	524.74518	322.98999	212.39432	143.23718	97.561836	66.937057	46.272789	32.406693	23.1891	17.2412	13.7235
0.11	525.91449	330.4613	220.91389	151.50867	104.9159	73.13472	51.356201	36.373718	26.216	19.4061	15.1118
0.12	526.93134	337.17346	228.6658	159.13733	111.81965	79.044495	56.273441	40.308334	29.2928	21.7051	16.7004
0.13	527.82288	343.25055	235.77133	166.20868	118.29866	84.688614	61.006256	44.180367	32.358952	24.0981	18.4209
0.14	528.61621	348.79077	242.32367	172.78664	124.4034	90.074455	65.572334	47.98941	35.402908	26.4982	20.2257
0.15	529.32703	353.87344	248.39876	178.93962	130.16205	95.215378	69.979095	51.715134	38.454216	28.9402	22.1232
0.16	529.96844	358.55966	254.05339	184.7126	135.612	100.11944	74.241043	55.357311	41.455372	31.359028	24.0708
0.17	530.55225	362.89941	259.3374	190.14171	140.78378	104.82299	78.366234	58.887928	44.416431	33.792091	26.013
0.18	531.08313	366.93674	264.29462	195.26167	145.71111	109.34191	82.364311	62.335732	47.345165	36.20401	27.9968
0.19	531.5733	370.70636	268.96548	200.10863	150.40768	113.66891	86.228027	65.684113	50.230213	38.616386	29.9617
0.20	532.02563	374.23816	273.37936	204.70576	154.89261	117.83235	89.97168	68.948112	53.064274	40.992363	31.938595
0.21	532.44495	377.55508	277.55527	209.08022	159.17715	121.83641	93.593254	72.134895	55.839161	43.352489	33.905289
0.22	532.83636	380.68158	281.51721	213.24646	163.28102	125.69048	97.095245	75.236511	58.550411	45.681484	35.861046
0.23	533.20007	383.63339	285.28278	217.2224	167.21733	129.40144	100.48956	78.274139	61.213322	47.994648	37.827461
0.24	533.54309	386.42731	288.87012	221.02357	170.99426	132.98248	103.78452	81.240318	63.819626	50.273472	39.764206
0.25	533.86285	389.07822	292.29236	224.66267	174.62517	136.44308	106.99426	84.133316	66.370415	52.520615	41.69326
0.26	534.16461	391.59875	295.56491	228.15532	178.12387	139.78688	110.10787	86.954865	68.871384	54.736591	43.600292
0.27	534.45111	393.99832	298.69742	231.50928	181.498	143.02785	113.13077	89.713089	71.329208	56.908215	45.490845
0.28	534.72058	396.28522	301.70016	234.73587	184.75171	146.17056	116.06902	92.403816	73.737274	59.042686	47.369858
0.29	534.9776	398.47202	304.58374	237.8445	187.8952	149.21928	118.93282	95.031059	76.096596	61.151237	49.234215
0.30	535.22113	400.56293	307.35495	240.84219	190.93121	152.1745	121.71893	97.591118	78.423042	63.222553	51.070866
0.31	535.45306	402.56647	310.02289	243.73538	193.86826	155.04565	124.43207	100.09657	80.707672	65.257164	52.890293
0.32	535.67499	404.48953	312.59271	246.53149	196.71315	157.83165	127.07195	102.54288	82.94558	67.259178	54.686455
0.33	535.88599	406.33527	315.07162	249.23543	199.47214	160.53972	129.64632	104.94702	85.141151	69.231995	56.452789
0.34	536.08746	408.10944	317.46487	251.85248	202.14777	163.17651	132.15756	107.30125	87.299507	71.178444	58.195301
0.35	536.28094	409.81812	319.77686	254.38733	204.74489	165.74115	134.60693	109.60124	89.416801	73.093369	59.916042
0.36	536.46692	411.46487	322.0145	256.84592	207.27167	168.23894	137.00108	111.85256	91.496696	74.976013	61.619488
0.37	536.64581	413.05267	324.17972	259.2309	209.7274	170.67024	139.33575	114.05338	93.53688	76.834084	63.296936
0.38	536.81592	414.58649	326.27676	261.54596	212.11603	173.0412	141.62027	116.21394	95.538414	78.670792	64.948463

0.39	536.97998	416.06854	328.31113	263.7977	214.44048	175.35378	143.85759	118.33074	97.50116	80.479485	66.580597
0.40	537.13934	417.4996	330.2858	265.98917	216.70653	177.6136	146.04927	120.40755	99.434418	82.26075	68.194519
0.41	537.29144	418.88721	332.20248	268.12045	218.91159	179.81956	148.19038	122.44232	101.32941	84.014549	69.786499
0.42	537.43866	420.22913	334.064	270.19888	221.06358	181.97481	150.28783	124.43721	103.19665	85.741394	71.361053
0.43	537.58295	421.52997	335.87442	272.2226	223.16154	184.07927	152.34122	126.39256	105.03696	87.445328	72.915886
0.44	537.72052	422.79282	337.63507	274.19641	225.21089	186.13821	154.35388	128.31152	106.84943	89.125504	74.449409
0.45	537.85358	424.01691	339.34772	276.11984	227.21135	188.14944	156.32376	130.19489	108.62891	90.778954	75.961632
0.46	537.98169	425.2066	341.01697	277.99814	229.16687	190.11913	158.25462	132.04489	110.37968	92.410881	77.465942
0.47	538.10779	426.36243	342.64313	279.83099	231.07819	192.04604	160.14713	133.86105	112.0996	94.016411	78.951126
0.48	538.2276	427.48621	344.22827	281.62082	232.94702	193.93129	162.00356	135.64566	113.79067	95.597824	80.417862
0.49	538.34558	428.57907	345.77463	283.37024	234.7766	195.77905	163.82759	137.39983	115.45872	97.156792	81.866013
0.50	538.45911	429.64316	347.28256	285.0791	236.56577	197.58951	165.61525	139.12175	117.10122	98.693901	83.297295
0.51	538.57031	430.67996	348.75528	286.75217	238.31952	199.36612	167.37068	140.81679	118.71793	100.20936	84.709641
0.52	538.67786	431.68924	350.19339	288.38788	240.03566	201.10548	169.0925	142.48431	120.30946	101.70275	86.10553
0.53	538.78296	432.67365	351.59763	289.98825	241.71889	202.81248	170.78493	144.12831	121.8795	103.1816	87.487686
0.54	538.88519	433.63315	352.97159	291.55463	243.36681	204.4874	172.4454	145.7451	123.42443	104.64265	88.852287
0.55	538.98401	434.56973	354.31265	293.0892	244.98433	206.13364	174.07745	147.33469	124.94518	106.08484	90.198647
0.56	539.0799	435.48419	355.62674	294.59259	246.56982	207.74814	175.68257	148.89978	126.44327	107.50783	91.531517
0.57	539.17413	436.3757	356.91208	296.06628	248.12608	209.33577	177.2628	150.442	127.92036	108.91277	92.849274
0.58	539.26605	437.24811	358.16928	297.50992	249.65392	210.8945	178.8167	151.96016	129.37692	110.29871	94.150185
0.59	539.35565	438.09973	359.40039	298.92645	251.15198	212.42422	180.34442	153.45493	130.81221	111.66463	95.433937
0.60	539.44189	438.93265	360.60699	300.31598	252.62444	213.92886	181.84737	154.92654	132.22699	113.01237	96.701744
0.61	539.52753	439.74741	361.78848	301.6792	254.06863	215.40733	183.32576	156.37569	133.62233	114.3425	97.955566
0.62	539.60999	440.54349	362.9476	303.01682	255.48965	216.86201	184.78162	157.80486	135.00092	115.66062	99.197357
0.63	539.69281	441.3233	364.082	304.33017	256.88486	218.29144	186.21443	159.21182	136.35971	116.96239	100.42397
0.64	539.77081	442.08691	365.19571	305.62012	258.25629	219.69814	187.62399	160.59793	137.69904	118.24638	101.63491
0.65	539.84851	442.83466	366.28824	306.88605	259.60419	221.08125	189.01256	161.96591	139.02068	119.51473	102.83567
0.66	539.92322	443.56622	367.36029	308.1322	260.9303	222.44391	190.38028	163.31615	140.32787	120.77009	104.02782
0.67	539.99731	444.28397	368.41129	309.35513	262.23401	223.78404	191.72708	164.64651	141.61629	122.00951	105.20647
0.68	540.07043	444.98688	369.44388	310.55664	263.51907	225.10471	193.0547	165.95903	142.89281	123.23504	106.37424
0.69	540.14044	445.67545	370.45807	311.73953	264.78357	226.40543	194.36275	167.2527	144.15158	124.44371	107.52685
0.70	540.21021	446.35205	371.453	312.90155	266.02692	227.68553	195.65108	168.52811	145.39619	125.63873	108.66792
0.71	540.27747	447.01523	372.43222	314.04611	267.25287	228.94803	196.92357	169.78801	146.6252	126.82053	109.79746
0.72	540.34497	447.66614	373.39355	315.16992	268.45932	230.19089	198.17714	171.03021	147.83835	127.98743	110.91331
0.73	540.41071	448.30389	374.33804	316.27649	269.64886	231.41547	199.41322	172.25496	149.03667	129.14067	112.01662
0.74	540.47333	448.93073	375.26706	317.36597	270.82037	232.62323	200.63248	173.46504	150.22168	130.28287	113.10906
0.75	540.53693	449.54599	376.181	318.43942	271.97464	233.81557	201.83643	174.66103	151.3929	131.41214	114.18969
0.76	540.59802	450.1521	377.07932	319.49466	273.11258	234.98956	203.02292	175.84131	152.54985	132.52817	115.26098
0.77	540.65668	450.74417	377.96188	320.53522	274.23459	236.14774	204.19351	177.00739	153.69308	133.6317	116.32217

0.78	540.71747	451.32797	378.83243	321.5589	275.3396	237.28992	205.35054	178.15918	154.82317	134.72441	117.37263
0.79	540.77679	451.90125	379.68738	322.56879	276.42981	238.4176	206.49353	179.29825	155.94066	135.80653	118.41371
0.80	540.83191	452.46423	380.53024	323.56302	277.50504	239.53059	207.62207	180.42351	157.04608	136.87624	119.4436
0.81	540.88757	453.01822	381.3587	324.54257	278.56442	240.6283	208.73555	181.53371	158.13715	137.93416	120.4642
0.82	540.94336	453.56238	382.17429	325.50812	279.60995	241.71165	209.83443	182.6315	159.2151	138.97951	121.47374
0.83	540.99548	454.09875	382.97821	326.45926	280.64099	242.78065	210.92036	183.71567	160.28229	140.01619	122.47444
0.84	541.04907	454.62558	383.76923	327.39822	281.65915	243.83684	211.99338	184.78856	161.33891	141.04198	123.4649
0.85	541.10162	455.14383	384.54898	328.32373	282.6636	244.87959	213.05318	185.84785	162.38432	142.05852	124.44592
0.86	541.15314	455.65442	385.31812	329.23654	283.65503	245.90939	214.09967	186.89433	163.41725	143.06497	125.41638
0.87	541.20361	456.1575	386.07486	330.13745	284.63315	246.92563	215.13383	187.9294	164.4408	144.06323	126.37807
0.88	541.25189	456.65155	386.82138	331.02676	285.59915	247.93173	216.15637	188.95349	165.45317	145.05194	127.33166
0.89	541.3009	457.1384	387.55649	331.90222	286.55304	248.92365	217.16631	189.96524	166.45374	146.03015	128.27525
0.90	541.34937	457.6181	388.28107	332.76666	287.49493	249.90413	218.1638	190.96567	167.44376	146.99794	129.20967
0.91	541.39526	458.09009	388.99734	333.6214	288.42499	250.87349	219.15123	191.95633	168.42427	147.95718	130.13788
0.92	541.44171	458.55475	389.70081	334.46353	289.34409	251.83112	220.12714	192.93555	169.39444	148.90765	131.05676
0.93	541.48761	459.01242	390.39706	335.29565	290.2518	252.77711	221.09311	193.90369	170.35428	149.84908	131.96706
0.94	541.53156	459.46536	391.08316	336.11642	291.14813	253.71289	222.04704	194.86237	171.3045	150.78085	132.86899
0.95	541.57568	459.91034	391.75992	336.92694	292.03406	254.63676	222.98964	195.80965	172.24425	151.70311	133.7628
0.96	541.61963	460.3493	392.42798	337.72775	292.90927	255.55048	223.9229	196.74846	173.1752	152.61868	134.65002
0.97	541.66193	460.78146	393.08649	338.51968	293.77536	256.45572	224.84767	197.67862	174.09857	153.52644	135.52986
0.98	541.70471	461.20792	393.73611	339.30051	294.63052	257.34872	225.76147	198.59763	175.01205	154.42419	136.40154
0.99	541.74487	461.62802	394.37845	340.07208	295.47543	258.23181	226.66458	199.50668	175.9169	155.31335	137.26443
1.00	541.78711	462.04327	395.01202	340.83533	296.31122	259.10602	227.5591	200.40807	176.81354	156.19525	138.12071
1.01	541.82697	462.45245	395.6376	341.58908	297.13794	259.97174	228.44412	201.29965	177.7014	157.06883	138.96852

Run 3

q\r	1	11	21	31	41	51	61	71	81	91	101
0.01	77.431999	9.9314241	15.417199	22.516928	25.609541	27.190525	28.096872	28.661589	29.036116	29.296795	29.485334
0.02	80.61763	18.286381	8.9080391	16.384909	21.551369	24.362616	26.03253	27.095932	27.811304	28.314127	28.680334
0.03	81.946373	25.937981	8.8424397	11.630346	17.998913	21.783735	24.102116	25.607801	26.634153	27.36223	27.89603
0.04	82.729782	31.501062	10.695454	9.2679749	14.852238	19.418352	22.291407	24.190857	25.501577	26.439501	27.13155
0.05	83.264915	35.742596	12.924485	8.4689932	12.210397	17.237478	20.587967	22.839304	24.410702	25.544392	26.385992
0.06	83.662102	39.147289	15.23701	8.5521946	10.298645	15.217891	18.98156	21.548349	23.359035	24.675602	25.658644
0.07	83.971527	41.974586	17.597998	9.146081	9.2657194	13.375827	17.46282	20.313313	22.344175	23.831772	24.94874
0.08	84.221916	44.380943	19.93705	9.9858313	8.7366581	11.810029	16.023726	19.12999	21.363871	23.011623	24.255476
0.09	84.429726	46.465694	22.1612	10.970009	8.4455919	10.544296	14.657269	17.994827	20.416122	22.214056	23.578215
0.10	84.605942	48.298374	24.196037	11.99987	8.4516573	9.7331543	13.392616	16.904713	19.499275	21.438135	22.916439
0.11	84.757004	49.927467	26.055479	13.032525	8.7117538	9.1852703	12.283209	15.856524	18.611622	20.682831	22.269531
0.12	84.889221	51.388706	27.746635	14.099236	9.0528831	8.843462	11.246484	14.847537	17.751446	19.94713	21.636839
0.13	85.005409	52.710075	29.295822	15.158595	9.4760761	8.5688791	10.462319	13.880333	16.917416	19.23027	21.017931
0.14	85.10894	53.913113	30.723623	16.236919	9.9891739	8.4426289	9.8938513	13.009549	16.10808	18.531322	20.412159
0.15	85.201637	55.014572	32.046913	17.319986	10.554816	8.4339142	9.4441214	12.205868	15.322242	17.849545	19.819138
0.16	85.285408	56.027977	33.278603	18.411636	11.109241	8.510993	9.1064615	11.431458	14.558698	17.184238	19.238325
0.17	85.361496	56.964039	34.429966	19.484411	11.701277	8.7077637	8.8804007	10.775434	13.824883	16.534664	18.669361
0.18	85.43074	57.832386	35.510353	20.530169	12.298758	8.9115295	8.6813745	10.295567	13.148696	15.900326	18.111841
0.19	85.49427	58.640686	36.527229	21.556051	12.88936	9.1602097	8.5077858	9.8799505	12.525515	15.280689	17.565559
0.20	85.553123	59.395802	37.487469	22.53544	13.490711	9.4417238	8.4434395	9.5395966	11.917621	14.674794	17.029655
0.21	85.607277	60.10228	38.396259	23.477041	14.105151	9.7560873	⭐ 8.42674	9.2635994	11.334159	14.082655	16.504383
0.22	85.657661	60.765995	39.258755	24.382105	14.712717	10.105942	8.4462805	9.0492277	10.831844	13.529891	15.988787
0.23	85.70433	61.390385	40.078602	25.249737	15.314869	10.476796	8.5233603	8.8854313	10.434652	13.00845	15.482866
0.24	85.748322	61.979137	40.859982	26.079309	15.939207	10.842728	8.6629391	8.7353134	10.111613	12.516235	14.98653
0.25	85.789169	62.53548	41.605385	26.87351	16.548819	11.196113	8.8000135	8.5955372	9.7964602	12.033605	14.499056
0.26	85.827515	63.062214	42.318409	27.635307	17.170116	11.579485	8.9555216	8.486619	9.5469475	11.560589	14.020457
0.27	85.863495	63.561871	43.001183	28.366825	17.796492	11.974319	9.1272631	8.445385	9.3270645	11.125649	13.574099
0.28	85.897583	64.03643	43.65575	29.070271	18.419344	12.354273	9.3185883	⭐ 8.42775	9.1418533	10.750344	13.143337
0.29	85.929878	64.487854	44.284565	29.747915	19.038086	12.734386	9.5229845	8.4331427	9.0029898	10.442899	12.740546
0.30	85.960365	64.91777	44.888775	30.40102	19.642048	13.122107	9.7453413	8.4520769	8.8772955	10.185671	12.344593
0.31	85.989281	65.328026	45.47047	31.031158	20.240417	13.511299	9.9917307	8.5102615	8.760767	9.9329844	11.954757
0.32	86.016815	65.720139	46.030922	31.6401	20.83371	13.90732	10.240246	8.6089516	8.6532688	9.7012024	11.571294
0.33	86.042732	66.094795	46.571625	32.22905	21.420527	14.305457	10.5019	8.7130566	8.5500259	9.5122948	11.212308
0.34	86.067543	66.45359	47.093536	32.799042	21.990376	14.696792	10.758873	8.8128967	8.4771013	9.3401241	10.891144
0.35	86.091377	66.797363	47.597801	33.351303	22.543869	15.087446	11.007078	8.9264469	8.4475422	9.1885118	10.617969
0.36	86.113663	67.126968	48.085674	33.886803	23.087671	15.480973	11.254761	9.0548038	8.4316521	9.0632582	10.384078
0.37	86.135429	67.443878	48.558018	34.406769	23.616919	15.885368	11.524218	9.1854267	⭐ 8.42652	8.9633865	10.176514
0.38	86.156326	67.748154	49.015194	34.911457	24.135431	16.279831	11.803216	9.3287945	8.4351921	8.8638201	9.9712677

