Otto-von-Guericke University Magdeburg

Faculty of Computer Science

Master Thesis

Teammate Tracking using Bayesian Sensor Fusion of Certainty Grids

Author:

Markus Hempel

October 21, 2022

Supervisor:

Prof. Dr.-Ing. habil. Sanaz Mostaghim
Chair of Computational Intelligence
Otto-von-Guericke University Magdeburg

Advisors:

Dr.-Ing. Christoph Steup
M.Sc. Sebastian Mai

Hempel, Markus:
Teammate Tracking using Bayesian Sensor Fusion of Certainty Grids
Master Thesis, Otto-von-Guericke University Magdeburg, 2022.

Abstract

Nowadays, swarm robotics grows importance because of their wide application field, for example: Internet of things or search scenarios such as radioactive source search or missing person search. In swarm robotics, teammate tracking is indispensable to ensure an intelligent and efficient problem-solving in swarm behaviour.

In this thesis, a new approach for teammate tracking in swarm robotics is presented. To maximize the confidence in the information gained by the sensors of the robots, the Bayes-Hempel Method for sensor fusion is introduced. The goal of this thesis is to experimentally validate the approach and to show the advantages compared to other state-of-the-art methods. This method maps sensor information into certainty grids, which are then fused using the Bayes theorem. With one resulting certainty grid per teammate, a probabilistic estimation on the position of the teammate is given. To gain information in different scenarios, an Ultra-Wideband sensor, which is reliable without line of sight, and a LIDAR sensor, which is very accurate in line of sight, are used.

For evaluation, a relative probability analysis and an absolute error calculation are performed. Those show, that the teammate can be tracked even when an obstacle is between both robots. The accuracy can compete with other state-of-the-art methods like the Kalman Filter. Furthermore, the method shows a high robustness and the possibility to adapt to different scenarios by manually tuning the parameters. Due to the approach using a prior mapping to certainty grids, a fusion of sensors with different modalities is possible. Thus, the method is applicable to many scenarios using multiple sensors. To improve the usability an implementation of an adaptive parameter behaviour would be possible. By archiving the certainty grids over multiple time steps, an estimation of movement is additionally possible. Therefore, the presented approach builds a new base for measurement processing in swarm robotics.

Contents

List of Figures

List of Tables

1. Introduction

In the last years, robotic swarms made of many autonomous operating individuals have seen increased applications. The knowledge about the environment and the location of other swarm members is crucial for most applications like the internet of things (IoT)[1], flying drone fireworks or robots utilized in search scenarios[2]. With this knowledge, collisions can be avoided and a correct distribution for best results can be maintained[3]. To also maintain self-sufficiency of swarm individuals, each one has to gather information about the environment themselves[4]. Therefore, it does not rely on an absolute localization like the Global Positioning System(GPS), and is more universally applicable. Here, sensor fusion plays an important role to get maximum information with high quality out of the environment, utilizing the limited possibilities of each swarm member. The technique combines single sensor information into a more informative result[5]. It finds wide application in many areas of robotics such as object recognition, environment mapping, and localization and is therefore regarded as the most appropriate method to track objects and determine their locations[6]. One specific application of this technique is teammate tracking for robotic swarms, which is covered in this work. Teammate tracking is especially useful to locate certain members of the swarm. With this information, high-level behaviours such as role decisions and directed interactions are possible[7].

In this thesis, we want to fuse different sensor information to determine the location of teammates. First, we want to map the findings of each sensor into modified certainty grids of the environment as a relative localization. These grids contain probability values representing the certainty on the location of teammates in each cell. After the creation of these grids, a sensor fusion algorithm utilizing the Bayes theorem is designed to combine them. Thereby, a resulting certainty grid of the environment containing all sensor information about the teammate is yielded.

To analyse and evaluate this result, the following research questions are answered.

Research Question 1 (RQ1)**:**
How can the quality of teammate tracking be measured?

Research Question 2 (RQ2)**:**
How big is the quantitative advantage or the improvement compared to the state of the art?

Research Question 3 (RQ3)**:**
How big is the influence of parameter changes?

Research Question 4 (RQ4)**:**
How high is the robustness against different kinds of faults?

The first question aims to design a suited metric for evaluating the performance of the sensor fusion approach. This is related to all further questions and enables to give well-founded concrete statements as answers. The second question is concerned with the overall performance of the fusion approach in relation to other state-of-the-art fusion techniques. Thereby, the designed evaluation metric is used to compare measurable and expressive results. The third and fourth question relate to properties of the designed approach. Here, the influence of parameter changes allows statements to be made about the adaptability of the approach. And the robustness against different kind of faults investigates about disturbances and the influence of those. Hence, the third and fourth question allow for conclusions to be drawn about the limitations and therefore possible applications of this sensor fusion approach by the user.

Structure of Work

First, in Chapter 2, the background needed as a foundation to this work is given and explained. Second, in Chapter 3, the related work is presented. Different sensors, which are able to track a teammate, are described and compared regarding their suitability for the use case. Afterwards, other sensor fusion techniques are described to give an overview on the state of the art. Third, in Chapter 4, the previously established knowledge is used to explain the procedure of tackling the stated problem and the design of a suited approach, the Bayes-Hempel method. Fourth, in Chapter 5, the environmental setup used to conduct experiments in is described, a suitable evaluation metric is introduced and ultimately the conducted experiments with the goal to answer the research questions are described and evaluated. Lastly, in Chapter 6, the laid out work is summarized and concluded. Additionally, an outlook on possible future work is given.

2. Background

In this chapter, the base of this work is clarified. In the beginning, the properties of a robotic swarm are described. Afterwards, a certainty grid as a mean of representation is introduced. Then, Sensor Fusion is described regarding definitions, benefits, levels, and types, followed by the basics of the Bayes theorem and the application in sensor fusion. Finally, it is described, how the requirements for the application of the Bayes theorem are determined.

2.1 Robotic Swarm Model

A swarm in nature consists of two or more independent individuals with a collective behaviour. In a robot environment, which is the main application field of this work, the individuals are single robots. And further, the robotic swarm applies swarm intelligence as a collective behaviour of decentralized, self-organized systems. Dorigo et al.[8][9] defined the emerging topic of swarm robotics as 'the study of how collectively intelligent behaviour can emerge from local interactions of a large number of relatively simple physically embodied agents'.
In a moving robotic swarm, the robot's position and movement mainly rely on every individual of the swarm and the robot's surroundings. Therefore, an individual must keep track of its local environment and especially other members of the swarm to avoid collisions and to calculate its next position and movement. Hence, some kind of tracking or localization between the swarm individuals is mandatory. This can be realized by using the earlier mentioned local interactions and can result in a collision free movement planning, navigation or other collective and intelligent behaviour of the swarm.
In this work, every robot is seen as an agent with specific sensors and actuators[10]. Here, the actuators of every robot enable them to move around in the given environment and the applied sensors, execute the local interactions by detecting the environment and especially changes in it. So in an application with multiple robots, the change in the environment mainly consists of the individual moving itself or other members of the swarm move around it. Therefore, the velocity of the environmental change is equal to the movement speed of a robot for most parts. Hence, a sufficient

sensor sampling rate according to the maximum movement speed a robot can achieve is needed. It also follows, that data sizes and computation time need to be adjusted according to the application and its speed of environmental change.

2.2 Certainty Grid

The information on the likelihood of residence of teammates need to be stored, transmitted and worked on by every component. Therefore, a suitable representation needs to be found, that combines the ability to easily locate a teammate inside the environment with a likelihood estimation and a lightweight storage solution. This thesis utilizes a certainty grid representation which is used and labelled by Moravec et al.[11] to be "a powerful and efficient unifying solution for sensor fusion, motion planning, landmark identification and, and many other central problems". Here, the first mentioned application is of main interest for this thesis.
A certainty grid used by Moravec is shown in Figure 2.1. Here, black parts represent

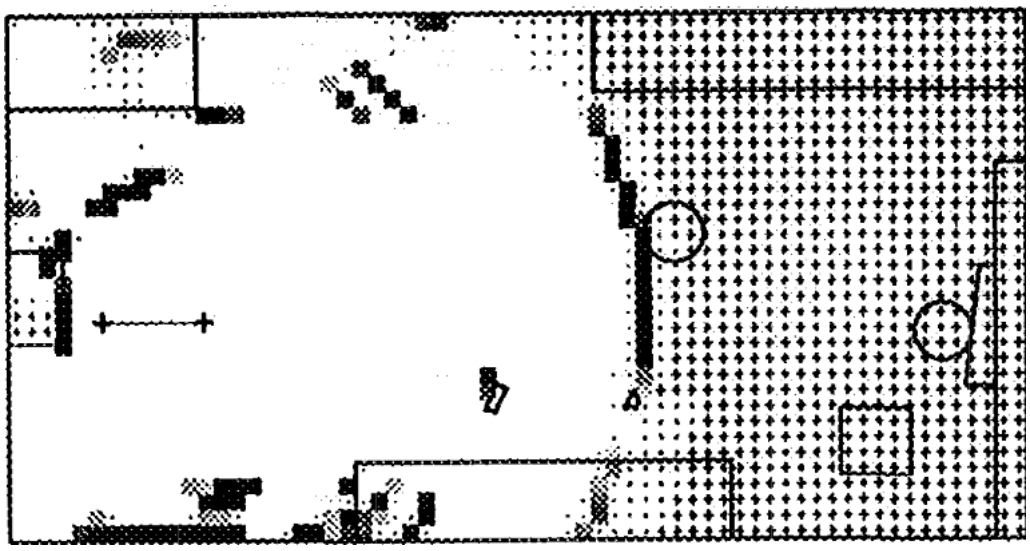

Figure 2.1: Certainty Grid used by Moravec - Occupied cells are marked by black squares, empty areas by dots fading to white representing certainty, and unknown territory by plus signs

occupied areas, plus signs are unknown areas and dots fading to white represent empty areas with certain likelihood. The lines are not part of the certainty grid itself, and simply show the outer bounds of objects in the environment for better visualization purposes.
Overall, a certainty grid is a 2D map the size of the given environment divided in cells. It maps the findings of obstacles all over the environment inside this certainty grid. Every cell stores the likelihood of residence of objects in this cell area. Cells with unknown likelihood get a fixed probability assigned, which represents a small likelihood of residence. Therefore, the grid is a discrete sampling of a continuous probability function. Hence, an initial uniform distribution of values in the grid is possible to achieve as well, which is especially useful to model an unknown starting state. Here, the sample size, and so to say the size of every cell, is determined by the application and should incorporate the sampling interval, the maximum velocity of change in the environment, and the teammate size to yield significant results.

With larger cell sizes, the grid gets more prone to sampling errors due to continuous information getting mapped onto discrete grid cells. To evade such errors and still save storage space, it is possible to have multiple resolutions in one grid. The less important parts can be mapped in a lower resolution than the important parts, so changes in environment can be detected when needed and the consumed storage space is still low. Additionally, further investigations are possible using interpolation to reconstruct the original continuous input.

2.3 Sensor/Data Fusion

2.3.1 Definitions

The fusion of information and data, so-called information fusion, is the key topic of this work. There are several definitions of fusions.

Information Fusion by Elmenreich[5]
Information fusion encompasses theory, techniques, and tools conceived and employed for exploiting the synergy in the information acquired from multiple sources (sensor, databases, information gathered by human, etc.) such that the resulting decision or action is in some sense better (qualitatively or quantitatively, in terms of accuracy, robustness, etc.) than would be possible if any of these sources were used individually without such synergy exploitation.

Data Fusion by Joint Directors of Laboratories workshop[12]
Data fusion is a multi-level procedure dealing with the association, correlation, integration of data and information from single and multiple sources to attain distinguished position, determine estimates and complete timely assessments of situations, threats, and their significance.

The main parts in both definitions include the existence of multiple sources of information and the processing of these sources to achieve new or better information, which would not be possible without the fusion process.
If the fusion happens in a robotic environment, sensor information are gathered as sources of information. This is a special case called sensor fusion with different definitions as well.

Sensor Fusion by Nagla et al.[13] and Castanedo[14]
Sensor fusion is the cooperative use of information provided by multiple sensors to aid in performing a function.

Sensor Fusion by Elmenreich[5]
Sensor fusion is a subset of information fusion, with sensor fusion being the combination of sensory data or data derived from sensory data such that the resulting information is in some sense better than what would be possible when these sources were used individually.

Sensor Fusion by Alatise and Hancke[6]
Sensor fusion is the integration of information from multiple sources to improve accuracy and quality content, also with the aim to reduce cost.

Especially, improving accuracy and simultaneously reducing cost is a desirable goal. Therefore, sensor fusion is well suited to expand the possibilities in a limited robotic environment due to sensor errors, robot size and component costs.

2.3.2 Benefits

As Alatise and Hancke[6] stated, sensor fusion has some significant advantages over the use of single sensor information. First, by combining the data gathered by multiple sensors, an increase in accuracy and reliability can be seen. This is caused by an overall reduction in uncertainty because each used sensor cancels out the errors of the other ones. So to say, the integration of more than one sensor leads to the complementation of the inadequacy of one sensor with another. Second, the usage of multiple sensors leads to an extended spatial and temporal coverage. Therefore, additional information can be derived from already existing sensor data. The stated advantages lead to an improved quality of information of fusion results, which can also lead to increased confidence in this information. Furthermore, by using multiple simple sensors instead of a single complex one, it is sometimes possible to reduce complexity and overall cost by simultaneously maintaining a feasible quality of results.

2.3.3 Fusion Level

There are different types of sensor fusion levels. The chosen architecture defines the hierarchical location, where the fusion happens. Elmenreich[5] differentiates between low-level, intermediate-level and high-level fusion. Kam et al.[15] describe the levels as follows. The low-level fusion is the direct integration of sensory data, resulting in parameter and state estimates used by planning and motion execution modules to generate command and control signals. Therefore, it is the level closest to the sensors. Contradictory to this, high-level fusion is the indirect integration of sensory data in hierarchical architectures, through command arbitration and integration of control signals, that are suggested by different modules. This means first the sensor data gets highly processed and afterwards the results get fused together. Lastly, as a way to combine both above-mentioned levels of fusion, intermediate-level or hybrid fusion is stated. Here, an architecture is created, that synthesizes command and control signals directly from sensory input, but does not make use of explicit construction of environmental models. Hence, the intermediate-level fusion combines the sensor data with light abstraction into the result.
As the low-level fusion combines raw data, the complexity and computation time is kept low. With increasing fusion level, the complexity and computation time increases, but also the extendability to further input sources as well as the abstraction of the result. Therefore, by choosing an appropriate fusion level, the best suited result for further processing in the desired use case is selected. Hereby, it is notable that these fusion levels are more theoretical, as many sensors already combine different simpler sensors to yield a better or more specific result. Hence, these sensors already yield results on an intermediate-level and therefore the corresponding fusion is also on intermediate-level.