0.39	86.176231	68.040672	49.458454	35.401947	24.642685	16.675064	12.074872	9.4784994	8.450655	8.771615	9.7723227
0.40	86.195145	68.32267	49.888813	35.879253	25.137117	17.076324	12.33958	9.6358337	8.4886608	8.6863489	9.6118641
0.41	86.213646	68.594391	50.306374	36.343357	25.618423	17.482651	12.604715	9.8113155	8.5552998	8.6013441	9.4608641
0.42	86.231125	68.856033	50.711674	36.79533	26.088182	17.886906	12.87365	9.9922256	8.6374493	8.5249901	9.3242435
0.43	86.247902	69.108894	51.105797	37.23576	26.546093	18.28953	13.144728	10.172899	8.7169247	8.4730196	9.2024469
0.44	86.264122	69.352936	51.488968	37.665329	26.993097	18.690115	13.417775	10.364533	8.793869	8.4496403	9.0922508
0.45	86.27961	69.588379	51.861797	38.08395	27.429443	19.088675	13.687914	10.558168	8.8729439	8.4372635	9.0103035
0.46	86.294945	69.816399	52.224533	38.492783	27.855793	19.479828	13.969407	10.747398	8.9710569	8.4277487	8.9292974
0.47	86.30957	70.037079	52.578117	38.891819	28.272919	19.866465	14.247827	10.931818	9.0687332	8.42873	8.8485403
0.48	86.323853	70.250237	52.922478	39.281387	28.680218	20.253378	14.519742	11.112787	9.1680727	8.4353952	8.7745314
0.49	86.337326	70.456917	53.258274	39.66227	29.079018	20.637777	14.795924	11.299285	9.2774506	8.446888	8.7053022
0.50	86.350525	70.657539	53.585621	40.034428	29.469032	21.019335	15.068272	11.498043	9.3907928	8.4726648	8.6362543
0.51	86.36322	70.851715	53.905376	40.39872	29.85129	21.397825	15.339382	11.705312	9.5070772	8.5139742	8.5679836
0.52	86.375595	71.040047	54.217335	40.754799	30.225298	21.768356	15.623135	11.908301	9.6279554	8.5719032	8.5119982
0.53	86.387512	71.223106	54.521969	41.103481	30.591743	22.131954	15.9044	12.107637	9.7600365	8.6368895	8.4718456
0.54	86.399208	71.400574	54.819286	41.444717	30.951004	22.488579	16.17984	12.302831	9.9005795	8.6999979	8.4516172
0.55	86.410309	71.573105	55.110062	41.779057	31.303181	22.842754	16.455727	12.498427	10.03803	8.7615662	8.4416666
0.56	86.421364	71.740746	55.393925	42.106571	31.64852	23.190342	16.732431	12.692739	10.177909	8.8214884	8.4313326
0.57	86.431938	71.904083	55.671963	42.427559	31.987476	23.531998	17.011908	12.893738	10.324147	8.8868351	8.4269333
0.58	86.442024	72.062553	55.943752	42.742542	32.320126	23.869873	17.297424	13.093014	10.474356	8.9664469	8.4293642
0.59	86.451759	72.217079	56.209419	43.050983	32.646351	24.202602	17.580223	13.296058	10.621531	9.0441971	8.4348087
0.60	86.461884	72.367119	56.469978	43.353966	32.967213	24.531099	17.862698	13.49527	10.765951	9.120101	8.4423313
0.61	86.471336	72.513542	56.724747	43.651173	33.281952	24.853655	18.144804	13.695518	10.907637	9.2044449	8.4579058
0.62	86.480354	72.656372	56.974243	43.942745	33.59137	25.170986	18.425262	13.903366	11.046835	9.290534	8.4835501
0.63	86.489441	72.795441	57.2188	44.22905	33.895485	25.483082	18.705021	14.110231	11.187451	9.3810587	8.5211344
0.64	86.497932	72.930862	57.458267	44.510418	34.194344	25.790178	18.984198	14.313889	11.332636	9.4722004	8.5688715
0.65	86.506462	73.063026	57.693008	44.786568	34.488094	26.092098	19.260012	14.514197	11.485598	9.5658369	8.621789
0.66	86.514839	73.192307	57.923393	45.058155	34.777287	26.389393	19.531805	14.718089	11.645604	9.6657944	8.6734686
0.67	86.52298	73.318123	58.149029	45.324657	35.061485	26.681843	19.802334	14.920727	11.802715	9.7719011	8.7241926
0.68	86.531006	73.44133	58.370365	45.58696	35.341328	26.970055	20.072515	15.120665	11.95796	9.8833895	8.7738905
0.69	86.538429	73.561188	58.587677	45.84481	35.616653	27.253698	20.343111	15.320436	12.110472	9.992857	8.8222961
0.70	86.545845	73.678513	58.800724	46.098167	35.887497	27.532801	20.611149	15.529171	12.260736	10.101007	8.874754
0.71	86.553337	73.792816	59.009922	46.347801	36.15432	27.808189	20.878052	15.738893	12.410233	10.215351	8.9376898
0.72	86.560532	73.904938	59.215309	46.592857	36.416836	28.079306	21.144604	15.945116	12.561028	10.331007	9.0015221
0.73	86.567238	74.01461	59.41721	46.834476	36.675823	28.3465	21.407824	16.148272	12.710465	10.44991	9.0640593
0.74	86.574539	74.121834	59.615082	47.072071	36.930641	28.609936	21.667566	16.351282	12.865028	10.567147	9.1259165
0.75	86.58107	74.226677	59.809727	47.306122	37.181923	28.869579	21.924057	16.555063	13.017407	10.682758	9.1943846
0.76	86.587601	74.329285	60.000923	47.536327	37.429276	29.125456	22.176582	16.760071	13.173193	10.796571	9.2645283
0.77	86.594078	74.429893	60.18874	47.763157	37.673405	29.377865	22.425859	16.965815	13.328212	10.90858	9.3363104

0.78	86.600128	74.527969	60.373615	47.986347	37.913944	29.626844	22.674259	17.176579	13.481255	11.018997	9.4096498
0.79	86.606514	74.624374	60.555099	48.206501	38.151356	29.872728	22.920328	17.38592	13.633858	11.129639	9.4846163
0.80	86.612198	74.718628	60.733768	48.423542	38.385315	30.11517	23.163284	17.59499	13.792106	11.241343	9.5606298
0.81	86.618484	74.810852	60.909157	48.637047	38.616001	30.354269	23.40292	17.803293	13.952219	11.358924	9.6405487
0.82	86.623932	74.901505	61.081932	48.847507	38.843781	30.590063	23.640177	18.011871	14.110778	11.480655	9.7238178
0.83	86.629601	74.990387	61.251808	49.055157	39.068245	30.82312	23.875395	18.219294	14.267628	11.607538	9.8142557
0.84	86.635002	75.077286	61.419102	49.259743	39.289875	31.053392	24.108662	18.426521	14.422586	11.732778	9.9042444
0.85	86.640686	75.162727	61.583599	49.461338	39.508392	31.280376	24.33997	18.63283	14.576521	11.856564	9.9930134
0.86	86.646049	75.246605	61.745361	49.66037	39.724197	31.504601	24.568243	18.839157	14.733443	11.978906	10.080677
0.87	86.651199	75.328667	61.904816	49.856453	39.937149	31.726049	24.793911	19.044952	14.889256	12.099737	10.171458
0.88	86.65638	75.40921	62.061802	50.050129	40.147682	31.945009	25.017105	19.248505	15.043636	12.219226	10.266006
0.89	86.661293	75.488426	62.216228	50.240875	40.355236	32.161095	25.23749	19.449396	15.196209	12.337313	10.360738
0.90	86.666245	75.566078	62.368294	50.42939	40.56028	32.374516	25.455347	19.648962	15.351042	12.456547	10.457086
0.91	86.67099	75.642593	62.518253	50.615395	40.762955	32.585896	25.670752	19.84844	15.511893	12.575807	10.552345
0.92	86.675446	75.717239	62.66589	50.798782	40.963028	32.79425	25.883781	20.047783	15.673286	12.693775	10.646511
0.93	86.680321	75.79113	62.811272	50.980011	41.160854	33.000549	26.09454	20.247824	15.833017	12.816533	10.739397
0.94	86.684929	75.863434	62.95454	51.158646	41.356262	33.204407	26.302752	20.447197	15.990693	12.937914	10.831245
0.95	86.689438	75.934471	63.095848	51.33511	41.548988	33.405621	26.508547	20.644136	16.146589	13.05835	10.921856
0.96	86.693665	76.004509	63.235104	51.509346	41.739746	33.604664	26.712084	20.841169	16.302282	13.182505	11.011441
0.97	86.698097	76.073059	63.372437	51.681557	41.928448	33.801895	26.913799	21.038258	16.459223	13.305678	11.100737
0.98	86.702324	76.140709	63.507549	51.851566	42.114822	33.996624	27.113194	21.234154	16.615608	13.427369	11.191767
0.99	86.706474	76.207245	63.640873	52.019356	42.298985	34.189194	27.310135	21.427856	16.773157	13.547516	11.284151
1.00	86.710594	76.272621	63.772339	52.185028	42.481018	34.379574	27.505167	21.619724	16.9312	13.66979	11.380223
1.01	86.714828	76.336563	63.901978	52.349125	42.661263	34.568161	27.698483	21.809847	17.0924	13.795389	11.479379

Run 4

q\r	1	11	21	31	41	51	61	71	81	91	101
0.01	13.998676	13.334585	25.088985	29.732632	31.789677	32.848583	33.457787	33.83812	34.090687	34.266628	34.393963
0.02	12.877877	9.8864059	19.036577	25.714996	29.094452	30.957954	32.07251	32.785095	33.265675	33.604031	33.850754
0.03	12.208158	9.4772177	14.940596	22.506023	26.764118	29.247852	30.784586	31.788513	32.475388	32.963875	33.322647
0.04	11.778276	9.5006571	12.695521	19.871458	24.724615	27.691841	29.583401	30.843632	31.71755	32.344975	32.80901
0.05	11.483071	9.5659342	11.373405	17.659901	22.920984	26.268427	28.459709	29.946163	30.989973	31.746147	32.30912
0.06	11.272314	9.6966686	10.6182	15.78689	21.312857	24.960335	27.405853	29.092447	30.290819	31.166426	31.822481
0.07	11.12001	9.8786526	10.011553	14.343093	19.867264	23.753006	26.414978	28.279119	29.618313	30.604822	31.348518
0.08	11.007483	10.09806	9.6867161	13.281921	18.558891	22.634399	25.481079	27.502998	28.97076	30.060371	30.886642
0.09	10.922688	10.309299	9.5721645	12.489166	17.367653	21.594662	24.59902	26.761461	28.346685	29.532238	30.436335
0.10	10.855832	10.538486	9.4995337	11.813956	16.277634	20.624969	23.764397	26.052191	27.744873	29.019758	29.997274
0.11	10.799218	10.735465	9.4767113	11.333063	15.316894	19.717981	22.973238	25.372913	27.164009	28.522116	29.568941
0.12	10.751606	10.908312	9.4809637	10.96143	14.510534	18.86726	22.222143	24.721577	26.602875	28.038597	29.150852
0.13	10.711614	11.086427	9.4951448	10.641374	13.857275	18.067274	21.507729	24.09638	26.060455	27.568651	28.742752
0.14	10.678238	11.266993	9.5009651	10.339728	13.278728	17.31332	20.827124	23.495625	25.535686	27.111551	28.344086
0.15	10.648991	11.454434	9.5039053	10.070355	12.800348	16.60104	20.177723	22.917761	25.0277	26.666794	27.95462
0.16	10.623947	11.635574	9.5183325	9.8653545	12.384408	15.934725	19.557379	22.361662	24.535589	26.233801	27.573935
0.17	10.602556	11.811697	9.5370522	9.731164	11.989533	15.328768	18.963877	21.825813	24.058598	25.812075	27.201756
0.18	10.583963	11.981634	9.5604649	9.6401119	11.651761	14.777313	18.395569	21.309128	23.595978	25.401245	26.837742
0.19	10.567241	12.142117	9.5826635	9.5966234	11.386519	14.319921	17.85078	20.810579	23.14724	25.000898	26.481838
0.20	10.552342	12.302546	9.6139708	9.5562668	11.168982	13.903768	17.327612	20.328791	22.711329	24.610315	26.133352
0.21	10.539469	12.453796	9.6577606	9.5144558	10.959461	13.514215	16.825237	19.863354	22.288267	24.229551	25.792458
0.22	10.52762	12.594007	9.7030249	9.4976759	10.774182	13.166025	16.341848	19.412956	21.876873	23.857752	25.458485
0.23	10.516348	12.726585	9.7529449	9.4868078	10.595398	12.858388	15.886759	18.97699	21.476776	23.494823	25.131403
0.24	10.505883	12.850012	9.8015833	9.4765606	10.422384	12.585077	15.461877	18.554924	21.087753	23.14053	24.811142
0.25	10.496017	12.965717	9.8507643	9.4776649	10.254659	12.320999	15.059336	18.14563	20.708885	22.794333	24.497225
0.26	10.486724	13.076173	9.9062557	9.4796305	10.103469	12.065605	14.693951	17.748875	20.340105	22.456259	24.189583
0.27	10.47844	13.181631	9.9645987	9.4874763	9.9706039	11.827985	14.374078	17.363705	19.980629	22.125574	23.887852
0.28	10.470624	13.282902	10.025548	9.4933777	9.8641224	11.623249	14.079741	16.989933	19.630489	21.802387	23.592134
0.29	10.463128	13.376603	10.088329	9.4979343	9.7772465	11.446025	13.794009	16.626511	19.288801	21.486059	23.301849
0.30	10.456092	13.463021	10.14789	9.5004129	9.7157021	11.300847	13.525214	16.273619	18.955776	21.176771	23.017288
0.31	10.449416	13.542962	10.205938	9.501112	9.6600952	11.161535	13.276227	15.937704	18.630596	20.873837	22.737875
0.32	10.443094	13.618971	10.262619	9.5007381	9.6237183	11.025435	13.045073	15.621267	18.313391	20.577425	22.463882
0.33	10.436881	13.689625	10.32185	9.5050583	9.60186	10.897722	12.839702	15.312938	18.003485	20.287106	22.194851
0.34	10.431181	13.755533	10.388544	9.5120516	9.5792933	10.778903	12.648029	15.022272	17.70063	20.002556	21.930513
0.35	10.425648	13.816985	10.452143	9.519104	9.5559063	10.662641	12.461092	14.748267	17.404987	19.723995	21.671116
0.36	10.420344	13.87387	10.51241	9.5265284	9.5319214	10.548771	12.278387	14.507027	17.115694	19.450724	21.416058
0.37	10.415305	13.926926	10.569991	9.536644	9.5099897	10.437253	12.100045	14.281322	16.833023	19.183016	21.16568
0.38	10.410437	13.976614	10.626024	9.5455904	9.5001354	10.327992	11.928243	14.065243	16.55645	18.920502	20.9196

0.39	10.405773	14.02241	10.679684	9.5579395	9.4947853	10.221656	11.772655	13.854079	16.285986	18.663109	20.677761
0.40	10.401319	14.064908	10.73087	9.569047	9.4884157	10.125278	11.630742	13.64766	16.023563	18.410208	20.439707
0.41	10.397237	14.103951	10.779881	9.5790195	9.4811697	10.035883	11.500486	13.457663	15.777966	18.162374	20.205925
0.42	10.393316	14.14018	10.827415	9.5902586	9.476572	9.9536562	11.3923	13.275724	15.537239	17.919107	19.976007
0.43	10.389524	14.173326	10.873747	9.6030931	9.476449	9.8858891	11.292811	13.104559	15.301462	17.680313	19.749842
0.44	10.385811	14.203964	10.921116	9.6214466	9.4780083	9.8229294	11.195205	12.941527	15.076837	17.445633	19.527102
0.45	10.382221	14.231987	10.970472	9.6419926	9.478632	9.7714806	11.099248	12.797885	14.862216	17.215345	19.308245
0.46	10.378925	14.257819	11.018101	9.6626101	9.4821987	9.7316036	11.004748	12.656673	14.664824	16.98897	19.092606
0.47	10.375583	14.281151	11.067442	9.6820946	9.4867344	9.693696	10.915322	12.517973	14.480559	16.766636	18.880543
0.48	10.372437	14.302738	11.11727	9.7043381	9.490201	9.6590509	10.831827	12.381842	14.307741	16.548311	18.671789
0.49	10.36931	14.322391	11.166778	9.7262955	9.493412	9.6328535	10.749454	12.247887	14.140695	16.333378	18.466047
0.50	10.366412	14.340282	11.215912	9.7501278	9.4962101	9.6168165	10.668369	12.116419	13.976752	16.122335	18.263708
0.51	10.363457	14.356445	11.265674	9.7729187	9.4984465	9.6027374	10.588375	11.986978	13.815452	15.923049	18.06431
0.52	10.360632	14.370894	11.316446	9.7949657	9.4999571	9.5882311	10.509678	11.866548	13.656924	15.729667	17.867977
0.53	10.357935	14.383887	11.368015	9.8169432	9.5007963	9.5733995	10.432001	11.754612	13.51018	15.539193	17.674341
0.54	10.35531	14.395435	11.420099	9.839262	9.50105	9.5582752	10.355515	11.649651	13.366029	15.351916	17.483768
0.55	10.352678	14.405827	11.471145	9.8629837	9.5009918	9.5429106	10.280011	11.550983	13.231269	15.169188	17.296009
0.56	10.350246	14.414664	11.52191	9.8892555	9.5007477	9.5272665	10.207639	11.462923	13.100131	14.997379	17.110863
0.57	10.347761	14.422674	11.571028	9.914897	9.503458	9.5125217	10.140273	11.384841	12.973554	14.827993	16.928217
0.58	10.345469	14.429502	11.620016	9.9409513	9.5056133	9.5047741	10.075205	11.311218	12.860212	14.673484	16.74823
0.59	10.343176	14.434984	11.669788	9.9692135	9.5098791	9.4994373	10.015317	11.238866	12.75075	14.527152	16.570887
0.60	10.340921	14.43998	11.718169	9.9977436	9.5143814	9.4961205	9.9581699	11.167296	12.64275	14.385918	16.396051
0.61	10.338772	14.444147	11.765748	10.025085	9.5183325	9.4922123	9.9102077	11.096663	12.536224	14.252282	16.223581
0.62	10.33666	14.447486	11.813778	10.054145	9.5217772	9.4879599	9.8631697	11.02677	12.431023	14.120504	16.05431
0.63	10.3346	14.450178	11.861875	10.08278	9.5267029	9.4833755	9.8207016	10.957758	12.32732	13.990581	15.895259
0.64	10.332653	14.451817	11.908686	10.110822	9.5327253	9.4784546	9.7833338	10.895075	12.22503	13.862484	15.738445
0.65	10.33065	14.452929	11.95416	10.137911	9.538125	9.476491	9.7530565	10.833546	12.12408	13.736053	15.58359
0.66	10.328758	14.453238	11.999511	10.164263	9.5431738	9.47565	9.7265425	10.772599	12.024323	13.613808	15.430671
0.67	10.3269	14.45295	12.044215	10.191152	9.5494452	9.4769955	9.700057	10.712373	11.928513	13.498055	15.279805
0.68	10.325054	14.451924	12.088364	10.217711	9.5564041	9.4779406	9.6738176	10.652721	11.837465	13.383652	15.134227
0.69	10.323212	14.450377	12.131514	10.243856	9.563098	9.4785395	9.6526785	10.59381	11.753055	13.275173	14.995036
0.70	10.321531	14.448228	12.176074	10.269341	9.5692406	9.478694	9.6346722	10.53552	11.671258	13.17056	14.857474
0.71	10.319822	14.44556	12.220947	10.295841	9.5750713	9.4816618	9.6220694	10.477801	11.59504	13.067092	14.727215
0.72	10.318174	14.4421	12.264515	10.32457	9.5804977	9.4847136	9.6122236	10.420731	11.519957	12.968065	14.607614
0.73	10.316488	14.438554	12.307005	10.35567	9.5869341	9.4873791	9.6022701	10.364257	11.455674	12.878284	14.48947
0.74	10.314928	14.434433	12.349598	10.385906	9.5939884	9.4896727	9.5921879	10.308388	11.395525	12.791178	14.376235
0.75	10.313396	14.429761	12.391124	10.415547	9.6011467	9.4916725	9.5820065	10.253017	11.338708	12.704958	14.267655
0.76	10.311804	14.424741	12.431897	10.444339	9.6114588	9.4937897	9.5715847	10.200894	11.282704	12.619738	14.160449
0.77	10.310479	14.419392	12.472238	10.472637	9.6218824	9.4956989	9.5609818	10.151196	11.227293	12.535529	14.054472

0.78	10.308907	14.413617	12.511537	10.50017	9.6335211	9.4972382	9.5502949	10.101963	11.172367	12.452155	13.949639
0.79	10.307548	14.407423	12.549901	10.527314	9.6454334	9.4985437	9.5394535	10.056333	11.117908	12.369586	13.845784
0.80	10.306065	14.40107	12.587345	10.553687	9.6572237	9.4995413	9.5284882	10.012167	11.063918	12.287867	13.743099
0.81	10.304764	14.394304	12.624575	10.579652	9.6686659	9.5002232	9.5174503	9.9693155	11.01049	12.207083	13.642387
0.82	10.303436	14.387125	12.661085	10.605687	9.6797895	9.5007648	9.5095434	9.9325123	10.95757	12.127232	13.548111
0.83	10.302008	14.379789	12.697051	10.630892	9.6920557	9.5009756	9.5045843	9.8976278	10.909277	12.048134	13.454683
0.84	10.300741	14.372282	12.732759	10.655737	9.7048159	9.5011749	9.5003834	9.8629589	10.861756	11.969776	13.362202
0.85	10.299506	14.364377	12.767522	10.679938	9.7171831	9.500967	9.4981508	9.8314028	10.814675	11.89692	13.27499
0.86	10.298263	14.356365	12.801589	10.703863	9.7303944	9.5006266	9.4956627	9.8013868	10.768007	11.825459	13.190027
0.87	10.297049	14.348165	12.834515	10.727051	9.7440138	9.5014153	9.4929628	9.7764797	10.721712	11.758964	13.105803
0.88	10.295793	14.339635	12.867273	10.749912	9.7573156	9.5031023	9.4900694	9.7542191	10.675734	11.693027	13.02235
0.89	10.294611	14.331037	12.899253	10.772289	9.7702141	9.5046263	9.4869928	9.7345772	10.630211	11.631946	12.944431
0.90	10.293507	14.322201	12.930401	10.794154	9.7829275	9.5058441	9.483737	9.7150126	10.585095	11.571956	12.873014
0.91	10.292354	14.313079	12.962127	10.816006	9.7954893	9.5090036	9.4803247	9.6954498	10.540246	11.513119	12.802205
0.92	10.291256	14.303967	12.993296	10.837664	9.8079805	9.5119257	9.4773865	9.6759558	10.495877	11.462549	12.732098
0.93	10.290091	14.294576	13.023874	10.859004	9.8206539	9.5148878	9.4765196	9.6595182	10.451778	11.413568	12.662576
0.94	10.289106	14.285222	13.05388	10.879848	9.833148	9.5173883 ⭐	9.4754	9.6457491	10.408071	11.368397	12.593699
0.95	10.287978	14.275576	13.083106	10.901585	9.8467388	9.519783 ⭐	9.47627	9.6324215	10.364773	11.323633	12.525493
0.96	10.286923	14.265753	13.111893	10.923677	9.8601904	9.5219469	9.4771748	9.6234608	10.321766	11.279366	12.457824
0.97	10.285908	14.255927	13.141118	10.946608	9.8748913	9.5248661	9.4778128	9.6161652	10.279068	11.235363	12.390668
0.98	10.284852	14.245948	13.169947	10.969096	9.8899364	9.5287991	9.478281	9.6088915	10.236724	11.191747	12.32414
0.99	10.28388	14.235831	13.198588	10.991259	9.9046898	9.5325108	9.4785776	9.6015491	10.197701	11.148501	12.258271
1.00	10.282919	14.225614	13.226772	11.012924	9.9190521	9.5361023	9.4786501	9.5940857	10.159496	11.105498	12.192896
1.01	10.281934	14.215349	13.254583	11.034977	9.9339399	9.5396061	9.4804096	9.5865889	10.121531	11.062781	12.128018

Run 5

q\r	1	11	21	31	41	51	61	71	81	91	101
0.01	18.502064	15.518713	23.216028	27.76428	29.842701	30.926031	31.553186	31.946129	32.207664	32.390129	32.522305
0.02	20.920944	14.792185	17.9202	23.816317	27.126877	28.998074	30.131199	30.860834	31.355114	31.704161	31.959213
0.03	22.23563	14.586952	16.088444	20.787144	24.8318	27.279741	28.822723	29.841499	30.543236	31.0445	31.413834
0.04	23.075258	14.34212	15.351128	18.465534	22.86861	25.739183	27.614893	28.88241	29.769224	30.40971	30.885384
0.05	23.664455	14.133249	15.066381	17.126726	21.171982	24.350548	26.496595	27.978359	29.030493	29.798361	30.373041
0.06	24.111555	13.953503	14.924431	16.406979	19.693132	23.093193	25.458553	27.124874	28.324717	29.209232	29.876123
0.07	24.460968	13.809666	14.807504	15.87378	18.462076	21.949764	24.492558	26.317934	27.649817	28.641171	29.393984
0.08	24.743759	13.695372	14.755629	15.500108	17.621981	20.90596	23.591503	25.553787	27.003748	28.093014	28.925907
0.09	24.983988	13.605204	14.707449	15.302547	17.004045	19.949743	22.749216	24.829193	26.384747	27.563717	28.471266
0.10	25.18952	13.539636	14.648019	15.139367	16.585453	19.087145	21.960482	24.141443	25.791306	27.052523	28.029655
0.11	25.366728	13.491532	14.582541	15.059172	16.232243	18.353912	21.220545	23.487757	25.22187	26.558403	27.600437
0.12	25.521862	13.465084	14.515152	14.988347	15.932172	17.814314	20.525143	22.865725	24.674967	26.080517	27.183065
0.13	25.659576	⭐ 13.4518	14.447225	14.928503	15.699219	17.320799	19.870516	22.273252	24.149437	25.618225	26.777205
0.14	25.785002	13.456236	14.380153	14.873229	15.498989	16.982304	19.255096	21.708233	23.643919	25.170601	26.382174
0.15	25.897724	13.478327	14.316079	14.816763	15.375633	16.699387	18.717346	21.169003	23.157518	24.737143	25.997778
0.16	25.999897	13.522385	14.257433	14.789679	15.277657	16.461966	18.247131	20.653774	22.688995	24.317051	25.623375
0.17	26.093115	13.569442	14.200517	14.765707	15.182632	16.236626	17.874697	20.161133	22.23757	23.909819	25.258762
0.18	26.178493	13.618554	14.145288	14.744703	15.110849	16.031799	17.520376	19.689697	21.802296	23.514877	24.903492
0.19	26.257162	13.669277	14.091985	14.722478	15.068748	15.865465	17.213379	19.241365	21.382656	23.13192	24.557474
0.20	26.329945	13.721394	14.04077	14.697433	15.028748	15.715896	16.988031	18.841957	20.977238	22.759972	24.21987
0.21	26.397545	13.774237	13.99263	14.670506	14.987964	15.578379	16.786819	18.458906	20.586115	22.399107	23.890949
0.22	26.460499	13.827948	13.947708	14.641953	14.951091	15.462726	16.608093	18.156298	20.207882	22.048294	23.569811
0.23	26.519487	13.881968	13.905397	14.612182	14.920389	15.389091	16.444096	17.882868	19.842257	21.707447	23.256519
0.24	26.57468	13.936006	13.865503	14.581639	14.88888	15.325182	16.286135	17.619509	19.488911	21.376377	22.950993
0.25	26.626663	13.990326	13.827942	14.550795	14.856784	15.262554	16.133688	17.365295	19.156784	21.054268	22.652576
0.26	26.675648	14.044383	13.792793	14.519813	14.824294	15.201084	16.000053	17.167109	18.851357	20.741148	22.3613
0.27	26.721943	14.098181	13.759781	14.488586	14.80179	15.142511	15.884903	17.002485	18.555111	20.436197	22.076601
0.28	26.76568	14.157406	13.728788	14.457479	14.78957	15.105502	15.779173	16.852751	18.293358	20.139673	21.798712
0.29	26.807251	14.216118	13.699664	14.426378	14.776274	15.078865	15.676464	16.708353	18.078278	19.850769	21.526987
0.30	26.846752	14.276717	13.672533	14.395629	14.76171	15.053318	15.582019	16.584068	17.869646	19.569628	21.261587
0.31	26.884354	14.339689	13.646997	14.365194	14.75026	15.027328	15.4981	16.46306	17.66663	19.295475	21.001886
0.32	26.9202	14.406187	13.623237	14.335217	14.738478	15.000984	15.430297	16.345407	17.469345	19.047726	20.748182
0.33	26.954445	14.472445	13.600928	14.307411	14.725687	14.974508	15.384764	16.230778	17.288284	18.807571	20.500029
0.34	26.987133	14.537593	13.580941	14.280259	14.711924	14.951878	15.340025	16.11916	17.14521	18.573162	20.257105
0.35	27.018488	14.603329	13.563141	14.253854	14.697253	14.93218	15.295892	16.021057	17.019035	18.358089	20.019619
0.36	27.048525	14.670122	13.546566	14.22758	14.681965	14.912054	15.252313	15.930985	16.903093	18.179989	19.786974
0.37	27.077358	14.736103	13.531397	14.201722	14.666108	14.891628	15.209404	15.851605	16.78993	18.011911	19.559484
0.38	27.105036	14.801214	13.5173	14.17612	14.650039	14.871012	15.167096	15.773956	16.683683	17.847599	19.336864

0.39	27.131706	14.865728	13.504427	14.150969	14.63332	14.850158	15.132096	15.69806	16.588409	17.686893	19.13097
0.40	27.157335	14.929166	13.492557	14.12609	14.616224	14.829113	15.106929	15.623718	16.495008	17.52948	18.933649
0.41	27.182085	14.991503	13.482127	14.101755	14.598818	14.810718	15.087418	15.55964	16.40366	17.375685	18.740538
0.42	27.205898	15.053328	13.475101	14.077765	14.581367	14.799703	15.069754	15.497929	16.314148	17.245306	18.551367
0.43	27.228903	15.116541	13.469007	14.054318	14.563711	14.792149	15.051847	15.445292	16.226511	17.135996	18.376017
0.44	27.251154	15.178675	13.463775	14.031236	14.546082	14.783946	15.033714	15.408321	16.14056	17.035316	18.228079
0.45	27.272625	15.239762	13.459389	14.008999	14.528312	14.775109	15.01539	15.375048	16.061434	16.943037	18.090498
0.46	27.293465	15.299857	13.455694	13.987597	14.510457	14.765719	14.996939	15.342041	15.988758	16.852331	17.955235
0.47	27.313599	15.361195	13.452709	13.966754	14.492737	14.756933	14.978402	15.309363	15.922072	16.763359	17.822622
0.48	27.333122	15.422772	⭐ 13.4516	13.946568	14.474876	14.750017	14.960559	15.277021	15.861067	16.680801	17.692329
0.49	27.352074	15.483377	⭐ 13.4526	13.926823	14.457051	14.742556	14.946951	15.245042	15.801051	16.605093	17.564207
0.50	27.370396	15.542989	13.454139	13.907673	14.439226	14.734558	14.933152	15.213391	15.742141	16.530745	17.438486
0.51	27.388201	15.601632	13.456161	13.888991	14.421618	14.726175	14.919161	15.182012	15.684182	16.457628	17.319315
0.52	27.405527	15.660535	13.460512	13.870847	14.40399	14.717359	14.904994	15.151056	15.627237	16.385698	17.221195
0.53	27.422321	15.720075	13.465934	13.853164	14.386455	14.708214	14.890677	15.129016	15.577032	16.314859	17.133282
0.54	27.438772	15.778681	13.47186	13.835923	14.369067	14.698709	14.876257	15.110476	15.529266	16.245245	17.050602
0.55	27.454578	15.836319	13.482154	13.819361	14.351905	14.689065	14.861741	15.094388	15.482458	16.17679	16.975382
0.56	27.470119	15.893107	13.494004	13.803185	14.334779	14.679051	14.847135	15.081467	15.444184	16.109419	16.901182
0.57	27.485191	15.948979	13.506098	13.787445	14.318844	14.668752	14.832379	15.0684	15.415185	16.049635	16.828035
0.58	27.499783	16.003981	13.518473	13.772141	14.303027	14.658356	14.817684	15.055247	15.389524	15.992113	16.755978
0.59	27.514133	16.058256	13.531034	13.757304	14.287436	14.647779	14.80803	15.041894	15.364068	15.937768	16.688732
0.60	27.528137	16.111637	13.543843	13.743013	14.272358	14.637074	14.800104	15.028425	15.338776	15.889141	16.627188
0.61	27.541805	16.164238	13.55684	13.729003	14.25723	14.626123	14.794971	15.014906	15.313699	15.841173	16.566435
0.62	27.554981	16.216091	13.569994	13.715416	14.242147	14.615017	14.789455	15.001292	15.288731	15.793806	16.506428
0.63	27.56798	16.267206	13.583304	13.702214	14.227194	14.603786	14.78364	14.987606	15.26403	15.7471	16.447273
0.64	27.580671	16.317574	13.596746	13.689502	14.212403	14.592541	14.777456	14.973852	15.239523	15.701081	16.388969
0.65	27.593012	16.367207	13.610308	13.677072	14.197623	14.58116	14.771054	14.960796	15.215207	15.65565	16.331387
0.66	27.605062	16.416185	13.624015	13.66502	14.182972	14.569748	14.764539	14.950862	15.191097	15.611355	16.27453
0.67	27.616941	16.464434	13.637799	13.653296	14.168426	14.558381	14.758059	14.940723	15.167182	15.573021	16.218452
0.68	27.628477	16.512075	13.651704	13.641971	14.154013	14.546963	14.753346	14.930496	15.144531	15.535111	16.163048
0.69	27.639767	16.558979	13.665733	13.63105	14.139836	14.535563	14.74835	14.920135	15.128753	15.497732	16.108503
0.70	27.650848	16.605333	13.679931	13.620402	14.125688	14.524071	14.743105	14.909674	15.114487	15.464111	16.059807
0.71	27.661631	16.651117	13.694211	13.610062	14.111685	14.512537	14.737574	14.899147	15.100249	15.435843	16.012903
0.72	27.672243	16.696659	13.708584	13.600055	14.09781	14.501014	14.731827	14.888566	15.090319	15.414022	15.966635
0.73	27.682657	16.741621	13.722961	13.590492	14.08418	14.489542	14.72583	14.877912	15.080357	15.39371	15.925058
0.74	27.692776	16.786028	13.737412	13.581788	14.070578	14.477987	14.719707	14.867202	15.070313	15.37348	15.885653
0.75	27.702751	16.829855	13.751875	13.573353	14.05716	14.466454	14.713593	14.856474	15.060188	15.353365	15.846705
0.76	27.712479	16.873081	13.766429	13.565245	14.043894	14.454926	14.707117	14.845689	15.050005	15.333369	15.808178
0.77	27.722065	16.915785	13.780982	13.557486	14.030855	14.443505	14.700466	14.834819	15.039731	15.313514	15.770096

0.78	27.731453	16.958065	13.795715	13.549944	14.017866	14.432036	14.693689	14.823964	15.029389	15.293777	15.732401
0.79	27.74062	17.001244	13.810446	13.542626	14.005448	14.420588	14.686827	14.814138	15.019005	15.274125	15.695084
0.80	27.749666	17.043894	13.825234	13.535633	13.993206	14.409184	14.67981	14.807554	15.008553	15.254592	15.658176
0.81	27.758469	17.086014	13.84006	13.528877	13.981129	14.397808	14.672673	14.801582	14.998077	15.235198	15.621693
0.82	27.767199	17.127661	13.854878	13.522408	13.969314	14.386462	14.665424	14.797407	14.987563	15.215976	15.589584
0.83	27.775772	17.168787	13.869688	13.516144	13.957609	14.375216	14.65816	14.793497	14.977029	15.196839	15.558645
0.84	27.784136	17.209368	13.884527	13.510117	13.946098	14.364202	14.650885	14.789467	14.966475	15.177836	15.528019
0.85	27.79232	17.249565	13.899349	13.504313	13.934753	14.353034	14.643386	14.785172	14.957764	15.158928	15.497661
0.86	27.800486	17.289225	13.914195	13.498823	13.923692	14.341908	14.635818	14.780726	14.950075	15.141943	15.469995
0.87	27.808382	17.32843	13.929006	13.493451	13.912666	14.331141	14.628162	14.776156	14.942306	15.129767	15.445624
0.88	27.81617	17.368639	13.943931	13.488297	13.901855	14.320856	14.620469	14.771389	14.934442	15.118443	15.425538
0.89	27.823851	17.408655	13.958871	13.483385	13.891181	14.310606	14.612645	14.766516	14.926535	15.10718	15.408952
0.90	27.831442	17.448175	13.97373	13.479941	13.88077	14.300426	14.604822	14.761506	14.91857	15.097168	15.392491
0.91	27.838791	17.48724	13.988616	13.476724	13.870434	14.290265	14.596894	14.757171	14.910518	15.089323	15.376043
0.92	27.846088	17.525883	14.00348	13.473698	13.860272	14.280431	14.588945	14.753694	14.902449	15.081439	15.359699
0.93	27.853306	17.564098	14.018309	13.470882	13.850264	14.270808	14.581094	14.750152	14.894375	15.073509	15.343419
0.94	27.860353	17.601851	14.033141	13.468207	13.840389	14.261008	14.57313	14.746395	14.886215	15.065494	15.327219
0.95	27.867283	17.639215	14.047932	13.465814	13.83074	14.25123	14.565162	14.742452	14.878032	15.05746	15.311134
0.96	27.874132	17.676186	14.062674	13.463518	13.821259	14.241529	14.55723	14.738361	14.869819	15.049365	15.295113
0.97	27.880873	17.712721	14.077431	13.461398	13.811922	14.231839	14.549253	14.734181	14.861548	15.041224	15.279162
0.98	27.887463	17.748837	14.092081	13.459484	13.802725	14.222208	14.541224	14.72985	14.85325	15.033059	15.263297
0.99	27.894007	17.784603	14.107822	13.457695	13.793764	14.212568	14.533224	14.725441	14.844968	15.024848	15.247542
1.00	27.90041	17.819935	14.124084	13.456073	13.784865	14.203042	14.525158	14.720901	14.836623	15.016602	15.231807
1.01	27.906717	17.854961	14.140363	13.454567	13.776119	14.193665	14.517219	14.716368	14.828316	15.008328	15.216174