2.3.4 Fusion Types

Additionally, there are several types of sensor fusion, but in contradiction to the beforehand described places of fusion, the different types are not mutually exclusive for the fusion. Durrant-Whyte et al.[16] distinguishes between 3 different types of how sensor data can be combined for fusion: competitive, complementary and cooperative.

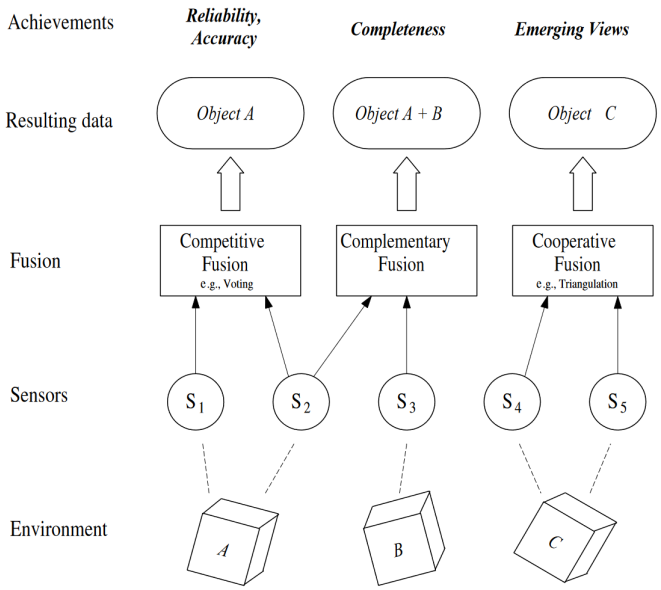

Figure 2.2: Competitive, complementary, and cooperative fusion[5]

These types and their sensor interactions can be seen in Figure 2.2. In a complementary fusion approach, each sensor does not directly depend on each other, but instead can be combined to give a wider look into the environment. The combination of these measurements resolves the incompleteness of the measurement data and produce a more complete image of the property under observation. This type leads to an improved completeness of the resulting data. Sensors S_2 and S_3 in Figure 2.2 get combined using this complementary approach.

Sensors S_1 and S_2 use the competitive approach. Here, each sensor produces independent measurements of the same property, which are redundant[17] but increase the accuracy as well as the robustness in case of measurement errors. This leads to more reliable and accurate result data. Furthermore, it is possible to combine data of different sensors and also different data from the same sensor source[18]. This means only 1 sensor is needed, which can provide multiple measurements, and some previous described advantages of sensor fusion are applicable.

Lastly, the cooperative approach uses the information made available by multiple sensors to originate data, that would not be obtainable from a single sensor. Stereoscopic vision is one example of this type. Here, two-dimensional images from two cameras at slightly different viewpoints get combined into a three-dimensional image of the observed scene. In contrast to the two other types, the cooperative type usually

decreases accuracy and reliability, because the result is sensitive to inaccuracies in all sensor measurements[19], but can produce emerging views on the observed property. Sensors S_4 and S_5 represent the cooperative type.

2.4 Bayes' Theorem and Use in Sensor Fusion

The sensor fusion worked on in this paper utilizes the Bayes' Theorem to combine multiple sensor information into a final result. The Bayes' Theorem describes the probability of an event, based on prior knowledge of conditions that might be related to the event[20].

$$p(X|Y) = \frac{p(Y|X) \cdot p(X)}{p(Y)} \tag{2.1}$$

Therefore, a suitable formula needs to be derived from the original Bayes formula shown in Equation 2.1, which is able to fuse multiple prior conditions into one resulting event, hereby paying attention to complexity and following the computational cost for a robotic environment. But first, some basic formulas need to be established, which can then be further used. To begin with, the conditional probability shown in Equation 2.2 needs to be known to be able to transform the numerator of the Bayes' Theorem.

$$p(X|Y) = \frac{p(X \cap Y)}{p(Y)} \tag{2.2}$$

The conditional probability is defined as the probability of an event occurring, given that another event has already occurred[21]. The joint distributions inside this equation need to be further decomposed by making use of the chain rule of probabilities shown in Equation 2.3.

$$p(X_n \cap ... \cap X_1) = p(X_n|X_{n-1} \cap ... \cap X_1) \cdot p(X_{n-1} \cap ... \cap X_1) \tag{2.3}$$

The chain rule permits the calculation of any member of a joint distribution of a set of random variables using only conditional probabilities[22], which were defined earlier. Additionally, the law of total probability shown in Equation 2.4 is required to transform the equation.

$$p(X) = p(X|Y) \cdot p(Y) + p(X|\neg Y) \cdot p(\neg Y) \tag{2.4}$$

This law relates marginal probabilities to conditional probabilities. Combined with the definition of a complementary probability event shown in Equation 2.5, the transformation with the law of total probability can be further transformed.

$$p(X) + p(\neg X) = 1 \tag{2.5}$$

With the help of these equations, it is possible to derive a suitable formula for the fusion, which only contains known or obtainable and simple inputs. A full derivation is described in Chapter 4.

To apply these equations on continuous sensor measurements, the measurements are taken at concrete time steps. Therefore, they are discretized, and an event here is a sensor data at a certain time step.

2.5 Stochastic Independence

In this work, the use of a Bayesian fusion approach implies the independence of probability values from different sources, so-called stochastic independence. For an approach on this independence, the following equation must be valid.

$$\text{for all independent probabilities applies: } P(\bigcap_{j \in J} A_j) = \prod_{j \in J} P(A_j) \tag{2.6}$$

This should apply to the use case, and it may apply for conditional independence, because 2 sensors are independent if the result is known as shown earlier in Equation 4.11. For using the equation of independence, it is necessary to measure all parts and compare them. But in the case presented here, it is not possible to directly measure the probability of the intersection between the events. Hence, another approach to determine the independence is needed. Here, a two-step test is constructed. First, the input events are tested for linear independence by checking the covariance correlation between them[23]. If the covariance value is 0, then the sources under consideration are linear independent. The higher the covariance gets, the more linear dependent they are. Second, a test for joint or multivariate normal distribution is performed. The measurements of each event get examined whether they differ from such a joint normal distribution. So the null hypothesis, that the samples comes from a joint normal distribution, is tested. This means instead of testing for ordinary normal distribution of each input by utilizing the omnibus test combining skew and kurtosis by D'Agostino et al.[24, 25], the multivariate normal distribution is tested between the inputs with the help of the Henze-Zirkler test[26]. The resulting *pValue*, which is a 2-sided chi squared probability for the hypothesis test, provides information about the rejection or acceptance of the hypothesis. A threshold level α is chosen, typically 0.05, and the *pValue* is compared against it. If the *pValue* lies above the threshold, the null hypothesis can not be rejected and the measurements are likely to come from a joint normal distribution, else the null hypothesis is rejected, and the measurements are not likely to originate from a joint normal distribution. In case both test parts provide positive results for linear independence and multivariate normal distribution origins, the events can be treated as stochastic independent[27]. In a robotic use case, however, a naive Bayes approach assuming the independence between input sources is beneficial[28]. Here, the independence may be violated anyway by mutual localization errors and sensor faults, as well as communication errors. Therefore, the assumption of stochastic independence can be neglected for a robotic environment and must be evaluated for every use case individually.

3. Related Work

In this chapter, first, different sensors are described. Those sensors include the ones used in this work. Second, other approaches on sensor fusion are presented and evaluated in terms of usability.

3.1 Tracking Sensors

The following section will take a look at different sensors and their properties, whose results can be used for sensor fusion. Here, sensors, that are pre-installed on the used robot, as well as other sensors will be compared in terms of accuracy, robustness, measurement frequency and price, previously done for some sensors by Hempel et al.[29]. This leads to a comparison about the feasibility of these sensors in the fusion use case.

3.1.1 Ranging based on Signal Strength

A property of radio waves, which can be used for ranging, is signal strength. As radio waves travel, the signal strength received at a receiving antenna gets smaller due to the signal attenuation rate[30]. So, the further the source of the signal is away, the weaker the signal gets. This property is calculable and can be used to build a ranging based on Received Signal Strength Indication (RSSI)[31].
Bluetooth, WLAN and RFID [32, 33, 34, 35] are technologies, which enable RSSI-based ranging, so already existing technologies can be reused RSSI-transceivers lie within an affordable price range of under 50 dollar [36], are small and hence easy to integrate in applications.
However, there are several problems with RSSI-based ranging. The mentioned signal attenuation rate is the rate α at which the signal strength decreases over distance:

$$SignalStrength \propto Distance^{\alpha}.$$

As a rule of thumb, if $\alpha = 2$ then signal strength drops by 3 decibel every time distance doubles. This sublinear attenuation rate means, that the difference in signal

strength between 1 metre and 2 metres is similar to the difference between 10 metres and 20 metres. A constant level of noise can therefore result in an ever-increasing error, when signal strength is used to estimate distance. As shown in Figure 3.1, changes in signal strength due to distance become small relative to noise, even if the level of noise remains the same over distance [37].

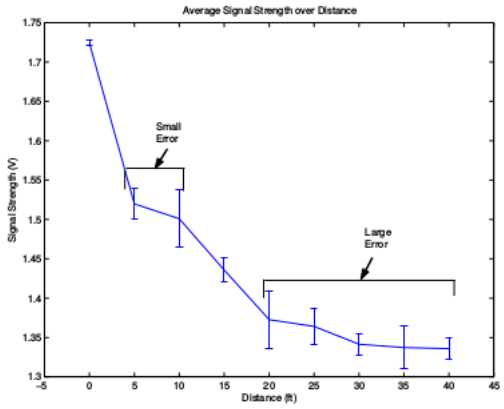

Figure 3.1: RSSI - Error increase over distance [37]

Additionally, there are severe multipath effects by radio wave reflections from walls and obstacles during indoor use. Furthermore, through the build characteristics of transceivers, the antennas never propagate radio waves with equal signal strength in every direction from the source [38].
The transceiver will be mounted on robots, which can block radio signals depending on the used materials[39]. Based on the orientation of the robot and the transceiver, the signal strength varies a lot.
The above-mentioned characteristics lead to a small usable range with bad accuracy and robustness.

3.1.2 Ranging based on Phase Differences

Another property of radio signals, that can be utilized for ranging, is phase shift. By sending out radio signals of different frequencies, the reflected signals from the target can be compared with the sent ones. Between these signals, phase differences can be observed. A resulting distance can be computed out of these phase differences [40].

The property of phase shift can be used to get high accurate result values, as well as reasonable ranging distances. An implementation of ranging with phase differences by Qiu [41] yields values shown in Figure 3.2.

With this implementation, a mean error of 10 to 20 centimetres can be achieved. The range measurements were repeated several times for different tag angles. Here, range measurements show a tag angle insensitivity [41].
This is especially useful for use with mobile robots.

By using radio signals to measure the phase shift, a high update frequency is given. Also, small and affordable radio frequency transceivers [36] can be used, which enable an easy integration.

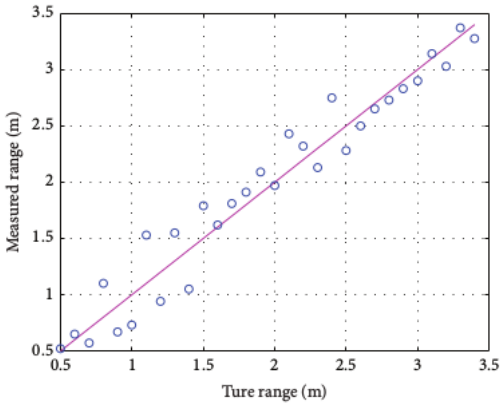

Figure 3.2: Yielded values with phase difference ranging [41]

However, ranging inaccuracy can be caused by multipath effects, frequency interferences in the environment or variations over temperature [41].

3.1.3 Ultra Wideband Time of Flight

A distance can be measured by calculating the time of flight of radio signals between a sender and a receiver. Because bandwidths over 500 MHz are used to communicate in, it is called ultra-wideband.
Here, time stamped messages are used to communicate between sender and receiver. With these time stamps a time of flight, hence an adequate distance, can be computed by multiplication with the speed of light as shown in Equation 3.1. Here, d is the resulting distance, T_{prop} is the time of flight and c_{air} is the speed of light in the medium air. The speed of light in air equals 299700 kilometres per second [42].

$$d = c_{air} \cdot T_{prop} \tag{3.1}$$

Michler et al. [43] states, that in reflectionless environments, as well as in typical multipath environments, an error deviation better than three centimetres and a mean error better than three millimetres can be reached.
In free space scenarios, ranges of up to 280 meters could be achieved while still keeping the system's accuracy at a high level.
Influential here is the use of ultra-wideband, because a larger bandwidth will increase the robustness against frequency interferers and multipath propagation [43].
Experiments by Cardinali et al. [44] support Michler's statements as their results showed, that the presence of multipath effects strongly affects ranging accuracy but favours a larger bandwidth [44].

By using radio signals to measure the phase shift, a high update frequency can be achieved, which enables a steady and continuous measurement input. The DWM1000 modules are small and affordable radio frequency transceivers [45], which enable an easy integration.

3.1.4 Optical TOF LIDAR Sensor

The time of flight light detection and ranging technology utilizes light to yield ranges to objects and is used for mapping and obstacle avoidance in many of today's

applications. It is a laser scanner capable of sensing 360 degrees around itself by having a laser mounted on a rotating base. By measuring the time T_{round} between the sending of the laser beam and the arrival of the reflected light of an obstacle, a distance d can be computed as shown in Equation 3.2.

$$d = c_{air} \cdot \frac{T_{round}}{2} \tag{3.2}$$

The used speed of light in air c_{air} equals 299700 kilometres per second [42]. Therefore, the update frequency can be high, which supports stable and continuous measurement results even of fast moving objects and is only limited by the rotation speed of the base.
Raj et al.[46] states, that the time of flight LIDAR is physically limited in range to about 100 metres, because the detector is not able to sense the reflected light above this range with a typical sensitivity. But according to the application and the object size, a much shorter range suffices for object detection. Nowadays, especially with lower detection range, the size, weight, and the price are decreasing to suitable units for use within a mobile robotic application[46]. LIDAR specifications[47] of the HLS-LFCD2 show highly accurate distance values yielded by those optical systems, with deviations of up to 5% on accuracy and respectively 3.5% on precision on maximum range.
Drawbacks while using an optical LIDAR system, is the former mentioned limited range and the decrease in accuracy with increasing range. Further, a necessary unobstructed line of sight on the object is required to range to it. Though, with typical light sensitivity, the brightness of the environment should not be a limiting factor and the ranging is not susceptible to other noise.