Appendix II

-

Kalman filter experiment

Overall data

q\r	1	11	21	31	41	51	61	71	81	91	101
0.01	0.0077331	0.0029068	0.0038709	0.0046541	0.005047	0.0052788	0.0054132	0.005497	0.0055527	0.0055915	0.0056196
0.02	0.0080898	0.0032785	0.0031331	0.0039668	0.0045397	0.0048816	0.0051076	0.0052648	0.0053708	0.0054454	0.0054998
0.03	0.0082498	0.0037145	0.0029187	0.0034993	0.0041356	0.004567	0.0048487	0.0050468	0.0051964	0.0053042	0.0053833
0.04	0.0083452	0.0040523	0.0029261	0.0032019	0.0038161	0.0042933	0.0046272	0.0048599	0.005032	0.0051676	0.00527
0.05	0.0084098	0.0043143	0.0030124	0.0030342	0.0035563	0.0040548	0.0044274	0.004693	0.0048878	0.0050379	0.0051597
0.06	0.0084578	0.0045302	0.0031287	0.0029466	0.0033515	0.0038515	0.0042438	0.0045393	0.0047563	0.0049218	0.0050539
0.07	0.0084954	0.0047161	0.0032431	0.0029079	0.0032015	0.0036728	0.0040783	0.0043956	0.0046335	0.0048148	0.0049577
0.08	0.0085259	0.0048796	0.0033671	0.0029076	0.0030965	0.0035168	0.0039306	0.0042605	0.0045177	0.0047139	0.0048681
0.09	0.0085517	0.0050238	0.0034937	0.0029348	0.0030182	0.0033837	0.0037973	0.0041352	0.0044075	0.004618	0.0047834
0.10	0.0085737	0.0051551	0.0036117	0.0029697	0.0029657	0.0032763	0.0036744	0.0040202	0.0043025	0.0045264	0.0047023
0.11	0.0085925	0.0052724	0.0037215	0.0030179	0.0029303	0.0031879	0.0035635	0.003914	0.0042024	0.0044383	0.0046246
0.12	0.0086089	0.0053785	0.0038236	0.0030711	0.00291	0.0031199	0.0034612	0.0038157	0.0041089	0.0043536	0.0045497
0.13	0.0086233	0.0054768	0.0039181	0.0031248	0.002905	0.0030599	0.0033736	0.0037226	0.0040216	0.0042719	0.0044773
0.14	0.0086364	0.0055685	0.0040052	0.003177	0.0029076	0.0030154	0.0032976	0.0036363	0.003939	0.0041935	0.004407
0.15	0.0086481	0.0056553	0.0040857	0.0032293	0.0029216	0.0029798	0.0032317	0.0035558	0.0038616	0.0041194	0.004339
0.16	0.0086587	0.0057372	0.0041615	0.003285	0.0029393	0.002952	0.0031742	0.0034797	0.0037878	0.0040493	0.0042728
0.17	0.0086685	0.0058139	0.0042328	0.0033423	0.0029589	0.0029307	0.0031274	0.0034111	0.0037171	0.0039821	0.0042089
0.18	0.0086773	0.005886	0.0042999	0.0033999	0.0029824	0.0029147	0.0030843	0.0033511	0.0036504	0.0039183	0.0041478
0.19	0.0086855	0.0059536	0.0043629	0.0034587	0.0030106	0.0029076	0.0030457	0.0032958	0.0035875	0.0038576	0.0040891
0.20	0.0086931	0.0060181	0.0044232	0.0035154	0.0030412	0.0029052	0.0030162	0.0032467	0.0035274	0.003799	0.0040332
0.21	0.0087002	0.0060789	0.0044816	0.0035696	0.0030714	0.002905	0.0029905	0.0032011	0.00347	0.0037421	0.003979
0.22	0.0087067	0.0061362	0.0045375	0.0036227	0.0031022	0.0029096	0.0029684	0.0031626	0.0034174	0.0036879	0.0039271
0.23	0.0087128	0.0061906	0.0045912	0.0036738	0.0031326	0.0029192	0.0029502	0.0031285	0.00337	0.0036362	0.0038773
0.24	0.0087185	0.006242	0.0046424	0.0037229	0.0031628	0.0029308	0.0029353	0.0030962	0.0033263	0.0035866	0.0038291
0.25	0.0087238	0.0062909	0.0046915	0.0037708	0.0031918	0.0029421	0.0029221	0.0030656	0.0032851	0.0035388	0.0037821
0.26	0.0087288	0.0063376	0.0047391	0.0038168	0.003222	0.0029548	0.0029132	0.0030396	0.0032479	0.0034925	0.0037362
0.27	0.0087336	0.0063822	0.0047851	0.0038615	0.0032531	0.0029687	0.0029082	0.003018	0.0032124	0.0034488	0.0036923
0.28	0.0087381	0.0064251	0.0048296	0.0039044	0.0032854	0.0029849	0.0029061	0.0029987	0.0031803	0.0034083	0.0036499
0.29	0.0087423	0.0064661	0.0048729	0.0039457	0.0033179	0.0030028	0.0029047	0.002981	0.0031529	0.0033711	0.0036091
0.30	0.0087463	0.0065053	0.0049145	0.0039856	0.003351	0.0030225	0.0029049	0.0029655	0.0031268	0.0033362	0.0035695
0.31	0.0087501	0.0065429	0.0049547	0.0040239	0.0033837	0.0030422	0.0029076	0.0029521	0.0031018	0.0033027	0.0035311
0.32	0.0087538	0.0065792	0.0049934	0.0040608	0.0034172	0.0030618	0.0029122	0.0029409	0.0030781	0.0032716	0.0034936
0.33	0.0087572	0.006614	0.0050312	0.0040968	0.0034509	0.0030815	0.00292	0.0029302	0.0030556	0.0032426	0.0034577
0.34	0.0087605	0.0066473	0.0050685	0.0041319	0.0034838	0.0031013	0.0029281	0.002921	0.0030367	0.0032146	0.003424
0.35	0.0087637	0.0066794	0.0051047	0.004166	0.0035159	0.0031212	0.002936	0.0029141	0.0030202	0.0031884	0.0033926
0.36	0.0087667	0.0067103	0.0051398	0.0041991	0.0035472	0.0031407	0.0029441	0.00291	0.0030051	0.0031656	0.0033632
0.37	0.0087696	0.0067402	0.0051737	0.0042314	0.0035777	0.0031602	0.002953	0.0029073	0.0029909	0.0031446	0.003335
0.38	0.0087724	0.0067689	0.0052066	0.0042626	0.0036081	0.003179	0.0029625	0.002906	0.0029778	0.003124	0.0033077

0.39	0.0087751	0.0067966	0.0052386	0.0042932	0.0036381	0.0031979	0.0029728	0.002905	0.002966	0.0031042	0.0032818
0.40	0.0087777	0.0068232	0.0052696	0.0043228	0.0036672	0.0032175	0.0029842	0.0029042	0.0029553	0.0030853	0.0032578
0.41	0.0087802	0.0068489	0.0052997	0.0043515	0.0036956	0.0032374	0.0029963	0.0029056	0.0029463	0.0030668	0.0032345
0.42	0.0087826	0.0068738	0.0053292	0.0043796	0.0037235	0.0032577	0.0030097	0.0029077	0.002938	0.0030499	0.003212
0.43	0.0087849	0.0068979	0.0053578	0.0044072	0.003751	0.0032786	0.0030236	0.0029108	0.0029298	0.0030354	0.0031907
0.44	0.0087871	0.0069212	0.0053859	0.0044345	0.003778	0.0032995	0.0030375	0.0029156	0.0029226	0.0030223	0.0031717
0.45	0.0087893	0.0069438	0.0054135	0.0044615	0.0038043	0.0033206	0.0030511	0.0029218	0.0029167	0.0030103	0.0031544
0.46	0.0087914	0.0069656	0.0054404	0.0044878	0.0038303	0.003342	0.0030648	0.0029278	0.0029127	0.0029987	0.0031375
0.47	0.0087934	0.0069868	0.0054669	0.0045135	0.0038558	0.0033632	0.0030786	0.0029336	0.0029096	0.0029877	0.0031209
0.48	0.0087954	0.0070074	0.0054929	0.004539	0.0038806	0.0033845	0.0030924	0.0029394	0.0029075	0.0029775	0.0031049
0.49	0.0087973	0.0070274	0.0055183	0.0045638	0.0039049	0.003406	0.0031064	0.0029455	0.0029064	0.0029681	0.0030895
0.50	0.0087991	0.0070468	0.0055433	0.0045883	0.0039287	0.0034279	0.0031202	0.0029521	0.0029055	0.0029593	0.0030744
0.51	0.0088009	0.0070657	0.0055679	0.0046121	0.003952	0.0034496	0.0031338	0.002959	0.0029048	0.0029516	0.0030597
0.52	0.0088026	0.007084	0.0055922	0.0046355	0.0039749	0.003471	0.0031476	0.0029662	0.0029042	0.0029446	0.0030467
0.53	0.0088043	0.0071019	0.0056161	0.0046583	0.0039972	0.003492	0.0031611	0.0029742	0.0029051	0.002938	0.0030351
0.54	0.0088059	0.0071193	0.0056397	0.0046808	0.004019	0.0035127	0.0031743	0.0029826	0.0029065	0.0029315	0.0030243
0.55	0.0088075	0.0071362	0.0056629	0.004703	0.0040404	0.0035332	0.0031874	0.0029913	0.0029084	0.0029253	0.0030145
0.56	0.0088091	0.0071527	0.0056859	0.004725	0.0040613	0.0035531	0.0032007	0.0030007	0.0029109	0.0029202	0.0030048
0.57	0.0088106	0.0071688	0.0057083	0.0047464	0.004082	0.0035728	0.0032144	0.0030108	0.0029144	0.0029158	0.0029956
0.58	0.0088121	0.0071844	0.0057303	0.0047676	0.0041024	0.0035925	0.0032282	0.003021	0.0029191	0.0029128	0.0029868
0.59	0.0088135	0.0071997	0.005752	0.0047885	0.0041225	0.0036121	0.0032423	0.0030313	0.0029238	0.0029103	0.0029785
0.60	0.0088149	0.0072145	0.0057733	0.0048091	0.0041423	0.0036315	0.0032564	0.0030414	0.0029284	0.0029083	0.0029708
0.61	0.0088163	0.007229	0.0057942	0.0048293	0.0041617	0.0036506	0.0032711	0.0030515	0.0029329	0.0029071	0.0029634
0.62	0.0088176	0.0072432	0.0058147	0.0048493	0.0041808	0.0036692	0.0032857	0.0030616	0.0029373	0.0029063	0.0029565
0.63	0.0088189	0.007257	0.005835	0.0048691	0.0041997	0.0036876	0.0033003	0.0030716	0.0029418	0.0029056	0.0029505
0.64	0.0088202	0.0072704	0.0058549	0.0048885	0.0042182	0.0037057	0.003315	0.0030819	0.0029466	0.002905	0.0029449
0.65	0.0088214	0.0072836	0.0058743	0.0049075	0.0042364	0.0037237	0.0033299	0.0030921	0.0029517	0.0029045	0.0029396
0.66	0.0088226	0.0072964	0.0058935	0.0049262	0.0042543	0.0037415	0.0033449	0.0031024	0.002957	0.0029044	0.0029343
0.67	0.0088238	0.0073089	0.0059124	0.0049446	0.004272	0.0037592	0.0033597	0.0031127	0.0029624	0.0029052	0.002929
0.68	0.008825	0.0073212	0.005931	0.0049627	0.0042894	0.0037766	0.0033745	0.0031229	0.0029681	0.0029063	0.0029243
0.69	0.0088261	0.0073331	0.0059492	0.0049805	0.0043065	0.0037937	0.0033895	0.0031329	0.0029743	0.0029077	0.0029202
0.70	0.0088272	0.0073448	0.0059673	0.0049981	0.0043233	0.0038105	0.0034045	0.003143	0.0029807	0.0029095	0.0029166
0.71	0.0088283	0.0073563	0.0059851	0.0050155	0.0043399	0.0038274	0.0034198	0.0031531	0.0029873	0.0029117	0.0029138
0.72	0.0088293	0.0073675	0.0060027	0.0050328	0.0043561	0.0038439	0.0034351	0.003163	0.0029942	0.0029146	0.0029116
0.73	0.0088304	0.0073785	0.0060199	0.00505	0.0043722	0.0038603	0.0034502	0.0031728	0.0030016	0.0029183	0.0029097
0.74	0.0088314	0.0073892	0.0060369	0.005067	0.0043881	0.0038763	0.0034652	0.0031824	0.0030093	0.002922	0.0029082
0.75	0.0088324	0.0073997	0.0060537	0.0050838	0.0044038	0.0038921	0.00348	0.0031921	0.0030172	0.0029257	0.0029072
0.76	0.0088334	0.00741	0.0060701	0.0051003	0.0044195	0.0039077	0.0034946	0.0032021	0.0030251	0.0029293	0.0029065
0.77	0.0088343	0.0074202	0.0060864	0.0051166	0.0044351	0.0039231	0.0035091	0.0032122	0.0030329	0.0029329	0.0029059

0.78	0.0088353	0.0074301	0.0061023	0.0051326	0.0044505	0.0039383	0.0035234	0.0032223	0.0030407	0.0029364	0.0029054
0.79	0.0088362	0.0074399	0.006118	0.0051485	0.0044658	0.0039533	0.0035376	0.0032326	0.0030484	0.0029399	0.0029049
0.80	0.0088371	0.0074495	0.0061334	0.005164	0.0044809	0.0039682	0.0035516	0.003243	0.0030562	0.0029436	0.0029045
0.81	0.008838	0.0074589	0.0061487	0.0051794	0.0044957	0.0039827	0.0035653	0.0032534	0.003064	0.0029475	0.0029043
0.82	0.0088389	0.0074681	0.0061637	0.0051946	0.0045104	0.0039971	0.003579	0.0032642	0.0030716	0.0029515	0.0029048
0.83	0.0088397	0.0074772	0.0061785	0.0052095	0.004525	0.0040113	0.0035928	0.003275	0.0030796	0.0029558	0.0029056
0.84	0.0088406	0.0074861	0.0061931	0.0052242	0.0045395	0.0040252	0.0036065	0.0032857	0.0030874	0.0029599	0.0029065
0.85	0.0088414	0.0074948	0.0062074	0.0052388	0.0045538	0.004039	0.0036202	0.0032965	0.0030953	0.0029644	0.0029077
0.86	0.0088422	0.0075034	0.0062216	0.0052531	0.0045679	0.0040525	0.0036337	0.0033073	0.0031032	0.0029689	0.0029091
0.87	0.008843	0.0075118	0.0062355	0.0052673	0.0045819	0.004066	0.003647	0.0033183	0.0031111	0.0029739	0.0029108
0.88	0.0088438	0.0075202	0.0062492	0.0052813	0.0045957	0.0040794	0.0036602	0.0033293	0.003119	0.0029789	0.0029129
0.89	0.0088446	0.0075284	0.0062628	0.005295	0.0046093	0.0040927	0.0036731	0.0033404	0.0031267	0.0029841	0.0029154
0.90	0.0088454	0.0075364	0.0062761	0.0053087	0.0046227	0.0041057	0.0036859	0.0033513	0.0031344	0.0029894	0.0029185
0.91	0.0088461	0.0075444	0.0062894	0.0053222	0.004636	0.0041188	0.0036986	0.0033623	0.0031422	0.0029949	0.0029216
0.92	0.0088468	0.0075521	0.0063024	0.0053355	0.0046491	0.0041316	0.0037112	0.0033731	0.00315	0.0030008	0.0029246
0.93	0.0088476	0.0075598	0.0063153	0.0053487	0.0046621	0.0041444	0.0037238	0.0033842	0.0031577	0.0030068	0.0029275
0.94	0.0088483	0.0075673	0.006328	0.0053617	0.004675	0.0041569	0.0037363	0.0033953	0.0031652	0.0030131	0.0029304
0.95	0.008849	0.0075747	0.0063406	0.0053746	0.0046878	0.0041694	0.0037487	0.0034064	0.0031727	0.0030192	0.0029333
0.96	0.0088497	0.007582	0.006353	0.0053874	0.0047004	0.0041816	0.003761	0.0034177	0.0031801	0.0030255	0.0029361
0.97	0.0088503	0.0075891	0.0063652	0.0054002	0.004713	0.0041938	0.0037732	0.003429	0.0031875	0.0030318	0.002939
0.98	0.008851	0.0075962	0.0063773	0.0054128	0.0047255	0.0042059	0.0037852	0.0034402	0.003195	0.003038	0.002942
0.99	0.0088517	0.0076031	0.0063893	0.0054252	0.0047378	0.0042179	0.0037971	0.0034513	0.0032027	0.0030441	0.002945
1.00	0.0088523	0.0076099	0.0064012	0.0054375	0.00475	0.0042297	0.0038088	0.0034624	0.0032105	0.0030502	0.0029482
1.01	0.008853	0.0076166	0.006413	0.0054497	0.0047621	0.0042414	0.0038207	0.0034734	0.0032183	0.0030564	0.0029515

Appendix III

-

Accuracy experiment

No movement case data

Time (s)	Run 1	Run 2	Run 3	Run 4	Run 5
0.1	-7.22E-11	-5.13E-11	-1.6E-12	-5.13E-11	-3.44E-11
0.2	-8.57E-09	-8.57E-09	-2.68E-09	-6.15E-09	-4.14E-09
0.3	-5.44E-08	-4.65E-08	-2.9E-08	-4.65E-08	-3.44E-08
0.4	-2.63E-07	-2.43E-07	-2.1E-07	-2.43E-07	-2.46E-07
0.5	-4.74E-07	-4.58E-07	-6.13E-07	-4.58E-07	-5.26E-07
0.6	-1.56E-06	-1.51E-06	-1.11E-06	-1.58E-06	-1.5E-06
0.7	-1.59E-06	-1.53E-06	-1.4E-06	-1.53E-06	-1.61E-06
0.8	-3.16E-06	-2.48E-06	-1.29E-06	-2.48E-06	-1.95E-06
0.9	-8.44E-06	-8.32E-06	-7.22E-06	-8.32E-06	-8.16E-06
1	-1.28E-05	-1.25E-05	-1.42E-05	-1.25E-05	-1.32E-05
1.1	-1.99E-05	-1.99E-05	-1.6E-05	-1.97E-05	-2.06E-05
1.2	-3.02E-05	-3.02E-05	-2.63E-05	-3.02E-05	-2.84E-05
1.3	-5.11E-05	-5.11E-05	-4.51E-05	-5.08E-05	-4.87E-05
1.4	-6.22E-05	-6.38E-05	-6E-05	-6.38E-05	-6.05E-05
1.5	-7.02E-05	-7.02E-05	-7.19E-05	-7.02E-05	-7.13E-05
1.6	-7.49E-05	-7.43E-05	-7.86E-05	-7.43E-05	-7.81E-05
1.7	-8.7E-05	-8.7E-05	-7.45E-05	-8.83E-05	-8.57E-05
1.8	-9.34E-05	-8.93E-05	-9.14E-05	-9.34E-05	-8.63E-05
1.9	-9.46E-05	-8.86E-05	-8.53E-05	-8.86E-05	-8.39E-05
2	-6.47E-05	-6.25E-05	-6.89E-05	-6.47E-05	-5.34E-05
2.1	-9.65E-05	-9.86E-05	-8.87E-05	-9.86E-05	-9.34E-05
2.2	-7.32E-05	-6.72E-05	-7.71E-05	-6.72E-05	-6.72E-05
2.3	-5.09E-05	-5.61E-05	-7.38E-05	-5.61E-05	-6.02E-05
2.4	8.223E-05	8.468E-05	6.092E-05	8.468E-05	7.774E-05
2.5	0.0001195	0.0001195	0.0001152	0.0001216	0.0001358
2.6	0.000249	0.000249	0.0002065	0.000249	0.0002369
2.7	0.0004361	0.0004374	0.000365	0.0004361	0.0004173
2.8	0.000686	0.0006742	0.0005765	0.0006742	0.0006571
2.9	0.0009318	0.0009318	0.0008634	0.0009318	0.0009075
3	0.00121	0.0012118	0.0010405	0.0012118	0.0011305
3.1	0.0013857	0.0013857	0.0013444	0.0013825	0.0013883
3.2	0.0017702	0.0017584	0.001681	0.0017702	0.0017397
3.3	0.0024179	0.0023881	0.0022609	0.0024179	0.0023539
3.4	0.0030706	0.0030388	0.0028886	0.0030388	0.0030154
3.5	0.0038176	0.0038035	0.0036965	0.0038176	0.0038035
3.6	0.0045606	0.0045499	0.0043703	0.0045499	0.0044695
3.7	0.0056576	0.0056576	0.0053943	0.0056576	0.0056225
3.8	0.0068844	0.0068844	0.00664	0.0068291	0.0067837
3.9	0.008081	0.0080339	0.0077767	0.0080262	0.0079987
4	0.0090681	0.0090681	0.0088208	0.0090679	0.0089554
4.1	0.0100691	0.010029	0.0099398	0.010029	0.0100053
4.2	0.0113964	0.0113776	0.0110522	0.0113776	0.0112752
4.3	0.0127494	0.0126806	0.0124257	0.0126806	0.012596
4.4	0.014369	0.0143374	0.0139752	0.014282	0.0142099
4.5	0.0155833	0.0155429	0.0153082	0.0155833	0.0154857
4.6	0.0175679	0.0174575	0.0168157	0.0173972	0.0172973
4.7	0.0190339	0.0190795	0.0188449	0.0190219	0.0189821
4.8	0.0215039	0.0214176	0.0208846	0.0214176	0.021379
4.9	0.0231397	0.0230461	0.0225747	0.0230938	0.0227842
5	0.0255246	0.0255581	0.0247946	0.0255581	0.0253384
5.1	0.0277809	0.0277809	0.027301	0.0277809	0.0276248
5.2	0.0303476	0.0302854	0.0296944	0.0302854	0.030076
5.3	0.0335068	0.0333534	0.0325376	0.0334046	0.0331984
5.4	0.0365895	0.0365049	0.0359152	0.0366549	0.0363692
5.5	0.0390629	0.0388484	0.0382693	0.0388484	0.0385592
5.6	0.041782	0.0415826	0.0409656	0.0416691	0.041473
5.7	0.044039	0.0439304	0.0434035	0.0440096	0.0438157
5.8	0.0471943	0.047157	0.046639	0.0472463	0.0470037
5.9	0.0512171	0.0510918	0.0498626	0.0510918	0.0510476
6	0.0545727	0.0544907	0.0537102	0.0544907	0.054285
6.1	0.0587303	0.0586045	0.0578814	0.0584869	0.0583625
6.2	0.0629852	0.0628332	0.0621778	0.0627698	0.0625071

6.3	0.0678511	0.0678511	0.0667296	0.0678511	0.0674057
6.4	0.0734243	0.0734243	0.0722466	0.0733976	0.0731366
6.5	0.0782268	0.0781348	0.0770565	0.0781348	0.0776752
6.6	0.0825592	0.0824716	0.0814284	0.0826443	0.0821652
6.7	0.0874354	0.0872447	0.086556	0.0872447	0.0869745
6.8	0.0926516	0.0925001	0.0916061	0.0925001	0.092237
6.9	0.0967979	0.0967887	0.0960587	0.0968802	0.0963857
7	0.1008788	0.1004561	0.100041	0.1004561	0.1002179
7.1	0.1046247	0.104609	0.1043049	0.104609	0.1045078
7.2	0.111154	0.1109308	0.1095609	0.1109308	0.110291
7.3	0.1179791	0.117761	0.1162685	0.117761	0.1173775
7.4	0.1228621	0.1228621	0.1219028	0.1228692	0.1227469
7.5	0.12796	0.1277222	0.1261164	0.12796	0.127145
7.6	0.1318273	0.1318273	0.1311967	0.1318026	0.131552
7.7	0.1362478	0.1358789	0.134995	0.1358789	0.1357477
7.8	0.1422398	0.1420092	0.1408063	0.1420092	0.1421091
7.9	0.1481604	0.1478793	0.145914	0.1480222	0.1473376
8	0.1547989	0.1546044	0.1531442	0.1547989	0.1544226
8.1	0.1601435	0.1599698	0.1591223	0.1601435	0.1598794
8.2	0.1662457	0.16607	0.1652553	0.1662457	0.1656429
8.3	0.1734305	0.1735173	0.1717072	0.1734305	0.1729039
8.4	0.1815867	0.1813905	0.1797423	0.1813905	0.1811039
8.5	0.1873636	0.1869048	0.1855577	0.187158	0.1864582
8.6	0.1934568	0.1936522	0.1927041	0.1935044	0.1935474
8.7	0.2010255	0.2008457	0.1994661	0.2010255	0.2007032
8.8	0.208809	0.2088878	0.2071748	0.2087004	0.2078702
8.9	0.2153851	0.2153851	0.2144823	0.2150796	0.2145983
9	0.2211526	0.2213349	0.2194488	0.220922	0.2203963
9.1	0.2257594	0.2256949	0.2244555	0.2256949	0.2255987
9.2	0.2325847	0.2325847	0.2306681	0.2325847	0.2317227
9.3	0.2385551	0.2385546	0.2354153	0.2383234	0.2374768
9.4	0.2441003	0.2440826	0.2428019	0.2440826	0.243815
9.5	0.2518205	0.2517292	0.2502851	0.2520866	0.2512697
9.6	0.2575516	0.2575516	0.2553856	0.2571156	0.2568294
9.7	0.2642169	0.2642169	0.2629384	0.2642169	0.2643458
9.8	0.267698	0.267698	0.266397	0.2676729	0.2673235
9.9	0.2763624	0.2759247	0.274582	0.2759247	0.2757527
10	0.2829021	0.2824223	0.2811596	0.2824223	0.2825595
10.1	0.2876817	0.2875486	0.2860124	0.2881295	0.287412
10.2	0.2940937	0.2937523	0.2934621	0.2937523	0.2937998
10.3	0.3009659	0.3009659	0.2994783	0.3007369	0.3006155
10.4	0.3071632	0.3067226	0.3048455	0.3064746	0.3067052
10.5	0.3137795	0.3137698	0.3127017	0.3136332	0.314126
10.6	0.3195827	0.3194113	0.3173812	0.3189454	0.3186143
10.7	0.3249011	0.3249011	0.324527	0.3251538	0.3249011
10.8	0.3297444	0.329945	0.3293076	0.3296993	0.3295424
10.9	0.3330284	0.3332665	0.3325518	0.3332665	0.333061
11	0.3406692	0.3406035	0.3395962	0.3406035	0.3402466
11.1	0.3469801	0.3469801	0.3458096	0.3469801	0.3469801
11.2	0.3546762	0.3544181	0.3532525	0.3542827	0.3538666
11.3	0.3590763	0.3586822	0.3580356	0.3586822	0.3588029
11.4	0.3672252	0.3672252	0.3660275	0.3672252	0.366808
11.5	0.3752238	0.3746785	0.3731501	0.3749511	0.3745611
11.6	0.3818166	0.3818166	0.3805023	0.3816862	0.3813871
11.7	0.3880699	0.3878017	0.3860384	0.3877656	0.3872235
11.8	0.3955809	0.394885	0.391518	0.3952003	0.3946366
11.9	0.4017943	0.4017943	0.4011825	0.4017943	0.4017674
12	0.408592	0.4074255	0.407895	0.408276	0.4074255
12.1	0.4129766	0.4129766	0.4115371	0.4130963	0.4129618
12.2	0.4176433	0.4172511	0.4159481	0.4170253	0.4167405
12.3	0.4248069	0.4248069	0.4232846	0.424596	0.4240089
12.4	0.4307393	0.4302076	0.4297305	0.4302076	0.4301132
12.5	0.4366519	0.4363673	0.4340904	0.4366519	0.4357671

12.6	0.4437951	0.443685	0.4419755	0.443685	0.4433321
12.7	0.4499133	0.4494641	0.4477849	0.449202	0.4490946
12.8	0.4558357	0.4560342	0.4552743	0.4559702	0.4563674
12.9	0.4630113	0.4620425	0.4605809	0.4620425	0.4614905
13	0.4676176	0.4678148	0.467628	0.4678148	0.4678148
13.1	0.4723119	0.47181	0.4717905	0.4731542	0.4711629
13.2	0.4789255	0.4789204	0.4779251	0.4787288	0.4784063
13.3	0.4840697	0.484219	0.4827873	0.4840697	0.4841246
13.4	0.4895084	0.4897327	0.4882759	0.4897327	0.489469
13.5	0.495146	0.4940033	0.4918741	0.4940033	0.4931798
13.6	0.4985246	0.4985246	0.4958551	0.4985246	0.4975709
13.7	0.5064356	0.5061539	0.5039742	0.5061539	0.5054343
13.8	0.5104851	0.5106385	0.5090916	0.5104851	0.5106068
13.9	0.5158142	0.5158142	0.5153303	0.5158142	0.5150465
14	0.5201535	0.5199308	0.5194798	0.5199308	0.5198764
14.1	0.5246304	0.523868	0.5227751	0.5240803	0.5238171
14.2	0.5252243	0.5253391	0.5243679	0.525324	0.5252647
14.3	0.5324758	0.5329016	0.5308286	0.5324758	0.5327336
14.4	0.5389302	0.5387201	0.537784	0.5389302	0.5388333
14.5	0.5398393	0.5398393	0.5424486	0.5398393	0.5410265
14.6	0.5449608	0.5447851	0.543558	0.5447851	0.5449677
14.7	0.5518665	0.5519906	0.5504507	0.5519906	0.5514402
14.8	0.5517193	0.5527269	0.5510775	0.5519247	0.5533366
14.9	0.5595773	0.5586317	0.5569729	0.5586317	0.5583841
15	0.5629737	0.5629737	0.561853	0.5636129	0.5630651

Appendix IV

-

Accuracy experiment

One axis movement case data

Time (s)	Sensor R1 (m)	Rift R1 (m)	Error R1 (m)	Sensor R2 (m)	Rift R2 (m)	Error R2 (m)
0.1	0.003	-3.63432E-12	0.003	0.001	-2.25718E-12	0.001
0.2	0.004	-2.34769E-09	0.004000002	0.002	1.13213E-07	0.001999887
0.3	0.002	-2.89601E-08	0.002000029	0	1.56434E-06	1.56434E-06
0.4	0.003	-1.88215E-07	0.003000188	0.004	7.31242E-06	0.003992688
0.5	0.01	-6.21791E-07	0.010000622	0.004	2.21676E-05	0.003977832
0.6	0.008	-1.10601E-06	0.008001106	-0.002	5.41281E-05	0.002054128
0.7	0.007	-1.26539E-06	0.007001265	0.004	0.000119194	0.003880806
0.8	0.012	-1.26226E-06	0.012001262	0.008	0.000240902	0.007759098
0.9	0.013	-6.92577E-06	0.013006926	0.013	0.000462842	0.012537158
1	0.016	-1.4153E-05	0.016014153	0.016	0.000736635	0.015263365
1.1	0.015	-1.59701E-05	0.01501597	0.022	0.001011268	0.020988732
1.2	0.003	-2.60048E-05	0.003026005	0.026	0.001505092	0.024494908
1.3	-0.01	-4.43497E-05	0.00995565	0.027	0.001959303	0.025040697
1.4	-0.013	-5.99712E-05	0.012940029	0.031	0.002629005	0.028370995
1.5	-0.007	-7.21254E-05	0.006927875	0.04	0.003817674	0.036182326
1.6	0.041	-7.87617E-05	0.041078762	0.041	0.004591162	0.036408838
1.7	0.08	-7.59416E-05	0.080075942	0.039	0.005809226	0.033190774
1.8	0.157	-9.13793E-05	0.157091379	0.041	0.007062507	0.033937493
1.9	0.189	-8.53298E-05	0.18908533	0.043	0.008957743	0.034042257
2	0.235	-6.88755E-05	0.235068875	0.046	0.010374994	0.035625006
2.1	0.253	-9.47788E-05	0.253094779	0.053	0.01317251	0.03982749
2.2	0.286	-7.4275E-05	0.286074275	0.057	0.017501932	0.039498068
2.3	0.298	-7.38354E-05	0.298073835	0.065	0.020972932	0.044027068
2.4	0.314	5.17461E-05	0.313948254	0.07	0.025140759	0.044859241
2.5	0.316	0.000113462	0.315886538	0.072	0.027668759	0.044331241
2.6	0.327	0.000198912	0.326801088	0.075	0.033074506	0.041925494
2.7	0.337	0.000364625	0.336635375	0.084	0.036221746	0.047778254
2.8	0.346	0.000576473	0.345423527	0.083	0.041879758	0.041120242
2.9	0.355	0.000863416	0.354136584	0.09	0.046453007	0.043546993
3	0.365	0.001056745	0.363943255	0.093	0.050183401	0.042816599
3.1	0.366	0.001335259	0.364664741	0.1	0.058241941	0.041758059
3.2	0.377	0.001681869	0.375318131	0.102	0.066222996	0.035777004
3.3	0.375	0.002260911	0.372739089	0.108	0.070700288	0.037299712
3.4	0.389	0.002841369	0.386158631	0.116	0.082292132	0.033707868
3.5	0.387	0.003687183	0.383312817	0.126	0.093034998	0.032965002
3.6	0.394	0.004370288	0.389629712	0.132	0.105424136	0.026575864
3.7	0.41	0.005407766	0.404592234	0.142	0.11294093	0.02905907
3.8	0.417	0.006640016	0.410359984	0.14	0.118225023	0.021774977
3.9	0.43	0.007776686	0.422223314	0.144	0.124940366	0.019059634
4	0.447	0.008820846	0.438179154	0.15	0.146009058	0.003990942
4.1	0.454	0.009944255	0.444055745	0.152	0.154998451	0.002998451
4.2	0.463	0.011143758	0.451856242	0.155	0.166122362	0.011122362
4.3	0.464	0.012439137	0.451560863	0.164	0.189559922	0.025559922
4.4	0.474	0.013975221	0.460024779	0.168	0.20488508	0.03688508
4.5	0.479	0.015277625	0.463722375	0.18	0.218127042	0.038127042
4.6	0.498	0.016815737	0.481184263	0.183	0.236291349	0.053291349

Time (s)	Sensor R1 (m)	Rift R1 (m)	Error R1 (m)	Sensor R2 (m)	Rift R2 (m)	Error R2 (m)
4.7	0.506	0.018844945	0.487155055	0.192	0.262372285	0.070372285
4.8	0.519	0.020884598	0.498115402	0.2	0.277858496	0.077858496
4.9	0.524	0.022574734	0.501425266	0.203	0.290133625	0.087133625
5	0.536	0.02479456	0.51120544	0.21	0.316646755	0.106646755
5.1	0.534	0.027301002	0.506698998	0.22	0.338549852	0.118549852
5.2	0.55	0.029591262	0.520408738	0.227	0.361599803	0.134599803
5.3	0.556	0.032537557	0.523462443	0.234	0.396270484	0.162270484
5.4	0.568	0.035915226	0.532084774	0.238	0.42408216	0.18608216
5.5	0.573	0.038269341	0.534730659	0.244	0.438648582	0.194648582
5.6	0.587	0.040970627	0.546029373	0.248	0.476240873	0.228240873
5.7	0.598	0.043403547	0.554596453	0.26	0.511749983	0.251749983
5.8	0.608	0.046638966	0.561361034	0.259	0.535548985	0.276548985
5.9	0.619	0.050166022	0.568833978	0.272	0.574071229	0.302071229
6	0.63	0.053716015	0.576283985	0.271	0.610475183	0.339475183
6.1	0.635	0.057881437	0.577118563	0.284	0.632157803	0.348157803
6.2	0.649	0.062177766	0.586822234	0.287	0.657554269	0.370554269
6.3	0.658	0.066729635	0.591270365	0.296	0.70943296	0.41343296
6.4	0.676	0.072246552	0.603753448	0.301	0.757548749	0.456548749
6.5	0.677	0.077056527	0.599943473	0.307	0.780355692	0.473355692
6.6	0.68	0.081428371	0.598571629	0.31	0.833230913	0.523230913
6.7	0.693	0.086367458	0.606632542	0.319	0.875372529	0.556372529
6.8	0.698	0.09178564	0.60621436	0.325	0.900268674	0.575268674
6.9	0.706	0.096177489	0.609822511	0.331	0.962602973	0.631602973
7	0.719	0.100040987	0.618959013	0.337	1.027768016	0.690768016
7.1	0.721	0.104304895	0.616695105	0.348	1.072747707	0.724747707
7.2	0.738	0.109560922	0.628439078	0.354	1.115792513	0.761792513
7.3	0.746	0.11626853	0.62973147	0.36	1.146374583	0.786374583
7.4	0.752	0.122009493	0.629990507	0.368	1.181949258	0.813949258
7.5	0.764	0.12611635	0.63788365	0.375	1.268941879	0.893941879
7.6	0.77	0.131196663	0.638803337	0.381	1.284325719	0.903325719
7.7	0.771	0.135063499	0.635936501	0.385	1.33515799	0.95015799
7.8	0.78	0.140806332	0.639193668	0.394	1.408890128	1.014890128
7.9	0.785	0.146249592	0.638750408	0.407	1.440142155	1.033142155
8	0.775	0.153144225	0.621855775	0.406	1.50142324	1.09542324
8.1	0.772	0.158798188	0.613201812	0.424	1.579709768	1.155709768
8.2	0.773	0.165079936	0.607920064	0.426	1.618688345	1.192688345
8.3	0.775	0.171978623	0.603021377	0.44	1.680971265	1.240971265
8.4	0.78	0.179470152	0.600529848	0.446	1.73400712	1.28800712
8.5	0.778	0.185427174	0.592572826	0.457	1.800658107	1.343658107
8.6	0.783	0.192704067	0.590295933	0.455	1.849512219	1.394512219
8.7	0.784	0.19958654	0.58441346	0.466	1.911289692	1.445289692
8.8	0.78	0.207174793	0.572825207	0.474	1.972068667	1.498068667
8.9	0.778	0.214272052	0.563727948	0.478	2.051461935	1.573461935
9	0.772	0.21944882	0.55255118	0.482	2.116867065	1.634867065
9.1	0.769	0.224359781	0.544640219	0.497	2.166821241	1.669821241
9.2	0.747	0.230674371	0.516325629	0.499	2.233855009	1.734855009

Time (s)	Sensor R1 (m)	Rift R1 (m)	Error R1 (m)	Sensor R2 (m)	Rift R2 (m)	Error R2 (m)
9.3	0.732	0.23541525	0.49658475	0.511	2.34944129	1.83844129
9.4	0.729	0.242801905	0.486198095	0.518	2.364860296	1.846860296
9.5	0.738	0.249998614	0.488001386	0.529	2.471960545	1.942960545
9.6	0.701	0.255261213	0.445738787	0.534	2.539828539	2.005828539
9.7	0.716	0.26293835	0.45306165	0.538	2.565778732	2.027778732
9.8	0.723	0.266194463	0.456805537	0.547	2.666137934	2.119137934
9.9	0.718	0.274581969	0.443418031	0.559	2.804042816	2.245042816
10	0.735	0.281159639	0.453840361	0.566	2.832215548	2.266215548
10.1	0.744	0.285891026	0.458108974	0.578	2.901602983	2.323602983
10.2	0.755	0.293462127	0.461537873	0.583	2.96807909	2.38507909
10.3	0.771	0.299478292	0.471521708	0.588	3.040693998	2.452693998
10.4	0.784	0.304845482	0.479154518	0.596	3.117794275	2.521794275
10.5	0.809	0.31273827	0.49626173	0.602	3.204020739	2.602020739
10.6	0.811	0.317198128	0.493801872	0.603	3.24040699	2.63740699
10.7	0.802	0.324823886	0.477176114	0.614	3.302151918	2.688151918
10.8	0.812	0.329307586	0.482692414	0.62	3.409439325	2.789439325
10.9	0.808	0.33234939	0.47565061	0.629	3.512171984	2.883171984
11	0.816	0.33946991	0.47653009	0.636	3.587306976	2.951306976
11.1	0.826	0.346274704	0.479725296	0.64	3.615535498	2.975535498
11.2	0.819	0.35336563	0.46563437	0.641	3.708727837	3.067727837
11.3	0.831	0.358035594	0.472964406	0.653	3.811717272	3.158717272
11.4	0.827	0.366027474	0.460972526	0.657	3.934234858	3.277234858
11.5	0.835	0.373073369	0.461926631	0.665	4.021919727	3.356919727
11.6	0.857	0.380351365	0.476648635	0.673	4.13453722	3.46153722
11.7	0.842	0.386038423	0.455961577	0.68	4.182407856	3.502407856
11.8	0.832	0.391517997	0.440482003	0.687	4.249371529	3.562371529
11.9	0.827	0.40137586	0.42562414	0.693	4.409321308	3.716321308
12	0.831	0.407894969	0.423105031	0.696	4.462875843	3.766875843
12.1	0.819	0.411681235	0.407318765	0.712	4.571305275	3.859305275
12.2	0.83	0.41580376	0.41419624	0.715	4.675339222	3.960339222
12.3	0.831	0.423384219	0.407615781	0.724	4.717147827	3.993147827
12.4	0.842	0.429760575	0.412239425	0.728	4.854754448	4.126754448
12.5	0.844	0.434027046	0.409972954	0.737	4.935510159	4.198510159
12.6	0.849	0.442081571	0.406918429	0.742	5.007411957	4.265411957
12.7	0.862	0.44778493	0.41421507	0.755	5.132417679	4.377417679
12.8	0.883	0.455274284	0.427725716	0.757	5.217625141	4.460625141
12.9	0.902	0.460489511	0.441510489	0.77	5.302294254	4.532294254
13	0.937	0.467628032	0.469371968	0.774	5.398850918	4.624850918
13.1	0.95	0.471790493	0.478209507	0.786	5.465549946	4.679549946
13.2	0.97	0.478000164	0.491999836	0.79	5.604640961	4.814640961
13.3	0.982	0.483034223	0.498965777	0.789	5.667058468	4.878058468
13.4	1.006	0.488275886	0.517724114	0.798	5.755474567	4.957474567
13.5	1.013	0.491874129	0.521125871	0.804	5.914860725	5.110860725
13.6	1.049	0.495720565	0.553279435	0.81	5.936445713	5.126445713
13.7	1.069	0.504567504	0.564432496	0.818	6.009094715	5.191094715
13.8	1.114	0.509091556	0.604908444	0.826	6.116493225	5.290493225