3.1.5 Sonar Sensors

Other ranging approaches, like Cricket and Active Bat [48], use sound to calculate a distance. By measuring the time between the sent sound signal and the related reflection of that signal (echo) a distance can be computed with Equation 3.3, where v_{sound} is the velocity of sound and T_{round} is the measured round trip time [49].

$$Distance = v_{sound} \cdot \frac{T_{round}}{2} \tag{3.3}$$

Sound signals propagate much slower than radio signals or light. This enables an easier measurement of the time of flight, but also limits the update frequency of distance values [50]. Especially for ranging on longer distances or when multiple robots want to range to each other, the slower propagation time sets an upper limit to the performance. To calculate distances between a swarm of 5 robots, the upper boundary of the update frequency is 5 Hz [51]. So the performance on fast moving robots is drastically limited.

The ranging accuracy in this approach is prone to decrease due to changes in temperature, air humidity and atmospheric pressure, because they interfere with the physical properties of sound propagation.

Ultrasonic range finders are purchasable at an affordable price of under 50$ [52]. They are small and because of that easy to integrate in a variety of applications.

Ultrasonic ranging is typically used for obstacle detection, instead of ranging, in submarines as well as in automotive applications or to control liquid levels in industrial tanks [53].

3.1.6 Comparison

As also stated by Hempel et al.[29] ultra-wideband time of flight achieves the best results in overall comparison, closely followed by optical LIDAR ranging.
A ranging based on signal strength will yield less useful results in terms of accuracy and robustness, and is not a feasible approach for the use with mobile robots.
Furthermore, because of the characteristics, the use of ranging based on phase differences is not feasible in an area with many obstacles or with a crowded frequency band. If that is not the case, this approach is only slightly inferior to ultra-wideband time of flight in terms of accuracy and maximal distance. Therefore, it is not universally suited for use in any environment, but it can be used in some cases.
Sonar can be used for single distance measurements, but for using it with multiple robots the update frequency limitations, due to a slower propagation velocity, are too severe for this approach to be practicable. Hence, in a teammate tracking situation, a bare minimum of 2 robots will be active and moving at a time and so the ultrasonic approach is not suited.
Ultra-wideband time of flight ranging is an accurate and robust approach, especially for high distances and obstacle rich environments. Further, optical LIDAR ranging with a 360 degree field of view yields even more accurate results in line of sight scenarios and is less susceptible to environmental noise and so the usage in areas with high external influences is possible. Both ultra-wideband time of flight and LIDAR ranging use the fastest propagation speeds available and are therefore suited for the use with mobile objects. Thus, it is most feasible for teammate tracking in a dynamic environment to combine the accuracy of LIDAR in line of sight scenarios and ultra-wideband time of flight in non line of sight scenarios.
Therefore, the DWM1000 ultra-wideband ranging modules, like the ones used by Michler et al.[43], were integrated alongside LIDAR HLS-LFCD2 modules[47] for a combining sensor fusion.
Additionally, each used DWM1000 module can differentiate between the modules it ranges to by sending a unique ID. Hence, it is possible to identify the range value and map it to the related teammate, which specifically enables our used approach to yield sensor fusion results for each tracked teammate.

This fusion approach using the above described sensors is further analysed over the course of this thesis.

3.2 Fusion Approaches

Following, after a sensor comparison was given, some state-of-the-art fusion approaches get described and advantages as well as disadvantages of these approaches get disclosed. Afterwards, these approaches get set in relation to the used method for this thesis.

3.2.1 Kalman Filtering

A common and widely used approach for the fusion of data is Kalman filtering developed by Kalman and Bucy[54][55]. It depends on the Markov assumption of state of random variables in a stochastic process, stating that the current state is only dependent on the past state. Therefore, past and future states are not correlated if the current state is known[56]. It uses a mathematical model for filtering signals using measurements with respectable amount of statistical and systematic errors, except for unknown systematic ones, which can not generally be handled. Here, the data from successive time intervals get fused, providing a maximum likelihood estimate of the parameters. Possible multisensory scenarios can be mapped to an internal state vector, alongside variables needed to suffice the Markov assumption[56], as long as there are only linear dependencies between inputs and system states[57]. The filter uses a discrete-time algorithm to remove noise from sensor signals in order to produce fused data. The basic Kalman Filter can be described by the following equations. Equation 3.4 describes the dynamic of the system.

$$x_k = F \cdot x_{k-1} + B \cdot u_k + w \tag{3.4}$$

Hereby, x_{k-1} represents the system state at time k-1, F is the state transition matrix, u_k is the input to the system at time k, B relates the input to the system state at time k x_k and w represents the system noise, modelled as white noise $\sim N(0,Q)$ with the covariance matrix as Q. Equation 3.5 describes the noisy observations of the system.

$$z_k = H \cdot x_k + v \tag{3.5}$$

With z_k being the sensor observations at time k, the matrix H relates the measurements to the internal state, and v is the measurement noise, also modelled as white noise $\sim N(0,R)$ with the covariance matrix R.
Then, the Kalman Filter predicts a state estimate utilizing Equation 3.6.

$$\hat{x}_{k|k-1} = F \cdot x_{k-1|k-1} + B \cdot u_k \tag{3.6}$$

Further, the innovation y_k given by the observations at time k and the prediction are calculated as shown in Equation 3.7.

$$y_k = z_k + H \cdot \hat{x}_{k|k-1} \tag{3.7}$$

Afterwards, by additionally incorporating the prediction of the estimate covariance $\hat{P}_{k|k-1}$ and the innovation of the covariance S_k, the optimal Kalman gain K_k can be calculated with Equation 3.8.

$$K_k = \hat{P}_{k|k-1} \cdot H^T \cdot S^{-1} \tag{3.8}$$

Finally, the updated state estimate can be calculated with Equation 3.9.

$$x_{k|k} = \hat{x}_{k|k-1} + K_k \cdot y_k \tag{3.9}$$

In cases where the model matches the real system, only white noise exists and the covariances of the noise are known exactly, the linear Kalman filer is proven to be optimal[58].

For initialization at start-up, a state estimate x_0 is needed as well as an estimate of the covariance matrices. The covariance estimate can be inaccurate, due to subsequent application of the filter, that will let the covariance estimate approach its exact value. Each iteration of the Kalman filter takes the same computational effort, therefore it is well-suited for real-time applications.

Besides the wide usage of the Kalman filter, disadvantages can still be found and need to be considered for every application individually[5]. For applications other than ones with linear systems, variants of the Kalman filter need to be used in order to achieve satisfying results. For these non-linear applications, the Extended Kalman Filter(EKF) needs to be used, which utilizes non-linear stochastic difference equations for the system model[59]. Other applications with unknown or volatile statistical parameters over time need variants of the Kalman filter, that also have to estimate the statistical parameters. Further disadvantages can be found in the used evaluation criteria. The Kalman filter approach minimizes the error using a least mean square approach, but depending on the application, other criteria would perform better. So for every application, suited criteria need to be found and used. Additionally, regarding the implementation in embedded systems, the computational effort can be disadvantageous. Kalman filtering requires matrix multiplication and inversion, which can lead to a high computational cost. Therefore, dedicated vector processing units or, if not available, efficient implementations of matrix operations are needed to achieve adequate performance in embedded systems. Furthermore, as Thrun et al.[56] stated, the combination of multimodal sensors is hard to achieve and can render the result useless. And moreover, spatial relations and especially shifts in it, are hard to model inside the state vector and therefore hard to achieve. In addition to this, an unknown starting state with a uniform distribution of no knowledge is hard to model. Therefore, another fusion approach was introduced with this thesis.

Though, the approach is used in many scenarios to combine multiple unimodal sensors and inputs. The Extended Kalman Filter in a distributed form was used by Karol et al.[60]. Here, multiple robots with multiple sensors track an object from multiple points of view. In this scenario, a mean translation error from the real position of the tracked object of 0.05 metres with a standard deviation of 0.01 metres was yielded.
Corrales et al.[61] utilized the sensor fusion with Kalman to fuse inertial motion capture data and ultra-wideband localization data. A human was successfully localized in a robotic workspace environment with an overall translation error of 0.14 metres.

3.2.2 Reliable Abstract Sensors

The idea of fusion using reliable abstract sensors was first introduced by Marzullo[62]. This approach incorporates fault tolerance in multi sensor environments by utilizing a geometric derivation for fault masking. Here, a sensor is seen as a piecewise continuous function with parameters shape and accuracy range. Shape is defined by the form of uncertainty around the sensor measurement value and accuracy range

is the interval containing the true value of a measurement entity. By combining those sensors, an improved abstract sensor can be created. The accuracy range of this abstract sensor is hereby derived by finding intersections of ranges of the single sensors. Further, with this approach, fault-tolerant sensor averaging is introduced, which results in an abstract sensor interval that contains the true value for sure, even if t sensors are faulty. Here, an arbitrary number of t out of a maximum of $2t+1$ sensors can be faulty, for a fault-tolerant sensor averaging algorithm to produce reliable results. In the original version by Marzullo, this sensor averaging is applicable in a linear one dimensional case. Chew[63] extends this to work with sensors of arbitrary dimensions. In a multidimensional case, $2t \cdot D+1$ sensors are required to still tolerate t faulty sensor, where D is the number of used dimensions.
In contrast to this, our approach only partially investigates reliability. Therefore, it is not used in this work.

3.2.3 Deep Reinforcement Learning

Another fusion approach can be achieved by Deep Reinforcement Learning. A deep neural network is able to fuse data by learning a shared representation between those inputs[64]. In case of unknown optimal feature representation, it is also possible, that this deep neutral network models a function approximation of Q-Learning[65], which then can be used to derive a policy. Here, the action-value function is $Q(s,a) = \mathbb{E}_{\pi}[\sum_{k=t}^{T} \gamma^{k-t} r_k | s_t = s, a_t = a]$ with Q as the expected discounted return of action a in state s after executing policy π including the discount factor γ and reward r. The policy $\pi(a,s)$ determines which action to take in a certain state. The optimal policy $Q^*(s,a) = max_{\pi} Q(s,a)$ is found by following the optimal Q function. Hereby, this function satisfies the Bellman optimality equation $Q^*(s,a) = \mathbb{E}_{r,s'}[r + \gamma \max_{a'} Q^*(s,a)]$ and is typically found by iteratively taking the action with the highest Q-value for each state $\pi(a,s) = argmax_a Q(s,a)$. The approximation of this Q-value with deep neural networks creates Deep Q-Networks(DQN), which take the different inputs for the fusion as the current state s.

This fusion approach is a high level end-to-end approach, meaning it creates commands regarding which action to take by using a derived policy without paying attention to the numerical values of the fusion of the inputs. It requires a lot of a priori training, to find the optimal policy and is not suited for instant use cases and therefore not well suited for our use case.
Additionally, the actions in this case consist of different commands instead of weights on the sensor inputs. Therefore, the following results are not perfectly suited to be compared against our work, but can still set a reference. In a real world scenario, the approach can deliver robust results, combining multiple unimodal sensors, with a position estimation accuracy in the interval $[10^{-2}, 10^{0}]$ meters depending on the fusion architecture used[65]. This is shown in Figure 3.3. Here, all fusion approaches have a steep increase in cumulative distribution and start to settle at a distance between 0.1 and 0.2. The cumulative distribution at the point of settlement ranges from approximately 0.6 to 0.9. This means 60 to 90% of the results have a position estimation error of 0.1 to 0.2 metres, depending on the used fusion architecture.

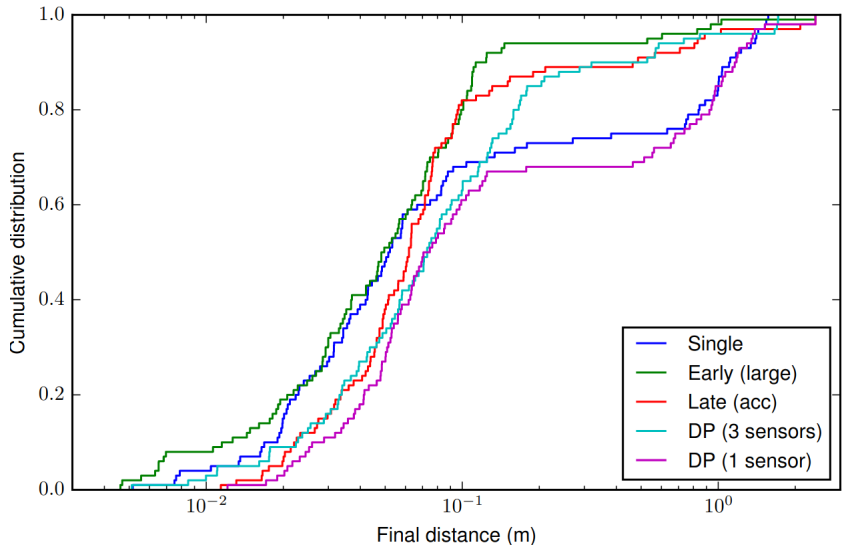

Figure 3.3: Cumulative distribution of the final position estimation error[65]

3.2.4 Multi-Sensor Occupancy Grid

Another fusion approach, that can be utilized, is through multi-sensor occupancy grids. Occupancy Grids are a form of certainty grids, which only track occupation of a cell or not. It can be used especially when multiple sensors, which track different types of obstacles, can not be combined using Bayesian Filtering methods like Kalman, because it will result in conflicting information[56]. Instead, each sensor k gets a preprocessed into its own occupancy grid m_k and the grids get combined afterwards. This can be done with De Morgan's law for independent measurements, shown in Equation 3.10.

$$p(m) = 1 - \prod_k (1 - p(m_k)) \tag{3.10}$$

Or the maximum can be computed like in Equation 3.11.

$$p(m) = \max_k (p(m_k)) \tag{3.11}$$

The latter yields the most pessimistic estimate of the inputs. Here, if any of the input grids show an occupation, the combined grid will also show it.