Time (s)	Sensor R1 (m)	Rift R1 (m)	Error R1 (m)	Sensor R2 (m)	Rift R2 (m)	Error R2 (m)
13.9	1.122	0.515330315	0.606669685	0.827	6.28082037	5.45382037
14	1.137	0.519479811	0.617520189	0.836	6.387803555	5.551803555
14.1	1.144	0.522996187	0.621003813	0.838	6.458694458	5.620694458
14.2	1.165	0.524594784	0.640405216	0.841	6.635846138	5.794846138
14.3	1.161	0.530828595	0.630171405	0.845	6.677906513	5.832906513
14.4	1.172	0.537869871	0.634130129	0.851	6.837869167	5.986869167
14.5	1.176	0.542012691	0.633987309	0.853	6.878777027	6.025777027
14.6	1.195	0.543558002	0.651441998	0.856	6.972804546	6.116804546
14.7	1.207	0.550795317	0.656204683	0.86	7.048436642	6.188436642
14.8	1.221	0.551077485	0.669922515	0.869	7.150793552	6.281793552
14.9	1.227	0.556958616	0.670041384	0.871	7.395423412	6.524423412
15	1.243	0.562601805	0.680398195	0.869	7.379557133	6.510557133

Time (s)	Sensor R3 (m)	Rift R3 (m)	Error R3 (m)	Sensor R4 (m)	Rift R4 (m)	Error R4 (m)
0.1	0.001	2.69294E-14	0.001	0.003	2.69294E-14	0.003
0.2	-0.001	2.60307E-09	0.001000003	-0.003	2.27458E-09	0.003000002
0.3	-0.002	1.28136E-07	0.002000128	0.002	1.28136E-07	0.001999872
0.4	0.002	7.83813E-07	0.001999216	-0.002	7.83813E-07	0.002000784
0.5	0.003	3.04335E-06	0.002996957	-0.003	3.00609E-06	0.003003006
0.6	-0.002	8.55283E-06	0.002008553	0.002	8.26853E-06	0.001991731
0.7	0	1.69311E-05	1.69311E-05	0.001	1.67972E-05	0.000983203
0.8	0.002	2.98944E-05	0.001970106	0.003	2.92202E-05	0.00297078
0.9	0	4.74992E-05	4.74992E-05	-0.001	4.77253E-05	0.001047725
1	-0.002	7.72426E-05	0.002077243	0.002	7.72426E-05	0.001922757
1.1	0.002	0.00012087	0.00187913	-0.002	0.000121961	0.002121961
1.2	0.001	0.000171556	0.000828444	0	0.00017326	0.00017326
1.3	-0.002	0.000245051	0.002245051	-0.001	0.000245051	0.001245051
1.4	0	0.000329963	0.000329963	0.004	0.000329963	0.003670037
1.5	0.003	0.000433074	0.002566926	0.005	0.000433074	0.004566926
1.6	0.001	0.000708431	0.000291569	-0.001	0.000708431	0.001708431
1.7	0.003	0.000906808	0.002093192	0	0.000906808	0.000906808
1.8	0.005	0.001185336	0.003814664	-0.002	0.001185336	0.003185336
1.9	0.002	0.001311942	0.000688058	-0.001	0.001317477	0.002317477
2	0.005	0.00164693	0.00335307	0	0.001623769	0.001623769
2.1	0.009	0.002515478	0.006484522	0.001	0.002461483	0.001461483
2.2	0.014	0.003745822	0.010254178	0.003	0.003731997	0.000731997
2.3	0.015	0.004686535	0.010313465	0.005	0.004722197	0.000277803
2.4	0.017	0.005749661	0.011250339	0.009	0.005623681	0.003376319
2.5	0.026	0.007350903	0.018649097	0.013	0.007299152	0.005700848
2.6	0.029	0.009856986	0.019143014	0.021	0.009842899	0.011157101
2.7	0.04	0.011423772	0.028576228	0.024	0.011423772	0.012576228
2.8	0.048	0.01437682	0.03362318	0.035	0.01437682	0.02062318
2.9	0.055	0.014717157	0.040282843	0.036	0.01477992	0.02122008
3	0.059	0.016156713	0.042843287	0.047	0.016129954	0.030870046
3.1	0.06	0.015471027	0.044528973	0.054	0.015353139	0.038646861
3.2	0.067	0.017743636	0.049256364	0.062	0.017743636	0.044256364
3.3	0.075	0.020157723	0.054842277	0.068	0.020157723	0.047842277
3.4	0.081	0.023986375	0.057013625	0.075	0.023986375	0.051013625
3.5	0.092	0.024737414	0.067262586	0.081	0.024737414	0.056262586
3.6	0.098	0.028120516	0.069879484	0.086	0.028246032	0.057753968
3.7	0.105	0.028490229	0.076509771	0.095	0.028365156	0.066634844
3.8	0.107	0.033866048	0.073133952	0.102	0.033900633	0.068099367
3.9	0.117	0.037923831	0.079076169	0.105	0.037760884	0.067239116
4	0.123	0.03749118	0.08550882	0.117	0.037808027	0.079191973
4.1	0.133	0.038994152	0.094005848	0.125	0.038881749	0.086118251
4.2	0.137	0.043070149	0.093929851	0.131	0.043114673	0.087885327
4.3	0.144	0.047547948	0.096452052	0.139	0.047547948	0.091452052
4.4	0.148	0.057293799	0.090706201	0.156	0.057293799	0.098706201
4.5	0.161	0.051291101	0.109708899	0.16	0.051208537	0.108791463
4.6	0.16	0.05273594	0.10726406	0.172	0.052631926	0.119368074

Time (s)	Sensor R3 (m)	Rift R3 (m)	Error R3 (m)	Sensor R4 (m)	Rift R4 (m)	Error R4 (m)
4.7	0.17	0.056473412	0.113526588	0.176	0.056473412	0.119526588
4.8	0.18	0.07781215	0.10218785	0.188	0.07781215	0.11018785
4.9	0.188	0.071126267	0.116873733	0.192	0.071126267	0.120873733
5	0.194	0.075565763	0.118434237	0.2	0.07630612	0.12369388
5.1	0.206	0.081949979	0.124050021	0.209	0.081891276	0.127108724
5.2	0.208	0.094433472	0.113566528	0.218	0.094348602	0.123651398
5.3	0.219	0.092980511	0.126019489	0.222	0.092980511	0.129019489
5.4	0.224	0.10618411	0.11781589	0.237	0.105958745	0.131041255
5.5	0.233	0.099201843	0.133798157	0.244	0.099201843	0.144798157
5.6	0.235	0.108360544	0.126639456	0.258	0.108360544	0.149639456
5.7	0.245	0.10633669	0.13866331	0.261	0.10633669	0.15466331
5.8	0.252	0.123624712	0.128375288	0.272	0.123737939	0.148262061
5.9	0.264	0.152362004	0.111637996	0.271	0.151976839	0.119023161
6	0.271	0.142596707	0.128403293	0.288	0.142596707	0.145403293
6.1	0.282	0.14130345	0.14069655	0.291	0.140953556	0.150046444
6.2	0.286	0.15335229	0.13264771	0.302	0.152232811	0.149767189
6.3	0.294	0.170518428	0.123481572	0.308	0.170373037	0.137626963
6.4	0.3	0.166396067	0.133603933	0.324	0.166078895	0.157921105
6.5	0.308	0.167219907	0.140780093	0.332	0.167219907	0.164780093
6.6	0.31	0.179664299	0.130335701	0.339	0.179664299	0.159335701
6.7	0.321	0.163284436	0.157715564	0.342	0.16249913	0.17950087
6.8	0.324	0.196047261	0.127952739	0.358	0.196047261	0.161952739
6.9	0.334	0.213942528	0.120057472	0.366	0.213942528	0.152057472
7	0.34	0.244466782	0.095533218	0.375	0.244466782	0.130533218
7.1	0.36	0.222955674	0.137044326	0.383	0.222955674	0.160044326
7.2	0.366	0.236444086	0.129555914	0.399	0.236444086	0.162555914
7.3	0.372	0.268144518	0.103855482	0.406	0.268144518	0.137855482
7.4	0.379	0.259688914	0.119311086	0.415	0.259519368	0.155480632
7.5	0.385	0.250896007	0.134103993	0.423	0.250896007	0.172103993
7.6	0.392	0.286484271	0.105515729	0.434	0.285604715	0.148395285
7.7	0.398	0.309334308	0.088665692	0.443	0.309347123	0.133652877
7.8	0.402	0.301529855	0.100470145	0.457	0.299818248	0.157181752
7.9	0.407	0.319310844	0.087689156	0.459	0.318591684	0.140408316
8	0.413	0.337655276	0.075344724	0.469	0.338063776	0.130936224
8.1	0.425	0.355933011	0.069066989	0.476	0.355933011	0.120066989
8.2	0.431	0.331204265	0.099795735	0.486	0.331204265	0.154795735
8.3	0.442	0.390837967	0.051162033	0.49	0.391464412	0.098535588
8.4	0.445	0.391597629	0.053402371	0.501	0.391597629	0.109402371
8.5	0.458	0.399995208	0.058004792	0.503	0.399995208	0.103004792
8.6	0.463	0.380287796	0.082712204	0.514	0.379911929	0.134088071
8.7	0.472	0.432399839	0.039600161	0.523	0.431874484	0.091125516
8.8	0.477	0.449479967	0.027520033	0.529	0.449171722	0.079828278
8.9	0.489	0.498548627	0.009548627	0.537	0.495551795	0.041448205
9	0.492	0.462382972	0.029617028	0.55	0.462439358	0.087560642
9.1	0.502	0.52622515	0.02422515	0.559	0.52622515	0.03277485
9.2	0.509	0.502778649	0.006221351	0.573	0.499673992	0.073326008

Time (s)	Sensor R3 (m)	Rift R3 (m)	Error R3 (m)	Sensor R4 (m)	Rift R4 (m)	Error R4 (m)
9.3	0.52	0.507215381	0.012784619	0.578	0.508979499	0.069020501
9.4	0.521	0.567647517	0.046647517	0.587	0.569691181	0.017308819
9.5	0.527	0.513274074	0.013725926	0.595	0.513274074	0.081725926
9.6	0.533	0.561181724	0.028181724	0.607	0.562900186	0.044099814
9.7	0.543	0.59044677	0.04744677	0.614	0.59044677	0.02355323
9.8	0.55	0.576471329	0.026471329	0.623	0.575807154	0.047192846
9.9	0.559	0.64962548	0.09062548	0.632	0.64962548	0.01762548
10	0.567	0.650355279	0.083355279	0.644	0.6496889	0.0056889
10.1	0.574	0.681285739	0.107285739	0.646	0.681285739	0.035285739
10.2	0.579	0.717453182	0.138453182	0.659	0.717453182	0.058453182
10.3	0.586	0.707104504	0.121104504	0.665	0.707104504	0.042104504
10.4	0.592	0.753827035	0.161827035	0.674	0.755728126	0.081728126
10.5	0.601	0.791186929	0.190186929	0.677	0.791186929	0.114186929
10.6	0.607	0.795301497	0.188301497	0.691	0.797718704	0.106718704
10.7	0.61	0.836391628	0.226391628	0.699	0.837811589	0.138811589
10.8	0.617	0.835143328	0.218143328	0.709	0.835143328	0.126143328
10.9	0.623	0.842499733	0.219499733	0.714	0.842499733	0.128499733
11	0.632	0.879017293	0.247017293	0.726	0.874982119	0.148982119
11.1	0.646	0.959875107	0.313875107	0.734	0.959273636	0.225273636
11.2	0.651	0.940411329	0.289411329	0.742	0.946612358	0.204612358
11.3	0.657	1.024030805	0.367030805	0.753	1.021188378	0.268188378
11.4	0.663	1.02402997	0.36102997	0.758	1.016747952	0.258747952
11.5	0.671	1.065438032	0.394438032	0.76	1.066975713	0.306975713
11.6	0.674	1.114583135	0.440583135	0.774	1.110673904	0.336673904
11.7	0.682	1.020620584	0.338620584	0.772	1.020620584	0.248620584
11.8	0.689	1.036930799	0.347930799	0.787	1.033993244	0.246993244
11.9	0.691	1.167147875	0.476147875	0.794	1.167147875	0.373147875
12	0.697	1.212331533	0.515331533	0.806	1.212331533	0.406331533
12.1	0.707	1.207096577	0.500096577	0.804	1.207096577	0.403096577
12.2	0.711	1.307444334	0.596444334	0.811	1.307444334	0.496444334
12.3	0.719	1.334426165	0.615426165	0.818	1.334426165	0.516426165
12.4	0.725	1.341439605	0.616439605	0.821	1.334995508	0.513995508
12.5	0.731	1.368707538	0.637707538	0.828	1.367168427	0.539168427
12.6	0.735	1.399018884	0.664018884	0.833	1.398171902	0.565171902
12.7	0.748	1.458247662	0.710247662	0.838	1.458137631	0.620137631
12.8	0.749	1.471012831	0.722012831	0.849	1.469772339	0.620772339
12.9	0.762	1.530665994	0.768665994	0.85	1.530665994	0.680665994
13	0.764	1.562230349	0.798230349	0.852	1.561421156	0.709421156
13.1	0.773	1.603390336	0.830390336	0.859	1.601388335	0.742388335
13.2	0.781	1.656029463	0.875029463	0.868	1.650403857	0.782403857
13.3	0.784	1.67866528	0.89466528	0.871	1.67866528	0.80766528
13.4	0.79	1.694577456	0.904577456	0.878	1.692903996	0.814903996
13.5	0.804	1.805575252	1.001575252	0.887	1.805203319	0.918203319
13.6	0.806	1.842264414	1.036264414	0.902	1.849651575	0.947651575
13.7	0.815	1.819151878	1.004151878	0.909	1.819151878	0.910151878
13.8	0.821	1.911301732	1.090301732	0.926	1.912475824	0.986475824

Time (s)	Sensor R3 (m)	Rift R3 (m)	Error R3 (m)	Sensor R4 (m)	Rift R4 (m)	Error R4 (m)
13.9	0.825	1.930167437	1.105167437	0.928	1.931107163	1.003107163
14	0.829	1.926888943	1.097888943	0.948	1.926888943	0.978888943
14.1	0.84	1.99026978	1.15026978	0.95	1.99026978	1.04026978
14.2	0.841	2.061908007	1.220908007	0.964	2.061908007	1.097908007
14.3	0.84	2.121297836	1.281297836	0.97	2.124519825	1.154519825
14.4	0.854	2.187485933	1.333485933	0.979	2.188254356	1.209254356
14.5	0.859	2.188296795	1.329296795	0.988	2.19075799	1.20275799
14.6	0.863	2.269643545	1.406643545	1.001	2.269643545	1.268643545
14.7	0.872	2.299753904	1.427753904	1.004	2.304967403	1.300967403
14.8	0.878	2.328968525	1.450968525	1.016	2.331830502	1.315830502
14.9	0.875	2.495809317	1.620809317	1.018	2.495809317	1.477809317
15	0.877	2.504092693	1.627092693	1.033	2.504092693	1.471092693

Time (s)	Sensor R5 (m)	Rift R5 (m)	Error R5 (m)
0.1	0.002	2.69294E-14	0.002
0.2	-0.002	2.27458E-09	0.002000002
0.3	-0.001	1.28136E-07	0.001000128
0.4	-0.002	7.49313E-07	0.002000749
0.5	0	3.00609E-06	3.00609E-06
0.6	-0.004	8.46553E-06	0.004008466
0.7	0.003	1.69311E-05	0.002983069
0.8	0	2.92202E-05	2.92202E-05
0.9	-0.001	4.73064E-05	0.001047306
1	0	7.72426E-05	7.72426E-05
1.1	0.003	0.000121961	0.002878039
1.2	-0.001	0.000174998	0.001174998
1.3	0.001	0.000245051	0.000754949
1.4	-0.007	0.000329963	0.007329963
1.5	-0.002	0.000433074	0.002433074
1.6	-0.001	0.000731988	0.001731988
1.7	0.001	0.000917833	8.21672E-05
1.8	-0.002	0.001197418	0.003197418
1.9	-0.001	0.001317477	0.002317477
2	-0.002	0.001607784	0.003607784
2.1	0.001	0.002515478	0.001515478
2.2	0	0.003745822	0.003745822
2.3	0.001	0.004722197	0.003722197
2.4	0.009	0.005623681	0.003376319
2.5	0.011	0.007350903	0.003649097
2.6	0.015	0.009826108	0.005173892
2.7	0.017	0.011423772	0.005576228
2.8	0.021	0.01437682	0.00662318
2.9	0.027	0.01477992	0.01222008
3	0.034	0.016156713	0.017843287
3.1	0.041	0.015353139	0.025646861
3.2	0.052	0.017748833	0.034251167
3.3	0.056	0.020157723	0.035842277
3.4	0.068	0.024001509	0.043998491
3.5	0.07	0.024863029	0.045136971
3.6	0.075	0.028246032	0.046753968
3.7	0.081	0.028365156	0.052634844
3.8	0.093	0.033866048	0.059133952
3.9	0.098	0.037760884	0.060239116
4	0.11	0.037662253	0.072337747
4.1	0.116	0.039174095	0.076825905
4.2	0.127	0.043114673	0.083885327
4.3	0.133	0.047986865	0.085013135
4.4	0.148	0.057284497	0.090715503
4.5	0.151	0.051291101	0.099708899
4.6	0.16	0.05273594	0.10726406

Time (s)	Sensor R5 (m)	Rift R5 (m)	Error R5 (m)
4.7	0.164	0.056473412	0.107526588
4.8	0.173	0.077811092	0.095188908
4.9	0.179	0.071172871	0.107827129
5	0.187	0.075979322	0.111020678
5.1	0.195	0.081949979	0.113050021
5.2	0.201	0.094348602	0.106651398
5.3	0.208	0.092781641	0.115218359
5.4	0.221	0.105958745	0.115041255
5.5	0.228	0.099201843	0.128798157
5.6	0.235	0.108360544	0.126639456
5.7	0.239	0.10633669	0.13266331
5.8	0.253	0.123737939	0.129262061
5.9	0.258	0.151541054	0.106458946
6	0.267	0.142897144	0.124102856
6.1	0.272	0.140953556	0.131046444
6.2	0.287	0.15335229	0.13364771
6.3	0.294	0.17014505	0.12385495
6.4	0.304	0.166396067	0.137603933
6.5	0.308	0.167180061	0.140819939
6.6	0.318	0.17890431	0.13909569
6.7	0.325	0.16249913	0.16250087
6.8	0.334	0.195840523	0.138159477
6.9	0.345	0.213942528	0.131057472
7	0.355	0.244025201	0.110974799
7.1	0.361	0.222955674	0.138044326
7.2	0.377	0.236444086	0.140555914
7.3	0.381	0.267350912	0.113649088
7.4	0.389	0.259519368	0.129480632
7.5	0.398	0.250896007	0.147103993
7.6	0.406	0.286484271	0.119515729
7.7	0.41	0.309334308	0.100665692
7.8	0.43	0.301529855	0.128470145
7.9	0.433	0.318591684	0.114408316
8	0.445	0.338063776	0.106936224
8.1	0.446	0.3547948	0.0912052
8.2	0.461	0.330279946	0.130720054
8.3	0.467	0.391464412	0.075535588
8.4	0.475	0.39327088	0.08172912
8.5	0.483	0.399880111	0.083119889
8.6	0.496	0.379911929	0.116088071
8.7	0.501	0.432399839	0.068600161
8.8	0.513	0.449859232	0.063140768
8.9	0.519	0.495551795	0.023448205
9	0.532	0.462382972	0.069617028
9.1	0.538	0.523255348	0.014744652
9.2	0.548	0.499673992	0.048326008

Time (s)	Sensor R5 (m)	Rift R5 (m)	Error R5 (m)
9.3	0.553	0.507215381	0.045784619
9.4	0.566	0.569691181	0.003691181
9.5	0.57	0.513274074	0.056725926
9.6	0.583	0.562900186	0.020099814
9.7	0.59	0.588206172	0.001793828
9.8	0.604	0.576471329	0.027528671
9.9	0.608	0.649348021	0.041348021
10	0.619	0.650355279	0.031355279
10.1	0.62	0.681285739	0.061285739
10.2	0.64	0.718677521	0.078677521
10.3	0.651	0.705750585	0.054750585
10.4	0.653	0.755728126	0.102728126
10.5	0.659	0.791186929	0.132186929
10.6	0.664	0.795301497	0.131301497
10.7	0.668	0.837811589	0.169811589
10.8	0.685	0.835143328	0.150143328
10.9	0.69	0.842499733	0.152499733
11	0.7	0.874982119	0.174982119
11.1	0.699	0.959875107	0.260875107
11.2	0.715	0.940411329	0.225411329
11.3	0.722	1.021188378	0.299188378
11.4	0.736	1.020541191	0.284541191
11.5	0.74	1.065438032	0.325438032
11.6	0.743	1.114583135	0.371583135
11.7	0.749	1.019800782	0.270800782
11.8	0.762	1.033993244	0.271993244
11.9	0.769	1.167147875	0.398147875
12	0.773	1.212712646	0.439712646
12.1	0.783	1.201785803	0.418785803
12.2	0.788	1.307444334	0.519444334
12.3	0.793	1.334426165	0.541426165
12.4	0.799	1.341439605	0.542439605
12.5	0.804	1.367849827	0.563849827
12.6	0.813	1.398171902	0.585171902
12.7	0.814	1.458247662	0.644247662
12.8	0.824	1.471721053	0.647721053
12.9	0.823	1.528483152	0.705483152
13	0.834	1.562230349	0.728230349
13.1	0.835	1.599679828	0.764679828
13.2	0.838	1.650403857	0.812403857
13.3	0.844	1.67866528	0.83466528
13.4	0.85	1.691207528	0.841207528
13.5	0.856	1.805575252	0.949575252
13.6	0.868	1.846087694	0.978087694
13.7	0.87	1.819066882	0.949066882
13.8	0.877	1.911301732	1.034301732

Time (s)	Sensor R5 (m)	Rift R5 (m)	Error R5 (m)
13.9	0.885	1.93230927	1.04730927
14	0.897	1.925240278	1.028240278
14.1	0.904	1.99026978	1.08626978
14.2	0.907	2.061908007	1.154908007
14.3	0.918	2.121297836	1.203297836
14.4	0.932	2.188254356	1.256254356
14.5	0.935	2.188296795	1.253296795
14.6	0.948	2.269585848	1.321585848
14.7	0.953	2.299753904	1.346753904
14.8	0.956	2.331830502	1.375830502
14.9	0.965	2.495809317	1.530809317
15	0.973	2.503801107	1.530801107

Appendix V

-

Accuracy experiment

Two axis movement case data

Time (s)	Sensor R1 (m)	Rift R1 (m)	Error R1 (m)	Sensor R2 (m)	Rift R2 (m)	Error R2 (m)
0.1	-0.003	-2.53513E-16	0.003	0.001	-2.53513E-16	0.001
0.2	0.001	-8.80074E-10	0.001000001	0	-6.9215E-10	6.9215E-10
0.3	-0.001	1.98988E-08	0.00100002	-0.002	2.60263E-08	0.002000026
0.4	0.001	1.87691E-07	0.000999812	-0.004	1.87691E-07	0.004000188
0.5	0	1.03224E-06	1.03224E-06	-0.001	1.03224E-06	0.001001032
0.6	-0.001	3.02066E-06	0.001003021	0	2.89244E-06	2.89244E-06
0.7	0.001	5.77112E-06	0.000994229	0.001	5.77934E-06	0.000994221
0.8	0.004	7.80442E-06	0.003992196	0.002	7.97797E-06	0.001992022
0.9	0.001	1.31414E-05	0.000986859	0.008	1.31536E-05	0.007986846
1	0.002	2.13951E-05	0.001978605	0	2.07439E-05	2.07439E-05
1.1	-0.003	3.10729E-05	0.003031073	-0.004	3.10729E-05	0.004031073
1.2	0	4.72519E-05	4.72519E-05	0.003	4.70259E-05	0.002952974
1.3	-0.005	7.00761E-05	0.005070076	0.005	7.00761E-05	0.004929924
1.4	0.003	8.33436E-05	0.002916656	-0.001	8.23035E-05	0.001082304
1.5	-0.001	0.00010703	0.00110703	0.003	0.00010703	0.00289297
1.6	-0.001	0.000205757	0.001205757	0.008	0.000193021	0.007806979
1.7	0	0.000280807	0.000280807	0.007	0.000280807	0.006719193
1.8	0	0.000371804	0.000371804	0.011	0.000381223	0.010618777
1.9	-0.004	0.000350476	0.004350476	0.013	0.000358401	0.012641599
2	0.002	0.000420405	0.001579595	0.012	0.000401284	0.011598716
2.1	0.001	0.000801473	0.000198527	0.014	0.000801473	0.013198527
2.2	0.004	0.00147301	0.00252699	0.022	0.00147301	0.02052699
2.3	0.006	0.001918987	0.004081013	0.023	0.001895238	0.021104762
2.4	0.009	0.002318957	0.006681043	0.033	0.002238043	0.030761957
2.5	0.013	0.00318776	0.00981224	0.035	0.003159697	0.031840303
2.6	0.01	0.004552359	0.005447641	0.044	0.004544322	0.039455678
2.7	0.012	0.005387773	0.006612227	0.047	0.005385292	0.041614708
2.8	0.013	0.007138394	0.005861606	0.057	0.007137223	0.049862777
2.9	0.014	0.006980424	0.007019576	0.056	0.006950284	0.049049716
3	0.025	0.007626242	0.017373758	0.069	0.007599591	0.061400409
3.1	0.027	0.006705619	0.020294381	0.071	0.006658866	0.064341134
3.2	0.034	0.007934316	0.026065684	0.075	0.00794117	0.06705883
3.3	0.036	0.009108552	0.026891448	0.078	0.009136165	0.068863835
3.4	0.043	0.011277142	0.031722858	0.083	0.011304522	0.071695478
3.5	0.045	0.01146139	0.03353861	0.091	0.01146139	0.07953861
3.6	0.056	0.013417472	0.042582528	0.097	0.013435389	0.083564611
3.7	0.061	0.01299101	0.04800899	0.1	0.012985778	0.087014222
3.8	0.067	0.016180485	0.050819515	0.109	0.016187493	0.092812507
3.9	0.072	0.018344261	0.053655739	0.119	0.018213755	0.100786245
4	0.084	0.018011697	0.065988303	0.118	0.018011697	0.099988303
4.1	0.088	0.017971022	0.070028978	0.127	0.017971022	0.109028978
4.2	0.091	0.020215662	0.070784338	0.141	0.020298846	0.120701154
4.3	0.09	0.022711828	0.067288172	0.142	0.022414526	0.119585474
4.4	0.106	0.02887808	0.07712192	0.15	0.028806483	0.121193517
4.5	0.112	0.023914015	0.088085985	0.16	0.024004947	0.135995053
4.6	0.118	0.024173429	0.093826571	0.165	0.024089549	0.140910451

Time (s)	Sensor R1 (m)	Rift R1 (m)	Error R1 (m)	Sensor R2 (m)	Rift R2 (m)	Error R2 (m)
4.7	0.122	0.026141373	0.095858627	0.172	0.025993427	0.146006573
4.8	0.138	0.040193133	0.097806867	0.177	0.040193133	0.136806867
4.9	0.14	0.035083983	0.104916017	0.188	0.035083983	0.152916017
5	0.145	0.03767968	0.10732032	0.195	0.037918847	0.157081153
5.1	0.15	0.040832408	0.109167592	0.199	0.040718146	0.158281854
5.2	0.156	0.04877162	0.10722838	0.21	0.048686828	0.161313172
5.3	0.161	0.04608028	0.11491972	0.217	0.046049163	0.170950837
5.4	0.172	0.054645292	0.117354708	0.222	0.054431181	0.167568819
5.5	0.176	0.049188402	0.126811598	0.227	0.048873279	0.178126721
5.6	0.188	0.054373641	0.133626359	0.233	0.054406125	0.178593875
5.7	0.191	0.052067999	0.138932001	0.239	0.052067999	0.186932001
5.8	0.202	0.063094296	0.138905704	0.248	0.062951088	0.185048912
5.9	0.205	0.081672847	0.123327153	0.252	0.081350483	0.170649517
6	0.217	0.074613824	0.142386176	0.265	0.074176304	0.190823696
6.1	0.227	0.072171964	0.154828036	0.272	0.071979217	0.200020783
6.2	0.235	0.080630705	0.154369295	0.277	0.078317381	0.198682619
6.3	0.241	0.091242477	0.149757523	0.284	0.091079488	0.192920512
6.4	0.251	0.08688128	0.16411872	0.3	0.086946532	0.213053468
6.5	0.256	0.086889267	0.169110733	0.307	0.08675006	0.22024994
6.6	0.262	0.093640544	0.168359456	0.314	0.093216822	0.220783178
6.7	0.268	0.081526347	0.186473653	0.321	0.082839675	0.238160325
6.8	0.279	0.104093179	0.174906821	0.327	0.104093179	0.222906821
6.9	0.291	0.115475275	0.175524725	0.331	0.114383794	0.216616206
7	0.302	0.13458319	0.16741681	0.341	0.135267034	0.205732966
7.1	0.305	0.120560303	0.184439697	0.345	0.120852605	0.224147395
7.2	0.305	0.128460124	0.176539876	0.359	0.128460124	0.230539876
7.3	0.317	0.149668783	0.167331217	0.368	0.149979874	0.218020126
7.4	0.321	0.142401159	0.178598841	0.373	0.141999796	0.231000204
7.5	0.32	0.13492471	0.18507529	0.383	0.134732276	0.248267724
7.6	0.329	0.1574945	0.1715055	0.392	0.1574945	0.2345055
7.7	0.338	0.173774347	0.164225653	0.396	0.173774347	0.222225653
7.8	0.346	0.166650921	0.179349079	0.407	0.166650921	0.240349079
7.9	0.35	0.178932115	0.171067885	0.406	0.178574309	0.227425691
8	0.36	0.191905379	0.168094621	0.415	0.19261831	0.22238169
8.1	0.366	0.202427804	0.163572196	0.423	0.201551855	0.221448145
8.2	0.375	0.185326576	0.189673424	0.433	0.184509948	0.248490052
8.3	0.38	0.225146607	0.154853393	0.442	0.225803554	0.216196446
8.4	0.389	0.224648356	0.164351644	0.448	0.223380983	0.224619017
8.5	0.395	0.231432557	0.163567443	0.454	0.231432557	0.222567443
8.6	0.407	0.215566769	0.191433231	0.463	0.215566769	0.247433231
8.7	0.412	0.251863867	0.160136133	0.472	0.251420259	0.220579741
8.8	0.425	0.262202412	0.162797588	0.477	0.261913598	0.215086402
8.9	0.428	0.295527399	0.132472601	0.486	0.290944457	0.195055543
9	0.442	0.269863129	0.172136871	0.491	0.270151675	0.220848325
9.1	0.448	0.315029651	0.132970349	0.498	0.315029651	0.182970349
9.2	0.457	0.288347036	0.168652964	0.512	0.288347036	0.223652964

Time (s)	Sensor R1 (m)	Rift R1 (m)	Error R1 (m)	Sensor R2 (m)	Rift R2 (m)	Error R2 (m)
9.3	0.464	0.298432887	0.165567113	0.519	0.299650341	0.219349659
9.4	0.469	0.339211971	0.129788029	0.526	0.34066385	0.18533615
9.5	0.471	0.299052358	0.171947642	0.531	0.298646212	0.232353788
9.6	0.483	0.329885036	0.153114964	0.543	0.330373049	0.212626951
9.7	0.488	0.346439868	0.141560132	0.542	0.343232542	0.198767458
9.8	0.494	0.334168196	0.159831804	0.554	0.332914829	0.221085171
9.9	0.497	0.384987205	0.112012795	0.558	0.384987205	0.173012795
10	0.508	0.381853193	0.126146807	0.572	0.381853193	0.190146807
10.1	0.515	0.402367532	0.112632468	0.577	0.402367532	0.174632468
10.2	0.522	0.426817805	0.095182195	0.591	0.426817805	0.164182195
10.3	0.527	0.419159859	0.107840141	0.6	0.417383999	0.182616001
10.4	0.543	0.449942023	0.093057977	0.606	0.446449846	0.159550154
10.5	0.545	0.473780334	0.071219666	0.619	0.473919004	0.145080996
10.6	0.553	0.479306579	0.073693421	0.624	0.479306579	0.144693421
10.7	0.558	0.50590539	0.05209461	0.625	0.503419995	0.121580005
10.8	0.569	0.504976988	0.064023012	0.638	0.504976988	0.133023012
10.9	0.571	0.508781731	0.062218269	0.642	0.510252059	0.131747941
11	0.586	0.526746452	0.059253548	0.657	0.524110079	0.132889921
11.1	0.594	0.586092234	0.007907766	0.662	0.586787343	0.075212657
11.2	0.601	0.57787627	0.02312373	0.669	0.583278179	0.085721821
11.3	0.605	0.629086554	0.024086554	0.672	0.626926005	0.045073995
11.4	0.616	0.62288487	0.00688487	0.677	0.62288487	0.05411513
11.5	0.62	0.65625596	0.03625596	0.683	0.657345533	0.025654467
11.6	0.626	0.690295517	0.064295517	0.689	0.687244058	0.001755942
11.7	0.634	0.628958523	0.005041477	0.692	0.630180001	0.061819999
11.8	0.637	0.632053077	0.004946923	0.701	0.63085562	0.07014438
11.9	0.642	0.72539717	0.08339717	0.706	0.725634813	0.019634813
12	0.646	0.755542397	0.109542397	0.712	0.755975187	0.043975187
12.1	0.653	0.754281223	0.101281223	0.72	0.746197939	0.026197939
12.2	0.66	0.822029233	0.162029233	0.727	0.818181157	0.091181157
12.3	0.664	0.847581387	0.183581387	0.726	0.847581387	0.121581387
12.4	0.667	0.844753742	0.177753742	0.735	0.841978729	0.106978729
12.5	0.672	0.876969516	0.204969516	0.739	0.876488328	0.137488328
12.6	0.678	0.887047231	0.209047231	0.752	0.888148069	0.136148069
12.7	0.685	0.926964939	0.241964939	0.75	0.927511632	0.177511632
12.8	0.697	0.937229335	0.240229335	0.761	0.937623262	0.176623262
12.9	0.705	0.979942858	0.274942858	0.77	0.975957274	0.205957274
13	0.72	1.002187252	0.282187252	0.78	1.002115011	0.222115011
13.1	0.727	1.027775288	0.300775288	0.786	1.026355028	0.240355028
13.2	0.736	1.06432879	0.32832879	0.801	1.061660171	0.260660171
13.3	0.738	1.086969972	0.348969972	0.808	1.086969972	0.278969972
13.4	0.743	1.097196341	0.354196341	0.817	1.095849037	0.278849037
13.5	0.756	1.177902222	0.421902222	0.823	1.178083301	0.355083301
13.6	0.768	1.209229469	0.441229469	0.837	1.21191895	0.37491895
13.7	0.779	1.190587759	0.411587759	0.842	1.190587759	0.348587759
13.8	0.792	1.256781459	0.464781459	0.852	1.256968975	0.404968975

Time (s)	Sensor R1 (m)	Rift R1 (m)	Error R1 (m)	Sensor R2 (m)	Rift R2 (m)	Error R2 (m)
13.9	0.795	1.274364233	0.479364233	0.862	1.275255322	0.413255322
14	0.803	1.272355556	0.469355556	0.87	1.271496177	0.401496177
14.1	0.816	1.320327878	0.504327878	0.877	1.319581509	0.442581509
14.2	0.826	1.376215339	0.550215339	0.89	1.376678228	0.486678228
14.3	0.823	1.424155831	0.601155831	0.889	1.421665788	0.532665788
14.4	0.834	1.468171358	0.634171358	0.901	1.469285846	0.568285846
14.5	0.844	1.466459036	0.622459036	0.902	1.468401313	0.566401313
14.6	0.851	1.527528524	0.676528524	0.913	1.527657032	0.614657032
14.7	0.858	1.550323009	0.692323009	0.918	1.553693175	0.635693175
14.8	0.864	1.578776598	0.714776598	0.926	1.580889583	0.654889583
14.9	0.869	1.696534872	0.827534872	0.93	1.696015596	0.766015596
15	0.878	1.70267725	0.82467725	0.941	1.701584816	0.760584816