This fusion approach is partly similar to the one we use in this thesis, but more inaccurate by using only occupation and not certainty in each cell. So, no gradient of certainty of the obstacle location can be tracked. Additionally, each cell in this grid is a single hypothesis about the obstacle and therefore less information about a single teammate is given Hence, both of these disadvantages lead to less knowledge about the overall teammate location.

4. Bayes-Hempel Method

This thesis covers the tracking and localization of teammates in a robotic swarm environment2.1. In a swarm of robots to avoid collisions and prepare movement planning, an individual robot tries to keep track of teammate positions using the likelihood of their residence. The tracking is done using different sensors, which are built onto each robot. Each sensor source gathers encoding-independent data, which is stored in an encoding-dependent discrete grid of the environment, as described in 2.2. The stored data of each sensor is then combined utilizing a sensor fusion approach using the Bayesian theorem as described in 2.4. Using these grids, data can be evaluated by comparing relative differences in likelihood and an estimate of absolute differences in meters of real positions of teammates and fused sensor information.

4.1 Solution Design - Bayes-Hempel Method

In this section, the design of the Bayes-Hempel Method is described. The method takes measurement data as the input as described in the following section. Afterwards, the processing and storage in modified certainty grids is laid out. The fusion of those modified certainty grids derived from the Bayes theorem is presented in the last part of this section.

4.1.1 Data Flow

To tackle the problem described in the section above, a solution was designed. The underlying data flow of the created solution used during this thesis can be seen in Figure 4.1. Each row in this figure represents a time step during the tracking and consists of several components.

The static map of the underlying environment is used as one component, which can be utilized to distinguish static walls and obstacles from possible teammates. This is done by comparing the other input sources with the static map and subtracting

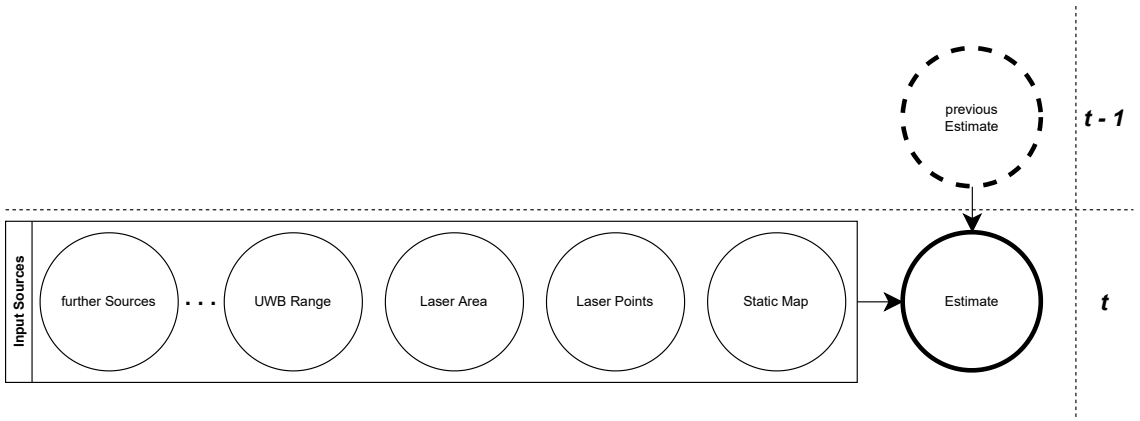

Figure 4.1: DataFlow - from source to estimate

those static walls and obstacles from the measurements.

Another component, if one result is already calculated, is the estimated result of the previous time step. This old result is used to insert a time component into the current estimate. But before it can be used, highlights of the old result with high likelihood of residence get emphasized to be more susceptible to further modifications. This means possible the likelihood of possible residence points gets increased. Then, a Gaussian filter is applied to the grid, which catches onto those highlights and expands them in every possible movement direction of the teammate. Hereby, a standard deviation according to the maximum movement speed of the robots is chosen in a way that incorporates most movements, but still provides a lot of information about the teammate. Hence, the blur, which reflects the uncertainty, caused by the Gaussian filter is correlated to the movement of the robot in each time step[56]. Therefore, the old estimate compensates for the movement of the teammates and increases the accuracy even if the robot lost track of its teammate.

Lastly, an arbitrary number of sensor sources n are the additional components needed to mainly track the teammates. Here, the range measurements of an ultra-wideband sensor and the assumptions created by a LIDAR sensor are used, but further ones can be added at will. They provide continuous information input and get sampled at discrete time steps. Therefore, depending on the sensor, the sampled data of each sensor may not represent the exact same time step as the others. But it is ensured, that the latest input data is used in each sample interval. Hence, the sample interval time is chosen according to the utilized sensors.

Like Thrun et al.[56] stated and described in Section 3.2 for occupancy grids, separate preprocessing of inputs and subsequent combination is a popular approach. The gathered data from the components get processed into modified certainty grids and is then combined using a Bayesian fusion approach. Both are further described in the following sections.

4.1.2 MCG - Modified Certainty Grid

To further process and combine all component data into a result, they have to be available as certainty grids as described in Section 2.2, but with additional improvements made in this work. The transformation of every component's gathered data into a modified certainty grid is performed as follows. Every input measurement has a corresponding assigned cell in the grid. Therefore, each measurement, recorded in the metric space M, must be transferred into the grid space G by using the coordinate of the origin of the underlying grid (X_O, Y_O) and the corresponding grid resolution R. With this information, a formula can be established to transform a metric coordinate (X_M, Y_M) into a grid coordinate (X_G, Y_G). This is shown in Equation 4.1.

$$X_G = (X_M - X_O)/R \text{ and } Y_G = (Y_M - Y_O)/R \tag{4.1}$$

Afterwards, the resulting coordinate has to be subject to discretization, so that every grid space coordinate has exactly one corresponding grid cell and the input measurement can be mapped onto the grid. After that, every cell of the grid, that is not affected by those measurement inputs, gets an unknown probability assigned. This probability reflects the uncertainty whether a teammate is present in this part of the map, considering lack of knowledge on this part or sensor inaccuracy.

Now, contrary to the ordinary certainty grid, in this thesis, each grid refers to a single tracked object. That means, instead of each cell being seen as a hypothesis about whether the teammate is there, the whole grid represents the hypothesis about the teammate's residence. This is achieved by dividing the assigned value of each grid cell C_{ij} with the sum of all grid cell values. It is shown in Equation 4.2.

$$C_{ij} = \frac{C_{ij}}{\sum_{i=0,j=0}^{I,J} C_{ij}} \tag{4.2}$$

Hence, all grid cells can be again summed up to 1, instead of each cell being in the probability range of [0,1]. Here, it is possible to obtain small to tiny numbers for every grid cell, depending on the number of grid cells in the grid. To not lose precision, values of data type double are used, which are able to store up to 15 digits as they are. Additionally, a conversion constant is applied to decrease the number of inexpressive decimal places.

These modifications to the certainty grid may increase the payload to process and transmit for each robot depending on the number of teammates, but also lead to more distinct knowledge about every single teammate, which is a goal to achieve in teammate tracking. Additionally, due to this method, every sensor output is transformed to a grid and is combinable independent of its modality.

4.1.3 Bayes Fusion

This approach represents a cooperative fusion because every input observes the location of the tracked teammate, but with different observation techniques. Additionally, concepts of a competitive fusion can be found here as well by observing

the same modality with different sensors. Therefore, the Bayesian fusion of the component inputs yields a result with increased accuracy, reliability, robustness as well as emerging views on the tracked teammate. Although, the fusion will be used in a robot use case, and therefore in an embedded environment, instead of a low level fusion, an intermediate level fusion is used, because the sensors utilized in this case are already on an intermediate level themselves by combining different raw sensors to yield only a range measurement. This means, instead of combining raw sensor data, data already processed to ranges gets combined and translated into probabilities in a 2D grid. Hereby, none of the information yielded by the sensors is or neglected. Hence, the Bayesian fusion approach is a fusion on intermediate-level. This may increase the complexity and computation time, but ensures that sensors of many kinds can be fused and therefore increases the expandability.
Based on the work of Moravec[11], but fully derived and adjusted to this work's use case, the goal is to calculate the probability of residence C_{ij} of a cell in the fused result, given two measurements of sensors A and B.

$$p(C_{ij}|A_{ij} \cap B_{ij}) \tag{4.3}$$

This is shown in probability 4.3. From the individual sensor measurements, only the probabilities shown in formula 4.4 are known.

$$p(C_{ij}|A_{ij}), p(C_{ij}|B_{ij}) \tag{4.4}$$

Now, the unknown probabilities need to be eliminated to be able to calculate the desired result, which is done with the help of the in Section 2.1 established formulas. By using the Bayes' theorem 2.1 on our goal probability 4.3, the Equation 4.5 can be derived.

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{p(C_{ij}) \cdot p(A_{ij} \cap B_{ij}|C_{ij})}{p(A_{ij} \cap B_{ij})} \tag{4.5}$$

Here, the law of total probability 2.4 can be used on Equation 4.5, which results in Equation 4.6.

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{p(C_{ij}) \cdot p(A_{ij} \cap B_{ij}|C_{ij})}{p(C_{ij}) \cdot p(A_{ij} \cap B_{ij}|C_{ij}) + p(\neg C_{ij}) \cdot p(A_{ij} \cap B_{ij}|\neg C_{ij})} \tag{4.6}$$

To further transform the preceding Equation 4.6, parts can be rewritten using the rules for conditional probabilities 2.2. This results in Equation 4.7.

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{p(C_{ij}) \cdot \frac{p(C_{ij} \cap A_{ij} \cap B_{ij})}{p(C_{ij})}}{p(C_{ij}) \cdot \frac{p(C_{ij} \cap A_{ij} \cap B_{ij})}{p(C_{ij})} + p(\neg C_{ij}) \cdot \frac{p(\neg C_{ij} \cap A_{ij} \cap B_{ij})}{p(\neg C_{ij})}} \tag{4.7}$$

Now, this equation can be simplified into Equation 4.8.

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{p(C_{ij} \cap A_{ij} \cap B_{ij})}{p(C_{ij} \cap A_{ij} \cap B_{ij}) + p(\neg C_{ij} \cap A_{ij} \cap B_{ij})} \tag{4.8}$$

Then, if the chain rule of probabilities is applied to the beforehand simplified equation, the combined probabilities can be resolved. This results in Equation 4.9.

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{p(B_{ij}|C_{ij} \cap A_{ij}) \cdot p(C_{ij}|A_{ij}) \cdot p(A_{ij})}{chainRule \cdot p(A_{ij})} \tag{4.9}$$

Hereby, the *chainRule* is equal to the following equation.

$$chainRule = p(B_{ij}|C_{ij} \cap A_{ij}) \cdot p(C_{ij}|A_{ij}) \cdot p(A_{ij}) + p(B_{ij}|\neg C_{ij} \cap A_{ij}) \cdot p(\neg C_{ij}|A_{ij})$$

Fortunately, the combined equation can be simplified again into Equation 4.10.

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{p(B_{ij}|C_{ij} \cap A_{ij}) \cdot p(C_{ij}|A_{ij})}{p(B_{ij}|C_{ij} \cap A_{ij}) \cdot p(C_{ij}|A_{ij}) + p(B_{ij}|\neg C_{ij} \cap A_{ij}) \cdot p(\neg C_{ij}|A_{ij})} \quad (4.10)$$

Now, only the probabilities containing B_{ij} are still unknown. Here, like Moravec[11] did, the conditional independence between probabilities A_{ij} and B_{ij} is assumed, which leads to the relation shown in 4.11 between these probabilities.

$$p(B_{ij}|C_{ij} \cap A_{ij}) = p(B_{ij}|C_{ij}), \text{ with } A_{ij} \text{ and } B_{ij} \text{ conditional independent} \quad (4.11)$$

By using this relation on Equation 4.10, the unknown parts can be simplified into Equation 4.12.

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{p(B_{ij}|C_{ij}) \cdot p(C_{ij}|A_{ij})}{p(B_{ij}|C_{ij}) \cdot p(C_{ij}|A_{ij}) + p(B_{ij}|\neg C_{ij}) \cdot p(\neg C_{ij}|A_{ij})} \quad (4.12)$$

The Bayes' Theorem can then be used on the still unknown probabilities of this equation. This results in Equation 4.13, which afterwards can again be simplified into Equation 4.14.

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{\frac{p(C_{ij}|B_{ij}) \cdot p(C_{ij})}{p(B_{ij})} \cdot p(C_{ij}|A_{ij})}{\frac{p(C_{ij}|B_{ij}) \cdot p(C_{ij})}{p(B_{ij})} \cdot p(C_{ij}|A_{ij}) + \frac{p(\neg C_{ij}|B_{ij}) \cdot p(\neg C_{ij})}{p(B_{ij})} \cdot p(\neg C_{ij}|A_{ij})} \quad (4.13)$$

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{ResidenceProbs}{ResidenceProbs + NoResidenceProbs} \quad (4.14)$$

Hereby, *ResidenceProbs* are the events related to the likelihood of residence and *NoResidenceProbs* are the complementary events to this. Both are shown in the following.

$$ResidenceProbs = p(C_{ij}|B_{ij}) \cdot p(C_{ij}) \cdot p(C_{ij}|A_{ij})$$
$$NoResidenceProbs = p(\neg C_{ij}|B_{ij}) \cdot p(\neg C_{ij}) \cdot p(\neg C_{ij}|A_{ij})$$

This resulting form contains the known parts shown in 4.4 and additionally the probability $p(C_{ij})$, which is the a priori assumption, that something is detected in the area. So to speak, it is the overall probability, which indicates the likelihood to encounter an object or other robot in the environment and is dependent on the size of the environment as well as the swarm size. Further, as visible in Equation 4.14, it can be simplified again to reduce the complexity. This results in the final Equation 4.15, which can be used to fuse sensor information with Bayes' reasoning.

$$p(C_{ij}|A_{ij} \cap B_{ij}) = \frac{1}{1 + \frac{p(\neg C_{ij}|B_{ij}) \cdot p(\neg C_{ij}) \cdot p(\neg C_{ij}|A_{ij})}{p(C_{ij}|B_{ij}) \cdot p(C_{ij}) \cdot p(C_{ij}|A_{ij})}} \quad (4.15)$$

4.2 Implementation

In this section, the architecture as well as the used hard- and software is explained. From those design decisions, the transformation from used sensor data to grids and the simulation of errors are derived.