Time (s)	Sensor R3 (m)	Rift R3 (m)	Error R3 (m)	Sensor R4 (m)	Rift R4 (m)	Error R4 (m)
0.1	-0.004	-2.53513E-16	0.004	-0.002	-2.53513E-16	0.002
0.2	-0.003	-8.80074E-10	0.002999999	-0.004	-3.08762E-10	0.004
0.3	-0.001	2.30889E-08	0.001000023	-0.003	2.30889E-08	0.003000023
0.4	-0.008	2.10338E-07	0.00800021	-0.002	2.10338E-07	0.00200021
0.5	-0.004	1.03224E-06	0.004001032	-0.001	1.03224E-06	0.001001032
0.6	-0.005	2.88007E-06	0.00500288	0.001	3.035E-06	0.000996965
0.7	-0.002	5.77934E-06	0.002005779	-0.003	5.77112E-06	0.003005771
0.8	-0.007	7.80442E-06	0.007007804	-0.001	7.97797E-06	0.001007978
0.9	0	1.30092E-05	1.30092E-05	-0.004	1.31414E-05	0.004013141
1	-0.002	2.07439E-05	0.002020744	-0.003	2.13951E-05	0.003021395
1.1	-0.004	3.10729E-05	0.004031073	-0.002	3.12112E-05	0.002031211
1.2	0	4.79087E-05	4.79087E-05	0.001	4.79087E-05	0.000952091
1.3	-0.003	7.00761E-05	0.003070076	-0.004	7.1112E-05	0.004071112
1.4	-0.004	8.33436E-05	0.004083344	-0.001	8.33436E-05	0.001083344
1.5	-0.005	0.000105985	0.005105985	-0.004	0.00010703	0.00410703
1.6	0.001	0.000222807	0.000777193	-0.001	0.000222807	0.001222807
1.7	0.005	0.000280807	0.004719193	0.004	0.000280807	0.003719193
1.8	0.009	0.000381223	0.008618777	0	0.000381223	0.000381223
1.9	0.01	0.000350476	0.009649524	0.003	0.000350476	0.002649524
2	0.012	0.000410469	0.011589531	-0.004	0.000410469	0.004410469
2.1	0.014	0.000836366	0.013163634	0.001	0.000836366	0.000163634
2.2	0.012	0.001478523	0.010521477	0.005	0.00148258	0.00351742
2.3	0.021	0.001918987	0.019081013	0.008	0.001918987	0.006081013
2.4	0.017	0.002360013	0.014639987	0.01	0.002318957	0.007681043
2.5	0.02	0.00318776	0.01681224	0.013	0.00318776	0.00981224
2.6	0.028	0.004552359	0.023447641	0.021	0.004544322	0.016455678
2.7	0.036	0.005385292	0.030614708	0.025	0.005374096	0.019625904
2.8	0.04	0.00712481	0.03287519	0.028	0.007138394	0.020861606
2.9	0.045	0.006950284	0.038049716	0.032	0.006980424	0.025019576
3	0.042	0.007626242	0.034373758	0.044	0.007626242	0.036373758
3.1	0.055	0.006705619	0.048294381	0.041	0.006778778	0.034221222
3.2	0.061	0.00794117	0.05305883	0.053	0.007934316	0.045065684
3.3	0.064	0.009140746	0.054859254	0.056	0.009140746	0.046859254
3.4	0.066	0.011261802	0.054738198	0.064	0.011246425	0.052753575
3.5	0.08	0.011360764	0.068639236	0.068	0.011360764	0.056639236
3.6	0.084	0.013417472	0.070582528	0.075	0.013403929	0.061596071
3.7	0.093	0.01299101	0.08000899	0.084	0.01299101	0.07100899
3.8	0.099	0.01612466	0.08287534	0.092	0.016180485	0.075819515
3.9	0.107	0.018344261	0.088655739	0.1	0.018344261	0.081655739
4	0.112	0.017840372	0.094159628	0.106	0.018011697	0.087988303
4.1	0.12	0.017720878	0.102279122	0.107	0.017835448	0.089164552
4.2	0.127	0.020166794	0.106833206	0.118	0.020166794	0.097833206
4.3	0.138	0.022711828	0.115288172	0.12	0.022711828	0.097288172
4.4	0.14	0.028949793	0.111050207	0.133	0.02887808	0.10412192
4.5	0.155	0.023866825	0.131133175	0.137	0.023866825	0.113133175
4.6	0.161	0.024173429	0.136826571	0.15	0.024173429	0.125826571

Time (s)	Sensor R3 (m)	Rift R3 (m)	Error R3 (m)	Sensor R4 (m)	Rift R4 (m)	Error R4 (m)
4.7	0.172	0.025993427	0.146006573	0.152	0.026141373	0.125858627
4.8	0.18	0.040247772	0.139752228	0.162	0.040247772	0.121752228
4.9	0.184	0.035083983	0.148916017	0.164	0.035083983	0.128916017
5	0.19	0.037454084	0.152545916	0.179	0.037454084	0.141545916
5.1	0.196	0.040842958	0.155157042	0.177	0.040832408	0.136167592
5.2	0.206	0.04877162	0.15722838	0.188	0.048874218	0.139125782
5.3	0.211	0.04608028	0.16491972	0.193	0.04608028	0.14691972
5.4	0.217	0.054645292	0.162354708	0.203	0.054645292	0.148354708
5.5	0.231	0.048875995	0.182124005	0.207	0.049063221	0.157936779
5.6	0.235	0.054373641	0.180626359	0.214	0.054410547	0.159589453
5.7	0.247	0.052067999	0.194932001	0.223	0.052067999	0.170932001
5.8	0.252	0.062962547	0.189037453	0.236	0.063094296	0.172905704
5.9	0.26	0.080812052	0.179187948	0.241	0.081672847	0.159327153
6	0.267	0.074382722	0.192617278	0.248	0.074382722	0.173617278
6.1	0.274	0.072092712	0.201907288	0.252	0.072171964	0.179828036
6.2	0.276	0.079072125	0.196927875	0.264	0.079850838	0.184149162
6.3	0.286	0.091169916	0.194830084	0.267	0.091169916	0.175830084
6.4	0.291	0.08688128	0.20411872	0.281	0.086671419	0.194328581
6.5	0.302	0.086889267	0.215110733	0.282	0.086889267	0.195110733
6.6	0.319	0.094285019	0.224714981	0.299	0.093640544	0.205359456
6.7	0.326	0.081526347	0.244473653	0.304	0.081526347	0.222473653
6.8	0.334	0.103958108	0.230041892	0.312	0.10373804	0.20826196
6.9	0.341	0.115475275	0.225524725	0.313	0.11488758	0.19811242
7	0.345	0.134949729	0.210050271	0.325	0.134949729	0.190050271
7.1	0.355	0.120560303	0.234439697	0.331	0.120852605	0.210147395
7.2	0.364	0.128460124	0.235539876	0.334	0.128460124	0.205539876
7.3	0.366	0.149668783	0.216331217	0.338	0.149080694	0.188919306
7.4	0.376	0.142276019	0.233723981	0.352	0.142276019	0.209723981
7.5	0.387	0.134694144	0.252305856	0.358	0.134694144	0.223305856
7.6	0.401	0.158718988	0.242281012	0.372	0.158071369	0.213928631
7.7	0.405	0.173727781	0.231272219	0.374	0.173644766	0.200355234
7.8	0.407	0.166650921	0.240349079	0.38	0.166650921	0.213349079
7.9	0.415	0.179499686	0.235500314	0.39	0.178932115	0.211067885
8	0.421	0.192198992	0.228801008	0.397	0.192198992	0.204801008
8.1	0.427	0.203279033	0.223720967	0.405	0.203279033	0.201720967
8.2	0.433	0.185326576	0.247673424	0.411	0.185326576	0.225673424
8.3	0.442	0.225803554	0.216196446	0.417	0.225146607	0.191853393
8.4	0.451	0.224648356	0.226351644	0.421	0.224648356	0.196351644
8.5	0.454	0.231282726	0.222717274	0.427	0.231320545	0.195679455
8.6	0.463	0.215566769	0.247433231	0.438	0.215566769	0.222433231
8.7	0.473	0.251711845	0.221288155	0.445	0.251711845	0.193288155
8.8	0.478	0.262202412	0.215797588	0.455	0.262202412	0.192797588
8.9	0.486	0.293234974	0.192765026	0.462	0.295527399	0.166472601
9	0.497	0.269926161	0.227073839	0.469	0.270151675	0.198848325
9.1	0.503	0.310815841	0.192184159	0.473	0.310815841	0.162184159
9.2	0.507	0.290424347	0.216575653	0.489	0.290424347	0.198575653

Time (s)	Sensor R3 (m)	Rift R3 (m)	Error R3 (m)	Sensor R4 (m)	Rift R4 (m)	Error R4 (m)
9.3	0.515	0.296996206	0.218003794	0.488	0.296996206	0.191003794
9.4	0.519	0.337862074	0.181137926	0.493	0.337862074	0.155137926
9.5	0.53	0.299052358	0.230947642	0.496	0.299525321	0.196474679
9.6	0.533	0.329885036	0.203114964	0.506	0.329885036	0.176114964
9.7	0.547	0.344978452	0.202021548	0.513	0.344978452	0.168021548
9.8	0.552	0.334168196	0.217831804	0.525	0.333864182	0.191135818
9.9	0.562	0.384987205	0.177012795	0.527	0.384987205	0.142012795
10	0.568	0.381948382	0.186051618	0.536	0.381853193	0.154146807
10.1	0.575	0.402367532	0.172632468	0.544	0.402367532	0.141632468
10.2	0.583	0.426817805	0.156182195	0.552	0.426817805	0.125182195
10.3	0.588	0.418244392	0.169755608	0.557	0.419159859	0.137840141
10.4	0.594	0.451282918	0.142717082	0.562	0.449942023	0.112057977
10.5	0.605	0.473780334	0.131219666	0.567	0.473780334	0.093219666
10.6	0.609	0.479306579	0.129693421	0.574	0.477577329	0.096422671
10.7	0.621	0.50590539	0.11509461	0.583	0.50590539	0.07709461
10.8	0.629	0.504976988	0.124023012	0.59	0.504976988	0.085023012
10.9	0.639	0.509735703	0.129264297	0.594	0.508781731	0.085218269
11	0.637	0.524110079	0.112889921	0.604	0.526746452	0.077253548
11.1	0.653	0.585672438	0.067327562	0.606	0.585672438	0.020327562
11.2	0.655	0.57787627	0.07712373	0.611	0.57787627	0.03312373
11.3	0.664	0.629086554	0.034913446	0.614	0.629086554	0.015086554
11.4	0.667	0.62524873	0.04175127	0.623	0.62524873	0.00224873
11.5	0.678	0.65625596	0.02174404	0.62	0.65625596	0.03625596
11.6	0.68	0.690295517	0.010295517	0.632	0.690295517	0.058295517
11.7	0.692	0.628958523	0.063041477	0.635	0.628958523	0.006041477
11.8	0.695	0.632053077	0.062946923	0.638	0.63517046	0.00282954
11.9	0.699	0.725144506	0.026144506	0.644	0.725144506	0.081144506
12	0.703	0.75459975	0.05159975	0.652	0.754942596	0.102942596
12.1	0.708	0.749954164	0.041954164	0.657	0.749954164	0.092954164
12.2	0.703	0.820289612	0.117289612	0.667	0.822029233	0.155029233
12.3	0.716	0.847581387	0.131581387	0.671	0.847581387	0.176581387
12.4	0.721	0.847232401	0.126232401	0.681	0.844753742	0.163753742
12.5	0.728	0.876969516	0.148969516	0.688	0.876969516	0.188969516
12.6	0.731	0.887047231	0.156047231	0.707	0.88612622	0.17912622
12.7	0.741	0.926964939	0.185964939	0.712	0.926964939	0.214964939
12.8	0.748	0.937623262	0.189623262	0.719	0.937229335	0.218229335
12.9	0.75	0.978342474	0.228342474	0.724	0.979942858	0.255942858
13	0.759	1.002187252	0.243187252	0.736	1.002385378	0.266385378
13.1	0.769	1.027775288	0.258775288	0.741	1.028897762	0.287897762
13.2	0.766	1.066903234	0.300903234	0.748	1.06432879	0.31632879
13.3	0.784	1.087077498	0.303077498	0.754	1.086969972	0.332969972
13.4	0.791	1.097196341	0.306196341	0.771	1.097196341	0.326196341
13.5	0.807	1.178083301	0.371083301	0.779	1.177902222	0.398902222
13.6	0.814	1.20692265	0.39292265	0.786	1.20692265	0.42092265
13.7	0.819	1.190609336	0.371609336	0.796	1.190609336	0.394609336
13.8	0.827	1.256406546	0.429406546	0.808	1.257050276	0.449050276

Time (s)	Sensor R3 (m)	Rift R3 (m)	Error R3 (m)	Sensor R4 (m)	Rift R4 (m)	Error R4 (m)
13.9	0.839	1.273682475	0.434682475	0.815	1.274364233	0.459364233
14	0.838	1.272355556	0.434355556	0.823	1.272355556	0.449355556
14.1	0.851	1.319581509	0.468581509	0.829	1.321295619	0.492295619
14.2	0.861	1.376215339	0.515215339	0.839	1.376215339	0.537215339
14.3	0.875	1.421665788	0.546665788	0.841	1.421665788	0.580665788
14.4	0.884	1.468171358	0.584171358	0.854	1.467086792	0.613086792
14.5	0.896	1.468401313	0.572401313	0.859	1.466459036	0.607459036
14.6	0.902	1.527528524	0.625528524	0.876	1.527528524	0.651528524
14.7	0.908	1.553693175	0.645693175	0.88	1.550323009	0.670323009
14.8	0.91	1.578776598	0.668776598	0.889	1.578776598	0.689776598
14.9	0.925	1.696352959	0.771352959	0.896	1.696534872	0.800534872
15	0.93	1.70267725	0.77267725	0.9	1.70267725	0.80267725

Time (s)	Sensor R5 (m)	Rift R5 (m)	Error R5 (m)
0.1	-0.006	4.28596E-11	0.006
0.2	-0.005	5.53831E-09	0.005000006
0.3	-0.003	7.41534E-09	0.003000007
0.4	-0.008	-1.60726E-07	0.007999839
0.5	-0.004	-7.03728E-07	0.003999296
0.6	-0.009	-9.5614E-07	0.008999044
0.7	-0.006	-1.95374E-06	0.005998046
0.8	-0.002	-2.94871E-06	0.001997051
0.9	-0.005	-2.85147E-06	0.004997149
1	-0.003	-6.86441E-06	0.002993136
1.1	-0.004	-1.33047E-05	0.003986695
1.2	-0.007	-2.06293E-05	0.006979371
1.3	-0.001	-2.93702E-05	0.00097063
1.4	-0.006	-4.39362E-05	0.005956064
1.5	-0.004	-4.88226E-05	0.003951177
1.6	0	-8.09106E-05	8.09106E-05
1.7	0.001	0.000291775	0.000708225
1.8	0.005	0.000319605	0.004680395
1.9	0.006	0.000402087	0.005597913
2	0.007	0.000251894	0.006748106
2.1	0.014	0.000799594	0.013200406
2.2	0.022	0.001176585	0.020823415
2.3	0.023	0.001041148	0.021958852
2.4	0.028	0.002295977	0.025704023
2.5	0.033	0.003151481	0.029848519
2.6	0.032	0.001661753	0.030338247
2.7	0.046	0.002239282	0.043760718
2.8	0.057	0.005430547	0.051569453
2.9	0.062	0.00681146	0.05518854
3	0.069	0.007597166	0.061402834
3.1	0.074	0.005951905	0.068048095
3.2	0.089	0.008602731	0.080397269
3.3	0.09	0.008140153	0.081859847
3.4	0.103	0.009761571	0.093238429
3.5	0.108	0.013662559	0.094337441
3.6	0.111	0.011424094	0.099575906
3.7	0.121	0.013017219	0.107982781
3.8	0.133	0.01134262	0.12165738
3.9	0.139	0.019252829	0.119747171
4	0.149	0.021170441	0.127829559
4.1	0.158	0.018611807	0.139388193
4.2	0.169	0.023134375	0.145865625
4.3	0.176	0.021274958	0.154725042
4.4	0.186	0.030238183	0.155761817
4.5	0.187	0.026818641	0.160181359
4.6	0.202	0.01807986	0.18392014

Time (s)	Sensor R5 (m)	Rift R5 (m)	Error R5 (m)
4.7	0.21	0.030343637	0.179656363
4.8	0.216	0.040395886	0.175604114
4.9	0.221	0.034077667	0.186922333
5	0.236	0.022841023	0.213158977
5.1	0.239	0.043815751	0.195184249
5.2	0.254	0.048901327	0.205098673
5.3	0.257	0.038242597	0.218757403
5.4	0.268	0.027026344	0.240973656
5.5	0.276	0.041349761	0.234650239
5.6	0.285	0.041105166	0.243894834
5.7	0.293	0.047698516	0.245301484
5.8	0.296	0.064239003	0.231760997
5.9	0.303	0.063322268	0.239677732
6	0.318	0.051225863	0.266774137
6.1	0.328	0.070216432	0.257783568
6.2	0.33	0.061455619	0.268544381
6.3	0.342	0.05230533	0.28969467
6.4	0.358	0.066791542	0.291208458
6.5	0.361	0.104109392	0.256890608
6.6	0.37	0.11483448	0.25516552
6.7	0.379	0.096234351	0.282765649
6.8	0.394	0.091192454	0.302807546
6.9	0.404	0.111029148	0.292970852
7	0.401	0.107872173	0.293127827
7.1	0.414	0.101677284	0.312322716
7.2	0.432	0.100171193	0.331828807
7.3	0.439	0.156049401	0.282950599
7.4	0.455	0.166329637	0.288670363
7.5	0.458	0.160061613	0.297938387
7.6	0.465	0.121079676	0.343920324
7.7	0.473	0.125433087	0.347566913
7.8	0.48	0.10790965	0.37209035
7.9	0.491	0.144580081	0.346419919
8	0.504	0.143925086	0.360074914
8.1	0.516	0.160088941	0.355911059
8.2	0.532	0.230574638	0.301425362
8.3	0.537	0.215744555	0.321255445
8.4	0.54	0.118548557	0.421451443
8.5	0.546	0.16900292	0.37699708
8.6	0.563	0.160074577	0.402925423
8.7	0.568	0.186829388	0.381170612
8.8	0.573	0.126374453	0.446625547
8.9	0.584	0.191457152	0.392542848
9	0.598	0.247925967	0.350074033
9.1	0.602	0.245852739	0.356147261
9.2	0.605	0.182452232	0.422547768

Time (s)	Sensor R5 (m)	Rift R5 (m)	Error R5 (m)
9.3	0.615	0.21018818	0.40481182
9.4	0.616	0.207009256	0.408990744
9.5	0.622	0.186354831	0.435645169
9.6	0.626	0.215934023	0.410065977
9.7	0.638	0.211041316	0.426958684
9.8	0.645	0.197118253	0.447881747
9.9	0.656	0.256625146	0.399374854
10	0.662	0.298639178	0.363360822
10.1	0.676	0.309241176	0.366758824
10.2	0.682	0.303179592	0.378820408
10.3	0.686	0.25456965	0.43143035
10.4	0.696	0.230749652	0.465250348
10.5	0.7	0.282745481	0.417254519
10.6	0.706	0.273492783	0.432507217
10.7	0.715	0.333506107	0.381493893
10.8	0.733	0.369335175	0.363664825
10.9	0.739	0.424352527	0.314647473
11	0.748	0.478474557	0.269525443
11.1	0.753	0.360695809	0.392304191
11.2	0.758	0.341302574	0.416697426
11.3	0.763	0.353337497	0.409662503
11.4	0.773	0.343670905	0.429329095
11.5	0.777	0.345739871	0.431260129
11.6	0.783	0.406570494	0.376429506
11.7	0.785	0.365263432	0.419736568
11.8	0.795	0.452273577	0.342726423
11.9	0.805	0.501960576	0.303039424
12	0.814	0.546647847	0.267352153
12.1	0.823	0.541579187	0.281420813
12.2	0.825	0.467741191	0.357258809
12.3	0.826	0.494170159	0.331829841
12.4	0.841	0.478747785	0.362252215
12.5	0.84	0.584216952	0.255783048
12.6	0.843	0.470632195	0.372367805
12.7	0.856	0.591582954	0.264417046
12.8	0.854	0.794046402	0.059953598
12.9	0.865	0.639378369	0.225621631
13	0.866	0.684816182	0.181183818
13.1	0.873	0.642030835	0.230969165
13.2	0.879	0.619136393	0.259863607
13.3	0.887	0.716303289	0.170696711
13.4	0.89	0.712046027	0.177953973
13.5	0.909	0.726117551	0.182882449
13.6	0.923	0.859652579	0.063347421
13.7	0.931	0.904165626	0.026834374
13.8	0.94	0.76552999	0.17447001

Time (s)	Sensor R5 (m)	Rift R5 (m)	Error R5 (m)
13.9	0.953	0.75996846	0.19303154
14	0.96	0.800736189	0.159263811
14.1	0.964	0.798389733	0.165610267
14.2	0.975	0.841635168	0.133364832
14.3	0.979	0.836643994	0.142356006
14.4	0.991	0.877027035	0.113972965
14.5	1.007	1.066090226	0.059090226
14.6	1.013	1.01080215	0.00219785
14.7	1.02	0.957376897	0.062623103
14.8	1.039	0.931788146	0.107211854
14.9	1.04	0.914842367	0.125157633
15	1.048	0.973914623	0.074085377