4.2.1 Languages, Software, Frameworks

The designed solution uses Turtlebot3 robots of type burger, which are designed and developed by ROBOTIS Co.,Ltd.[66]. Every used robot utilizes the Robot Operating System(ROS)[67]. It is an open-source software development kit for robotics operating cross-platform. Here, in detail, ROS 2[68] in version Foxy Fitzroy in combination with python3[69] was used to develop the software used in this thesis. For easy maintenance and deployment, the experiments in this thesis got conducted with virtualization using the Gazebo Simulation Framework[70]. The results from those experiments get collected using the rosbag package[71] from the ROS framework. And lastly, get analysed using python libraries pandas and NumPy.

4.2.2 Used Sensor Sources

LIDAR

Every robot used for this thesis has a factory equipped HLS-LFCD2 LIDAR scanning system from Hitachi-LG Data Storage, which scans 360 degrees around the robot with a rotation frequency of 5 hertz. This way, each input from this sensor delivers 360 ranges around the robot, one for each degree. Its specifications are well documented[47] and are used to transfer measurements in the grid space with related precision and errors. As this sensor is factory equipped on the robots, the specifications fit the usage with those robots very well. By using laser to detect the environment, this sensor can be classified as an optical sensor. According to Section 3.1 this type of sensor fits our use case in an excellent way, by having a high accuracy as well as a fast propagation time to track moving robots. Additionally, multiple input sources can be derived from information gained by this 360 degree optical tracking sensor. First the obvious assumption, when the sensor returns a certain range. At this range, an object is located, and the grid related cell gets a high probability assigned. This assumption will be used to create the first input source. This input created with by points from the laser range measurement is shown in Figure 4.2 Here, the used environment is overlaid on the yielded laser points to show the correlation. The tracking robot is shown with a little model of the robot. Notable while working with the LIDAR sensor to pinpoint a location is that the detected ranges do not range towards the centre of the object, but to the edge of it. Therefore, the LIDAR sensor detects the smallest range to it, regardless of the object size.
The second input source relies on the assumption of non-transparent objects and robots, that are given in our use case. Hence, it can be assumed that the area the laser traverses until finding an object is empty because otherwise the laser would be intercepted by other objects in its path. This is shown in Figure 4.3. Here, the used environment is also overlaid for clarification of relations and the tracking robot is shown with a little model of the robot. This means the second input is a polygon area of low probability grid cells around the robot with the coordinates of detected objects as vertexes, called the laser area.

Range UWB

The comparison in Section 3.1 reinforces the use of Ultra Wideband Time of Flight(UWB TOF) sensors, which are already available in the swarm laboratory

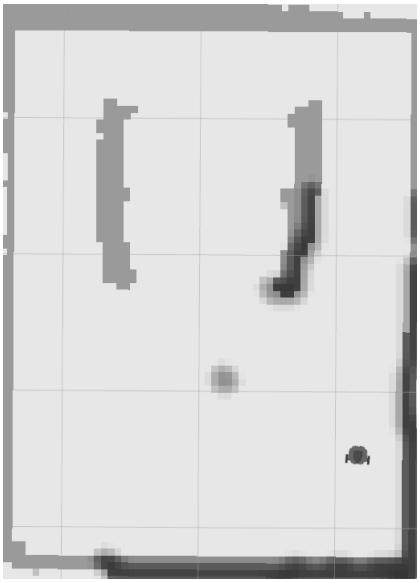

Figure 4.2: A certainty grid yielded by the laser points input

SwarmLab. Here, the DWM1000 modules by DecaWave Ltd.[72] shown in Figure 4.4 are used. Based on its high propagation speed and accuracy, it is well suited for range tracking of teammates. Furthermore, it is small hence easy to attach and costs an affordable amount of money. Additionally, the specifications, limitations and influences of disturbances of the DWM1000 modules is well documented[73] and can further be used to translate the range measurements into the grid space and to help in evaluating errors. Here, it has to be kept in mind, that each sensor module ranges to another module and therefore the yielded range is determined by the position of these modules on the robots.
The tracking robot and every teammate is equipped with one of those UWB sensors, utilizing a unique robot ID for each. Therefore, a distinguishable range tracking between every robot can be done, which results in a set of ranges from the tracking robot to each teammate. By utilizing the position of the tracking robot, the range to each teammate is then converted into the grid space in the form of a circle. The specific range functions as the radius around the robot. This is shown in Figure 4.5. Here, like the inputs above, the used environment is overlaid, and the robot is shown with a little model. So, the resulting input source is a set of grids with high cell probabilities on a circle around the tracking robot. According to this, the cardinality of this set is equal to the number of tracked teammates.

4.2.3 Architecture

The designed code architecture, that is deployed on each tracking robot, relies on the principle of independent, interchangeable and expandable modules like Shafer et al.[74] proposed. It is shown in Figure 4.6. Each sensor and input source, as well as the source combiner, is independently encapsulated and stores all data needed for its functionality in its respective module. Also, the method of combining the inputs can be exchanged easily by adding a new function in the combiner module containing the combine logic and using the data collected in this module as input. This leads to an easy to maintain code base, on which further development and expansion can be

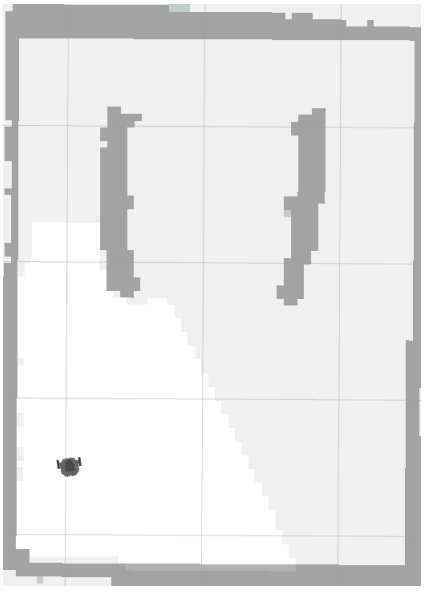

Figure 4.3: A certainty grid yielded by the laser area input

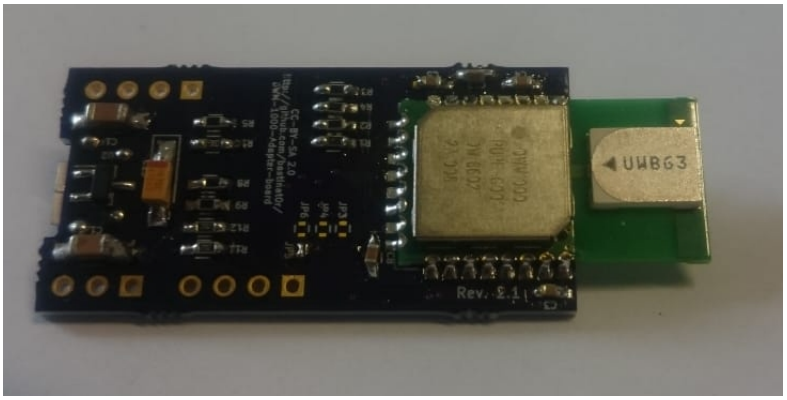

Figure 4.4: DWM1000 module used in this thesis

done.
The only interconnection between modules is established via ROS topics. Every module communicates by sending common Occupancy Grid ROS messages[75] over these topics. Therefore, the only static reference between every module are the corresponding ROS topic names. Hence, a new module can be added by solely defining those new ROS topic names. With increasing swarm size, new challenges arise in terms of transmitted size of data. More teammates lead to the sending of more grids, because all robots need a grid for each teammate. If the size of data exceeds a certain amount, the data has to be compressed or else some data can get lost. Further, the transmission latency between each module can increase to a point, at which the correlation between information input and processed result is lost, because of a too severe time delay. Therefore, this needs to be considered for big swarm sizes.

4.2.4 Simulating Errors

The experiments during this thesis are conducted within a simulation. Therefore, independent faults on the inputs need to be simulated as well. Here, the simulated

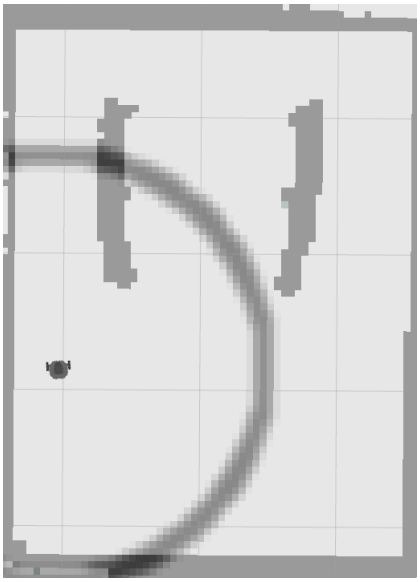

Figure 4.5: A certainty grid yielded by the range input

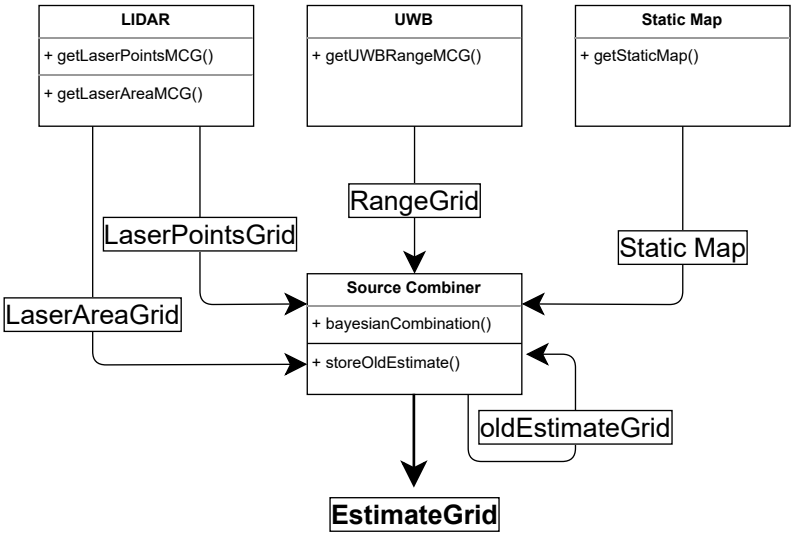

Figure 4.6: Modular architecture used in this thesis

robotic system is assumed to be an in a settled state, so the system is in the second part of the bathtub curve with a constant failure rate known as random failures[76]. The associated cumulative distribution function can be described with the help of a Weibull distribution[77]. Therefore, a simulation of faults like Weikert et al.[78] was applied. Each input is assigned a specific time based on the 2-parameter version of this Weibull distribution, which couples the fault rate to the runtime. Then, if a fault occurs, a second Weibull distribution, independent of the runtime, is used to determine the duration of the fault. Here, the parameters are chosen in a way, that faults on an input occur multiple times during one run and the mean duration of this fault is at least one time step of the system. Hence, significant fault scenarios can occur with one or more sensors having a fault during the run.
Measurement faults can be divided into two categories: random and systematic. Systematic faults are caused by the sensors themselves, and random faults are

mainly caused by interference with the environment. Zug et al. [79] distinguished systematic faults mainly between constant, continuous and non-continuous faults. Here, constant and continuous faults usually can be detected and corrected and are therefore not important to investigate. Additionally, the simulation does not include error modelling on the sensor inputs and the measurements should have no significant systematic faults. Random faults are of more interest here. They are classified as varying non-continuous faults and are usually caused by the physical properties of the measurement process and get influenced through varying environmental conditions[79]. In a real scenario, this can happen and hence they need to be analysed. In this work 3 different types of random faults and their influences were investigated: complete failure of inputs, value-correlated fault and non value-correlated fault. The first is modelled through a neglect of an input source. The value-correlated fault is modelled by applying a distortion on the yielded values and the non value-correlated one overwrites measurement values at random to the yielded result grid. This should cover most types of faults that can occur in a real scenario. Thus, an insight on environmental influences on the approach is gained from an experiment applying these faults.

5. Evaluation

In this chapter, we evaluate the introduced fusion approach by conducting experiments to answer the established research questions. First, the experimental design, a suited evaluation metric as well as feasibility requirements are described. Afterwards, several experiments investigating the overall performance, the adaptability, and the robustness to faults are presented and evaluated regarding the requirements and the research questions are answered.

5.1 Experimental Design

First, the prerequisites for all experiments are established by describing the underlying environment and the used robots. Then, the overall evaluation methods are introduced to examine the requirements concerning performance, adaptability, and robustness.

5.1.1 Environment

Every experiment is conducted with one robot tracking one teammate. A dynamic and time dependent environment is used to ensure that it is close to real scenario setting. This environment is shown in Figure 5.1. It was created using the SLAM

Figure 5.1: Arena, in which the swarm drives

Toolbox by Macenski et al.[80] integrated in the ROS Framework. Here, one robot drove through a custom arena and mapped the environment with its attached LIDAR

sensor. The black parts represent the boundaries and obstacles of the arena. It is flat, so that robots can not overlook each other through height differences, and has a size of 112x103 cells with a used resolution of 0.05 meters per cell. This means each cell represents a 25 cm^2 part of the arena, and every produced grid covers the whole size of the arena. Further, in a simplified approach neglecting the self localization of the tracking robot, the map is used as input for the static map component of every robot. So, every robot has the same map as the basis to operate on and knows its position in it. Practically, this is not realizable because each robot has to localize itself. However, the additional information provided by this were not utilized. Therefore, it has no impact on the experiments.
Each teammate moves along a predefined path through the arena with a duration of about 40 time steps. At first, the tracking robot follows one teammate around with an incorporated delay, but later strays from following to incorporate obstacles in between the teammate and itself. Hence, line of sight until around time step 12, non line of sight from time step 30 and a mix of both in between these steps, can be tested. When using real robots, various disturbances like Wi-Fi interferences or delays would arise. As we utilize a simulation and aim to examine quality and potential errors, those disturbances are neglected. The influence of different types of faults as described in Section 4.2.4 will be analysed when investigating the robustness.

5.1.2 Robot specs

The robots used for the experiments have a size of about 20x20 centimetres. Therefore, a robot takes up a space of roughly 4x4 cells inside the grid. This space will be used to create a square further called the robot box, which will be used in the following experiment evaluations and is described more precise in 5.1.3. Further, the robots have a maximum velocity of 0.22 metres per second. Hence, as mentioned in Section 4.1.1, the old estimate needs to adjust for possible movement of the robot. On average, this results in about 5 pixels within a sampling interval of 2 seconds. Therefore, the Gaussian filter utilized to smooth the old estimate is initialized with a sigma of 5 according to the velocity. With a normally distributed likelihood measurement, this leads to a possible movement area of 5x5 pixels.
Further, the sensor inputs get recorded at discrete time steps, but may be measured at any time within the time interval between discrete steps. Therefore, the real robot positions at every time step k do not match the position, when each individual sensor input yields the measurement data and therefore the result gets more inaccurate. To counteract this decrease in accuracy, the sensor inputs additionally contain a timestamp, which represents the obtaining time of the raw data. By averaging these raw data time stamps, the mean acquisition time, to which the sensor inputs correspond, can be calculated. And by using this mean time, the real robot position gets interpolated between the time steps k and $k-1$. This rudimentary measure reduces the average position error by up to 2 centimetres. In the following, only the interpolated real robot position is used to calculate the evaluation metrics.

5.1.3 Evaluation metrics

The evaluation metric consists of a relative and an absolute part. Because of the grid property of an overall grid cell sum of 1 described in the methods Section 4.1.2,

we can assume a probability density functions on the result grid. Therefore, we can evaluate different parts of the grid separately in terms of probability and make statements about the relative accuracy as follows. Here, a square of cells in the grid with size equal to the previously estimated robot size is marked. The centre of this square represents the real position of the teammate, whose result grid we want to evaluate. Therefore, it is called the robot box Box_r. The probability density function, inside the robot box, gets compared to the overall probability density function of the grid without the robot box. This is called the outside box Box_o. Additionally, another square the size of the robot with the highest probability density function in the whole grid is determined and used for further evaluation of the results. This is called the maximum box Box_m. Here, because of the grid adding up to 1, the concrete average probabilities in each box can be compared. Mathematically, this can be described as follows. Let $c \in C$ be a cell of all cells in the grid, with $P(c)$ the probability in the cell. Additionally, let $c_r \in Box_r$ and $c_m \in Box_m$ be a cell of the respective box. The average probability $P(Box)$ of both robot and maximum box can then be calculated with Equation 5.1.

$$P(Box) = \sum_{c \in Box} \frac{P(c)}{|Box|} \tag{5.1}$$

Moreover, let $c_o \in C \setminus Box_r$ be a cell in the outside box. The average probability of this outside box Box_o can then be calculated as seen in Equation 5.2.

$$P(Box_o) = \sum_{c_o \in C \setminus Box_r} \frac{P(c_o)}{|C| - |Box_r|} \tag{5.2}$$

Then with the help of the above, a threshold of 50% higher average probability values inside the robot box $P(Box_r)$ than outside of it $P(Box_o)$ is used to determine if the teammate was found or something went wrong and the teammate could not be tracked. Because of narrow distributed high probability values in the grid, a threshold of 50% was chosen to exclude noise on the cell probabilities but to still include not well performing measurements and to further evaluate them. This relative metric of the evaluation utilizing a gradient in certainty can therefore be described as follows.

$$TeammateFound = P(Box_r) \; > \; 1.5 \cdot P(Box_o) \tag{5.3}$$

Then, if a teammate was found, the absolute accuracy as distance between the result estimate, which is the cell with the highest likelihood in the maximum box, and the real position gets calculated. This happens by comparing the two-dimensional position of the grid cell with the highest likelihood of residence $\vec{p}_m$ to the two-dimensional real position of the teammate $\vec{p}_r$. Hence, the absolute accuracy can then be calculated in metres, as shown in Equation 5.4.

$$Distance = \sqrt{(x_r - x_m)^2 + (y_r - y_m)^2} \tag{5.4}$$

5.1.4 Requirements of Successful Teammate Tracking

The previously defined metrics can be used to check if the requirements for a successful teammate tracking apply. Hereby, the requirements are based on the research

questions about to be answered. First, the accuracy requirement of the yielded result, which is related to the size of the used robots, is described. As described in Section 5.1.2, the robot has a size of about 20x20 centimetres. Therefore, a mean distance from real to calculated position of the teammate shorter than 30 centimetres with an associated standard deviation of under 15 centimetres would be a feasible requirement in terms of accuracy. Second, a robustness requirement is established. Here, in case of an input faults, whether failure or error, the uncertainty of the result is still be decreased. This means, a gradient of certainty inside the result grid exists and can be confirmed through the relative evaluation metric described earlier in Equation 5.3. The mean absolute distance yielded with input faults will be larger according to the seriousness of the failure. Third, adaptability to different use cases is a requirement for a successful tracking. Therefore, by manually adjusting parameters inside the tracking approach, different results for different environments can be expected. The difference in these results can be confirmed by the relative and absolute evaluation metric.

In the following, the conducted experiments are analysed and evaluated regarding the feasibility of the Bayesian Fusion approach according to the defined requirements. Here, every section describes the experimental design at first. Then, the gathered data is presented, and lastly those data is evaluated in terms of accuracy and feasibility using the previously defined metrics.

5.2 Experiments for Independence of Sources

This experiment is conducted to give insight into in chapter background described stochastic independence 2.5 between the input sources. Independence is needed, but dependence is inevitable between the different sources. So, this experiment should show the correlation between the sources described earlier and investigate how large this correlation becomes.

5.2.1 Results

The two-step test described earlier was conducted with two robots over 40 different runs.
The first part of the test wants to examine the linear dependence between the sensor sources. This is done by utilizing the covariance correlation. So, for each run, the measurement data of all input sensors is collected and pair-wise compared with each other. The resulting covariance values of all runs are shown as box plots for each sensor pairing in Figure 5.2. As you can see, the x-axis shows the correlation pairings and the y-axis shows the degree of correlation. The box plots for each pairing differ in value and interval size. The box plot related to the correlation between laser points and the laser area has the smallest interval size, and the interquartile range is located in the interval $[0.01, 0.05]$. Here, the median of the data set is located at 0.02. The maximum of this box plot is at 0.08 and the minimum gets negative with a value of -0.025. The box plot showing the correlation between the range and the laser area has a larger variance in the data, with the interquartile range located in the interval $[0.1, 0.17]$, a maximum at 0.23 and a minimum at 0.02. The median

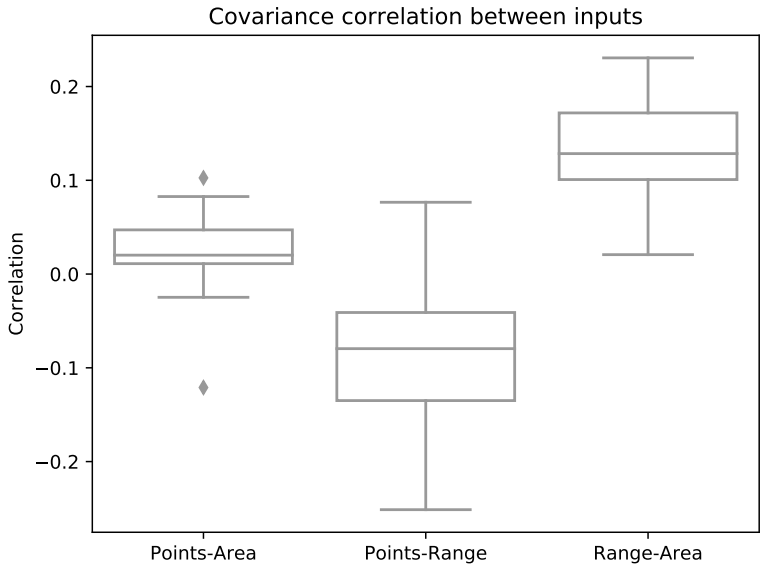

Figure 5.2: Covariance Correlation between the sensor input sources laser points, laser area, range

of the data set is hereby located at 0.13. The last box plot of the three, the laser points and range measurement correlation, is located mostly below zero and has a high variance, contrary to the other two. Here, the interquartile range is located in the interval $[-0.135, -0.04]$ with a median at -0.08. The minimum reaches as low as -0.25, and the maximum is located at a positive value of 0.077. For every box plot ordered as described, the interval size between the 0th percentile and the 100th percentile increases by approximately 1.0. Further notable for every box plot, the data sets are skewed towards 0.0 and therefore tend to be more accurate towards zero.

The second part of the test is designed to examine, whether the measured sensor data comes from a multivariate normal distribution by analyzing the *pValue* of the measurement data for each sensor pairing per run. This *pValue* is a two-sided chi squared probability to test the hypothesis, that the values are drawn from a joint normal distribution. It is tested against a threshold to specify whether to reject or accept the hypothesis. In this case, the threshold has a typical value of 0.05 and the *pValue* of 40 runs is plotted as box plots. This can be seen in Figure 5.3. Here, the *pValue* is shown on a logarithmic scale and the used threshold is drawn as a horizontal dashed line. The box plot for the laser points to laser area has the lowest value range with an interquartile range located in the interval $[1.49 \cdot 10^{-4}, 3.9 \cdot 10^{-4}]$ with a median at $2.83 \cdot 10^{-4}$. The minimum is located far below this at $1.3 \cdot 10^{-5}$, but outliers even range to almost the threshold with a maximum at 0.046. The *pValues* for the laser points to range distribution test have a minimum of $1 \cdot 10^{-5}$ and a maximum of 0.022. Hereby, the interquartile range is located more to the maximum within the interval between $[1.24 \cdot 10^{-4}, 3.2 \cdot 10^{-3}]$ with a median at $7.75 \cdot 10^{-4}$. The box plot of the range to laser area joint distribution has the interquartile range with the highest *pValues* of all the three multivariate distributions. This interval is located between $[2.28 \cdot 10^{-3}, 2.19 \cdot 10^{-2}]$. Here, the median is located at $7.67 \cdot 10^{-3}$.

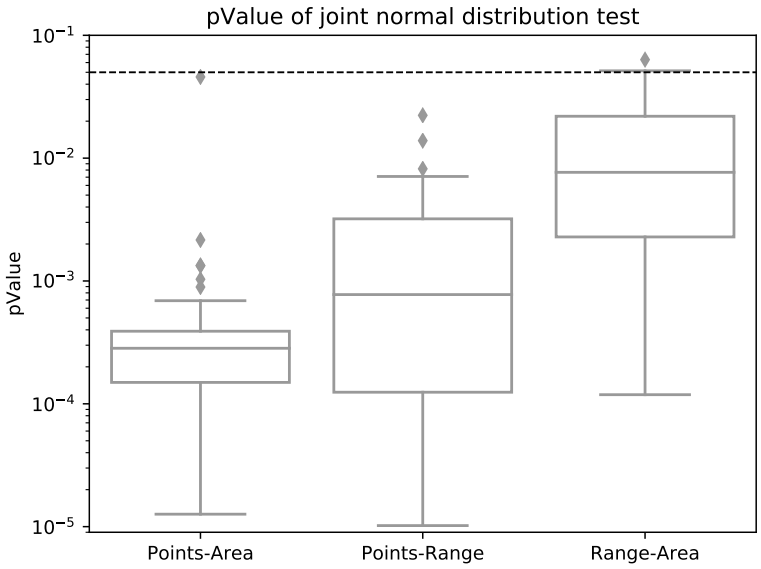

Figure 5.3: *pValue* from joint normal distribution test of input sources

In contrast to the other plots, the 100th percentile is located above the threshold of 0.05.

5.2.2 Evaluation

The experiment was divided into two parts, linear dependence correlation and joint normal distribution origins. The parts get evaluated separately, and the results are then combined for an overall evaluation regarding stochastic independence.
For the first part, all correlations between the sensor pairings tend to be near zero, with intervals including the desired value of zero, which represents a linear independence. This means linear independence is not strictly given, but the correlation between the sensors is only small. Additionally, the box plots indicate a possible influence by errors of different intensity, because of the skewness towards values near zero. That means, the closer the values are to zero, the less variance is in the data. Hence, the correlation might be even smaller.
For the second part, *pValues* above the marked threshold of 0.05 mean the hypothesis, whether the data originates from a normal distribution, can not be rejected. Thus, the sampled data is likely to have a normal distributed origin. If the *pValue* is smaller than the threshold, the hypothesis can be rejected and the sampled data has likely no origin in a normal distribution. The *pValue* box plots of the joint distribution of laser points and laser area are located far below the defined threshold. Only 1 outlier scratches at this threshold, but it is not significant. Therefore, the sampled values from this input correlation are very likely not to originate from a multivariate normal distribution. The *pValues* of the correlation between laser points and range measurements are located closer to the threshold and lie within a wider interquartile range, which encapsulates the points-area interquartile range between the 25th and the 50th percentile. That means, more than 50% of the *pValues* yielded are larger than the points-area *pValues*. This means, the points-range data is more likely to

come from a joint normal distribution than the points-area, but still not exceeds the threshold. The *pValues* yielded with the range and laser area correlation are even larger than the points-range values, which only reach to the 50th percentile at maximum. Few values between the 75th and 100th percentile exceed the threshold. Therefore, it is likely that this data originates from a normal distribution with consideration of possible errors. Hence, according to the *pValues* only one input correlation is likely to originate from a joint normal distribution.
In summary, the linear dependence correlation between all sensor pairings is very small to zero, which implies a near linear independence. Additionally, the range and laser area correlation set is likely to be drawn from a joint normal distribution. Therefore, in combination and under consideration of possible errors, near stochastic independence between the range and laser area input can be assumed. The other two input correlations can only be assumed to be linear independent. Nevertheless, this supports the usage of naive Bayes assumptions for the use with the Bayes theorem. Because linear independence can be assumed for all correlations and stochastic independence for one out of three, it may not interfere largely with the functionality of the Bayes theorem, as additionally deviations are not preventable in the robotic use case as described earlier in Section 2.5.

5.3 Experiments for Bayesian Fusion

This experiment is used to examine the accuracy of the tracking and is therefore directly correlated to the feasibility of the Bayesian Fusion. Here, the likelihood of residence inside the robot box in the result grid gets compared to the real position box of the teammate and the likelihood outside the robot box. First, the relative probabilities are used to identify if an increase in certainty happened and the teammate was found and afterwards the absolute positions get compared in terms of accuracy. This experiment is done to get insight into the feasibility defined by accuracy and deviation as described in Section 5.1.4.

5.3.1 Results

In the following, the results of the conducted experiment are shown. First, the relative probabilities inside the grid get compared. This is shown in Figure 5.4.

This figure contains box plots showing the average likelihood in the result grid on a log-scale over every experimental run.

The left box plots show the average probabilities inside the robot box, and the dashed line marks the mean over all runs. This mean of average probabilities is $8.59 \cdot 10^{-5}$ and the associated standard deviation is $6.82 \cdot 10^{-5}$.

The box plots in the middle show the average probability outside the pre-defined robot box, with a dashed line as mean as well. Here, the mean has a value of $6.46 \cdot 10^{-6}$ with a standard deviation of $9.17 \cdot 10^{-7}$. The overall deviation and thus the interquartile ranges are significantly smaller than inside the robot box and have smaller values so that the values do not overlap once. Here, the calculated mean over all average probabilities inside the robot box is over 13 times larger than the mean calculated outside the robot box.

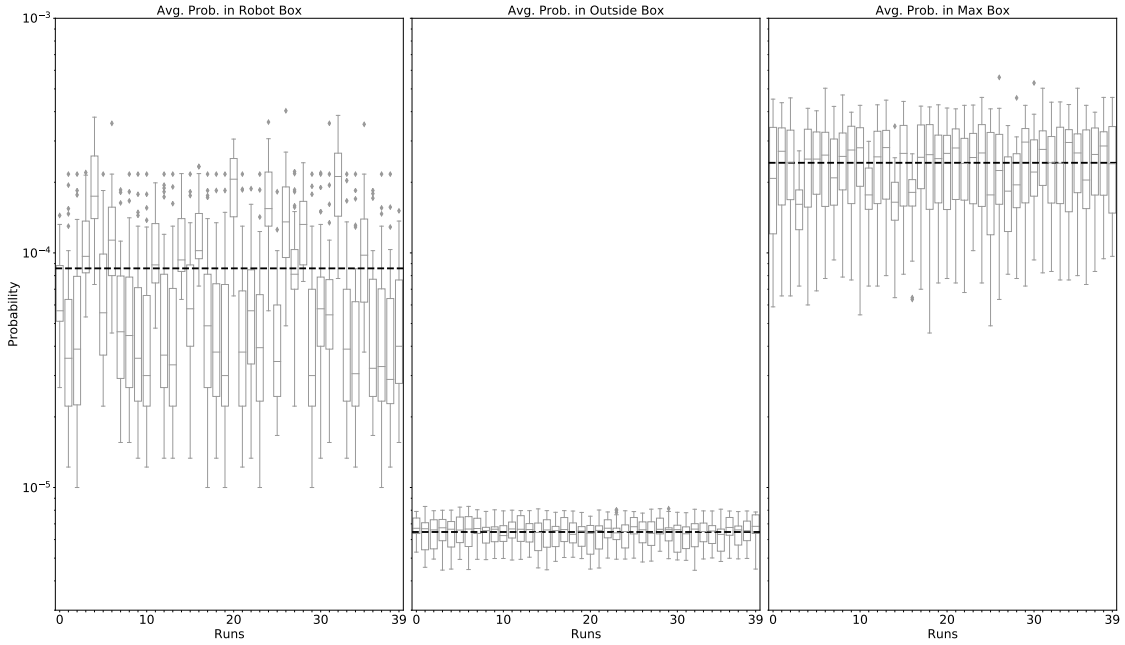

Figure 5.4: Overall Accuracy: average probability in robot box, outside robot box, and around max probability

The box plot on the right shows the average probability in a square equal to the size of the robot box with the maximum average probability, the maximum box in our grid estimate. The average probabilities in the maximum box have a mean value of $2.42 \cdot 10^4$ with an associated standard deviation of $9.84 \cdot 10^5$. As shown here, the mean of average likelihood is still higher than that of the robot box, but the intervals are overlapping. This means the probability inside the robot box is closer to the maximum likelihood than to the minimum likelihood of residence.
These plots show that the average probabilities inside the robot box exceeds the threshold of at least 50% higher probability values calculated with Equation 5.3. Therefore, the teammate was found, and the distance was calculated with Equation 5.4. The absolute distances in metres over every experimental run are shown as box plots in Figure 5.5. Here, only 2 out of 40 runs have interquartile ranges significantly exceed the 50 centimetres mark. The third quartile of the interquartile ranges of those runs can reach up to about 1 metre at maximum. The distribution in all interquartile ranges is not symmetrical, but heavily skewed to the left. This means 25% of all values in the interquartile range are crowded in a significantly smaller range from the first quartile to the median, and values above the median are wider distributed. Therefore, more accurate distances gather inside smaller intervals and distances, that are more inaccurate, are located in wider intervals also including scattered outliers of up to over 2 metres. The mean over all distance values yielded by all experimental runs is 33.85 centimetres, with a standard deviation of 33.95 centimetres overall. The mean is shown as the horizontal line in the figure.
A latency measurement, shown in Figure 5.6, can be used to examine the correlation between the latency behaviour of the input sources and the distances per run. Here, a significant standard deviation was rare and only occurred on run 11 with 1.5 seconds deviation on the range input, on run 15 with a deviation of 1 second for the laser area input and on run 28 with all 3 sensor inputs having a deviation with 3 seconds

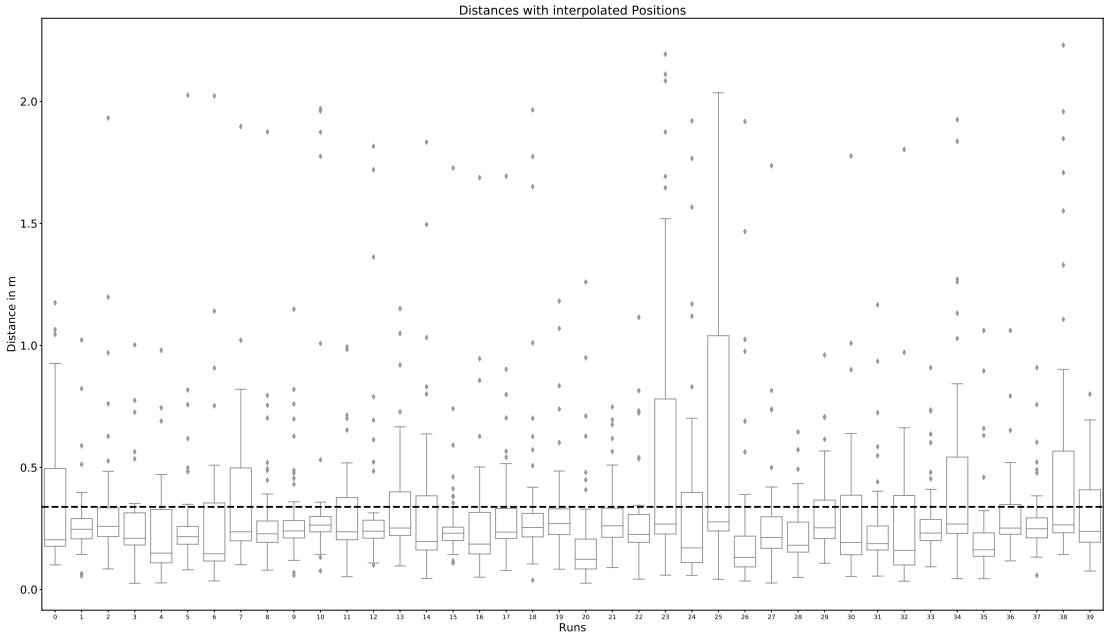

Figure 5.5: Overall Accuracy: absolute distance between calculated position and real robot

on the laser area input, 1.5 seconds on the laser points input and 0.5 seconds on the range input.
The former displayed distances incorporate values yielded over the whole run, including change from line of sight to non line of sight scenarios. To further analyse the feasibility of the Bayesian fusion and get insight into the influence of this non line of sight scenarios, the conducted experiments get plotted over time. This is shown in Figure 5.7. Here, the calculated distance from all experimental runs is plotted over the corresponding measurement time step as an average distance with the 95% confidence interval. In the first 10 time steps, an average distance of about 20 centimetres is achieved. Then, this average distance increases to 30 centimetres, with a deviation of about 10 centimetres. At time step 20, the average distance quickly increases to 60 centimetres, and the corresponding deviation sees an increase of about 10 centimetres as well. Between time steps 27 and 32, a short decrease in distance can be observed, which is immediately followed by a steep increase in the average distance to 1 metre. After that, an even steeper decrease to an average distance of again 20 centimetres can be seen in the time steps 38 to 40.

With the provided information, the experiment data can be further set into context.

5.3.2 Evaluation

The yielded distances over time shown in Figure 5.7 can be mainly explained by the path the robot takes in the environment. As described earlier, the path starts by following the teammate around. Here, many laser measurement points hit the teammate and the radius of the range sensor measurement is small. This leads to the small resulting distance of about 20 centimetres. As the experiment progresses, the teammate moves further away from the tracking robot and an object blocks a part of the line of sight to the teammate. This leads to a decrease in laser measurements, that hit the teammate, as well as a larger radius yielded by the range measurements.

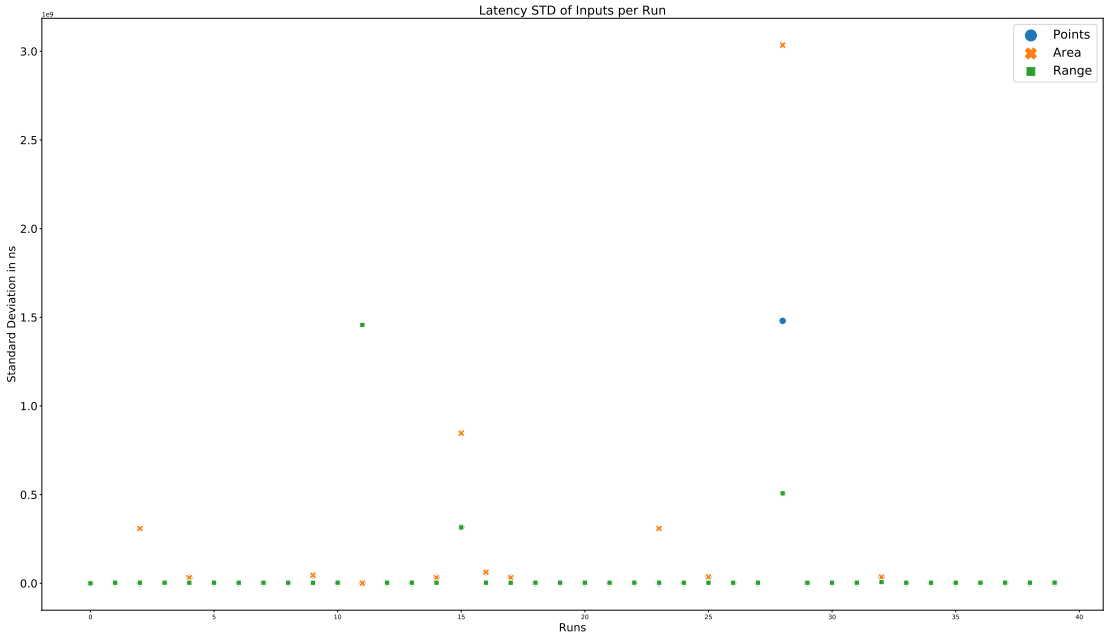

Figure 5.6: Standard Deviation of Latency per Run

Therefore, the yielded distance and its corresponding confidence increases both by 10 centimetres. Shortly before the line of sight between the robot and the teammate is fully lost, the robot gets again closer to the teammate, which results in a dip in distance to starting level accuracy. Thereafter, the line of sight is lost and the yielded distance increases to an average of 60 centimetres, mainly calculated by the range measurements and the old estimate, which restricts the range measurement into a region near the point where the teammate was visible at last. Then, still in a non line of sight scenario but slightly mixed with partially line of sight moment, the yielded distance drops again. This is caused by the robot and teammate to partially see each other for a brief moment and getting closer again and therefore the radius of the range measurements get smaller, which both leads to an increase in accuracy. The following steep increase in distance to an average of about 1 metre is correlated to the duration of the full non line of sight scenario and a further stray of the robot from the teammate. Here, the radius of the range measurements get larger again and the old estimate, which limited the region where the teammate is expected to be, gets blurred every time step according to the possible movement to a point, where it does not restrict the area to a point where the teammate can be anywhere on the radius. Lastly, the scenario turns into a full line of sight again and the yielded distance has a steep decrease to about 20 centimetres again.
This last decrease in distance shows how well the reappearance of a teammate in line of sight gets included into the result, and how the overall value of the pinpointing ability of the laser measurements affects the yielded result. Further, the described correlation of non line of sight scenarios and yielded distance provides information about how the fusion method, fusing line of sight dependent and independent input sources, dampens the impact of losing the line of sight on the teammate. Instead of a sudden spike in distance, a gradually decreasing accuracy over time is achieved, which is corrected at any time a line of sight scenario is established again.
The line of sight scenario in this experiment yields results of 20 centimetres. The

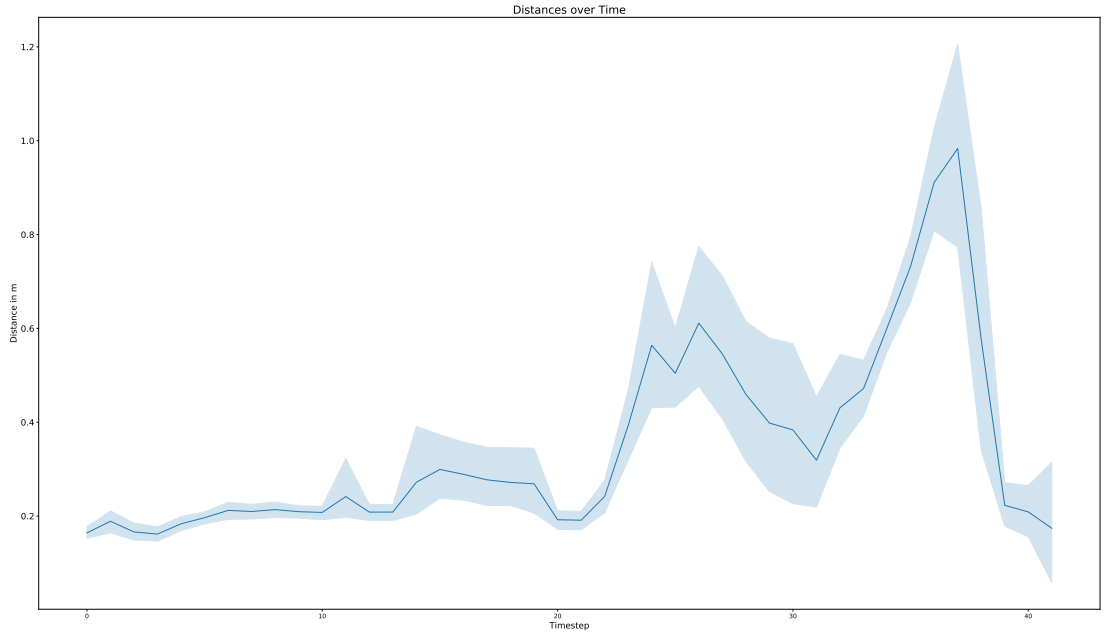

Figure 5.7: Overall Accuracy: absolute distance over time between estimate and real robot

results for the non line of sight scenario are related to the size of the arena, which determines how far the teammate can stroll away from the robot. Here, in this experimental setup, a maximum average distance of about 1 metre is achieved.
Taking a look at the distance box plots, consisting of time steps in line of sight and non line of sight, plotted over the whole run in Figure 5.5, the overall mean distance value yielded is 33.85 centimetres. Notable here, the average distance lies above most interquartile ranges and is only incorporated by them when considering the standard deviation of 33.95 centimetres. This means this average gets strongly influenced by some large outliers. These outliers may occur due to errors during the experimental run, which differ in intensity and impact on the yielded distance and thus result in a wider distribution and variance of these outlier values. In contrast, the interquartile ranges have a small distribution and are skewed to the left, meaning the more accurate the yielded values are, the less variance is in the distance data. This also supports the assumption of errors of different intensity.
Hereby, to examine the relation between those distance outliers and the latency of the input sources, the latency standard deviation was measured in Figure 5.6. Although there are some significant deviations on latency of the input sources, it can not be concluded that those latencies are related to the distance outliers because no run with significant latency overlaps with a run of wide distributed and high distances. It may lead to some small deviation in distances, but is not the cause of the significant worse distance measurements. Therefore, other errors, except latencies, must have happened and are to blame for high yielded distances. This good distance values on runs with high latency deviation support the mutual compensation of faulty input measurements expected for a fusion approach. Hence, it shows increased reliability against latency errors.
In context to the former mentioned average yielded distances, the box plots of probabilities in Figure 5.4 show a constant gradient of certainty established inside the result grid, which means the uncertainty of the teammate location is always reduced.

But additionally, the probability plots show a roughly constant difference between the probabilities inside the robot box and the box around the maximum of the grid, that may be caused by the placement of the robot box. The robot box is assumed to be centred around the centre of the robot. But the yielded measurements of the single inputs either detect the edge of the robot for the laser point input or detect the range to the place of the mounted range sensor as described in Section 4.2.2, and not the centre of the robot. This shift results in a possible offset of the robot box by half the length of a robot box due to the laser sensor. This causes the robot box and the box around the highest measurement to still overlap and increase the average probability of the robot box, but it will never reach the maximum. Following this, the yielded distances shown in the latter figures include this offset as well, because the centre of the robot is used to calculate them. Therefore, the distances may be smaller than measured by an offset about half the size of the used robots, which is 10 centimetres.
The average yielded distance over all experimental runs is slightly above the 30 centimetre requirement for successful tracking defined beforehand in Section 5.1.4. Here, it is mentionable that line of sight scenarios easily exceed the requirement, whereas non line of sight scenarios mostly can not fulfil the requirement, but prevent the yielded distance from exponential increase, as soon as the line of sight is lost. Additionally, by combining the information about the box offset, the line of sight scenario yields even better results, which is transferred to non line of sight scenarios by the old estimate. Therefore, the requirement for successful tracking is fulfilled. A comparison with other state-of-the-art fusion approaches shows that the yielded results can compete with values yielded by Kalman fusion with 0.05 and 0.14 metres as translation error. In addition, most values yielded by the Bayes-Hempel Method are located in the lower part of the interval range of values yielded by the deep learning fusion and therefore can compete here as well. Both other fusion approaches are described in Section 3.2. Notable here is that depending on the chosen sensor inputs and on other aspects of the Bayes-Hempel Method like the grid cell size, additional improvements on the result can be made.

5.4 Experiments for Influence of Sensor Sources

This experiment is conducted similar to the experiment on the overall Bayesian fusion before. But this time, instead of an equal distribution of weights, the selected source gets assigned a higher weight for every run. Here, the weights still add up to a sum of 1. The selected source gets a weight of 0.5 assigned, whereas the weight of every other source is equally distributed among the remaining 0.5. These weights have severe effects on the probabilities within the fusion. A weighted sensor has an almost five times higher effect on the result certainty of the grid, than a less weighted one. Therefore, the influence of the sources on the result is increased or decreased, depending on the current environment. This means, the information about the environment yielded by the weighted sensor is significantly increased, but the information yielded by the non-weighted sources are kept relevant. Hereby, it is expected for sources using the LIDAR to perform better in line of sight scenarios, whereas line of sight independent sources like ultra-wideband range are expected to perform more constant overall and better in non line of sight scenarios. The

weighting on the old estimate should make the result less affected by current changes in the environment, and therefore smooth the result at the cost of a small delay.

5.4.1 Results

Weight on Laser Points

First, the relative probabilities inside the grid get compared like in the experiment conducted before. The mean of average probability over all boxes is numerically shown in Table 5.1. The mean probability outside the robot box is significantly

RobotBox	OutsideBox	MaxBox
$8.28 \cdot 10^{-5}$	$6.47 \cdot 10^{-6}$	$2.39 \cdot 10^{-4}$

Table 5.1: Weighted laser points: Average probabilities in parts of the grid

smaller than inside the robot box. The MaxBox square equal to the size of the robot box around the maximum average likelihood in our grid estimate is still higher than that of the robot box, but with incorporated standard deviation the robot box probability is closer to that of the MaxBox than to that of the OutsideBox. This means the probability inside the robot box is closer to the maximum likelihood than to the minimum likelihood of residence of the teammate.
This shows that the average probability inside the robot box exceeds the threshold of at least 50% higher probability values calculated with Equation 5.3. Therefore, the teammate was found in every run and the distance was calculated with Equation 5.4. The absolute distances in metres over every experimental run are shown as box plots in Figure 5.8. Here, the interquartile ranges are mainly located in intervals below 50

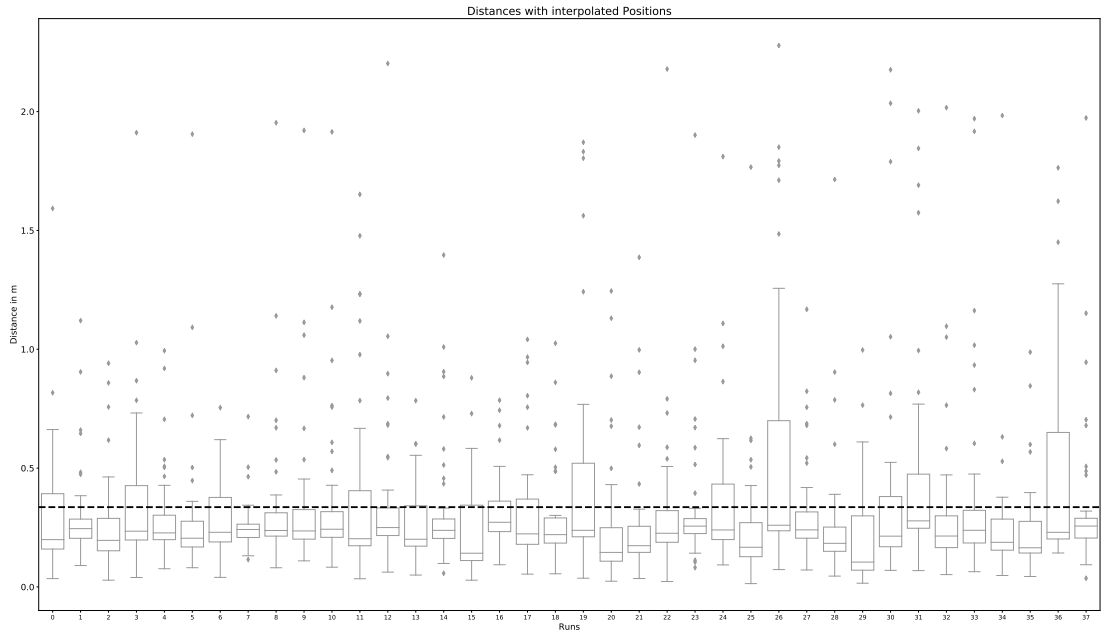

Figure 5.8: Weighted laser points: Absolute distance between calculated position and real robot

centimetres, except for 3 out of 38 runs. The distribution in the interquartile ranges

for every run is not symmetrical, but heavily skewed to the left. This means 25% of all values in the interquartile range are crowded in a significantly smaller range from the first quartile to the median, and values above the median are wider distributed. Therefore, more accurate distances gather inside smaller intervals and distances, that are more inaccurate, are located in wider intervals. The mean over all distance values yielded by all experimental runs is 33.46 centimetres and is shown as the horizontal line in the figure. Hereby, the corresponding standard deviation is 33.67 centimetres. The former box plots incorporate values yielded over the whole run, including change from line of sight to non line of sight scenarios. Therefore, the experimental results get plotted over time. This is shown in Figure 5.9. Here, the calculated distance

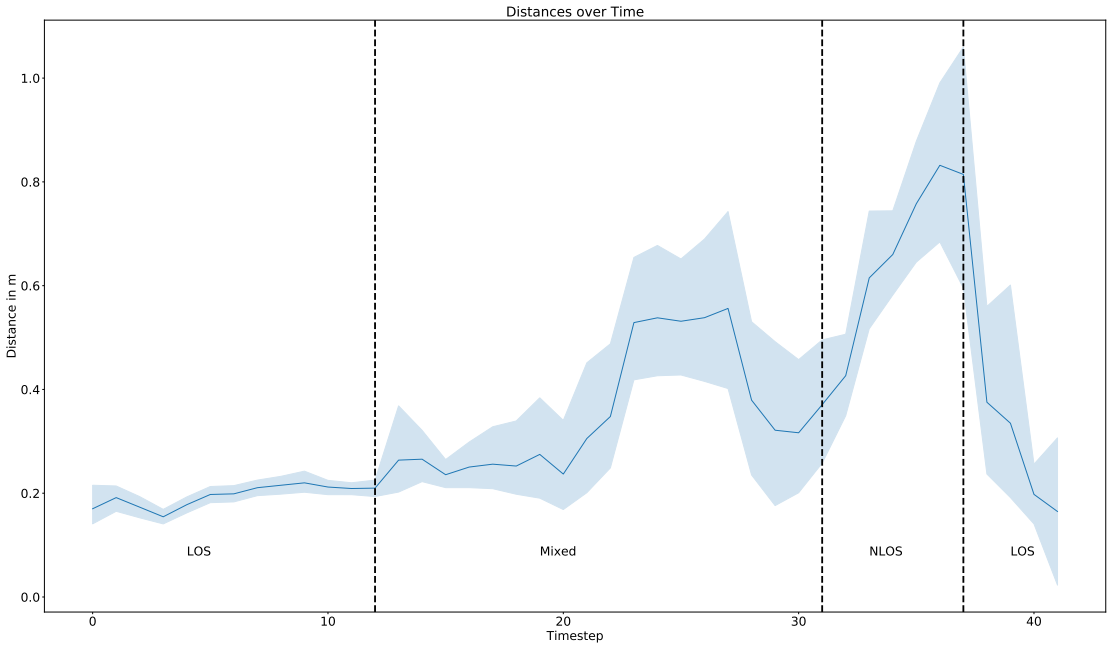

Figure 5.9: Weighted laser points: Absolute distance over time between estimate and real robot

from all experimental runs is plotted over the corresponding measurement time step as an average distance with the 95% confidence interval. The time axis is separated into different sectors according to the line of sight scenario in these time steps. These sectors are line of sight until time step 12, mixed line and non line of sight between time steps 12 and 31 and non line of sight after time step 31 until time step 38. In the first 12 time steps, an average distance of about 20 centimetres is achieved, including little to no confidence deviation. Then, this average distance increases by 5 centimetres, with a confidence deviation of about 10 centimetres. At time step 20, the average distance quickly increases to about 55 centimetres, and the corresponding confidence deviation sees an increase of up to 15 centimetres to an overall 25 centimetres. Between time steps 27 and 30, a short decrease in average distance back to 30 centimetres can be observed, maintaining a confidence deviation of 25 centimetres. This is followed by an increase of the average distance to 85 centimetres and an increase in confidence deviation to 30 centimetres. After that in time steps 38 to 40, the average distance again decreases to 20 centimetres, but keeping the confidence deviation high with values of up to 20 centimetres. Further, the mean and the corresponding standard deviation in metre for each earlier described sector are shown in the following Table 5.2.

LOS mean$\pm\sigma$ in m	Mixed mean$\pm\sigma$ in m	NLOS mean$\pm\sigma$ in m
0.195 ± 0.064	0.355 ± 0.374	0.619 ± 0.392

Table 5.2: Weighted laser points: Mean and standard deviation of sectors

Weight on Laser Area

First, the relative probabilities inside the grid get compared. The mean of average probability over all boxes is numerically shown in Table 5.3. The mean probability

RobotBox	OutsideBox	MaxBox
$9.12 \cdot 10^{-5}$	$6.51 \cdot 10^{-6}$	$2.27 \cdot 10^{-4}$

Table 5.3: Weighted laser area: Average probabilities in parts of the grid

outside the robot box is significantly smaller than inside the robot box. The MaxBox square equal to the size of the robot box around the maximum likelihood in our grid estimate is still higher than that of the robot box, but with incorporated standard deviation the robot box probability is closer to that of the MaxBox than to that of the OutsideBox. This means the probability inside the robot box is closer to the maximum likelihood than to the minimum likelihood of residence of the teammate. This shows that the average probability inside the robot box exceeds the threshold of at least 50% higher probability values calculated with Equation 5.3. Therefore, the teammate was found in every run and the distance was calculated with Equation 5.4. The absolute distances in metres over every experimental run are shown as box plots in Figure 5.10. Here, the interquartile ranges are mainly located in intervals below

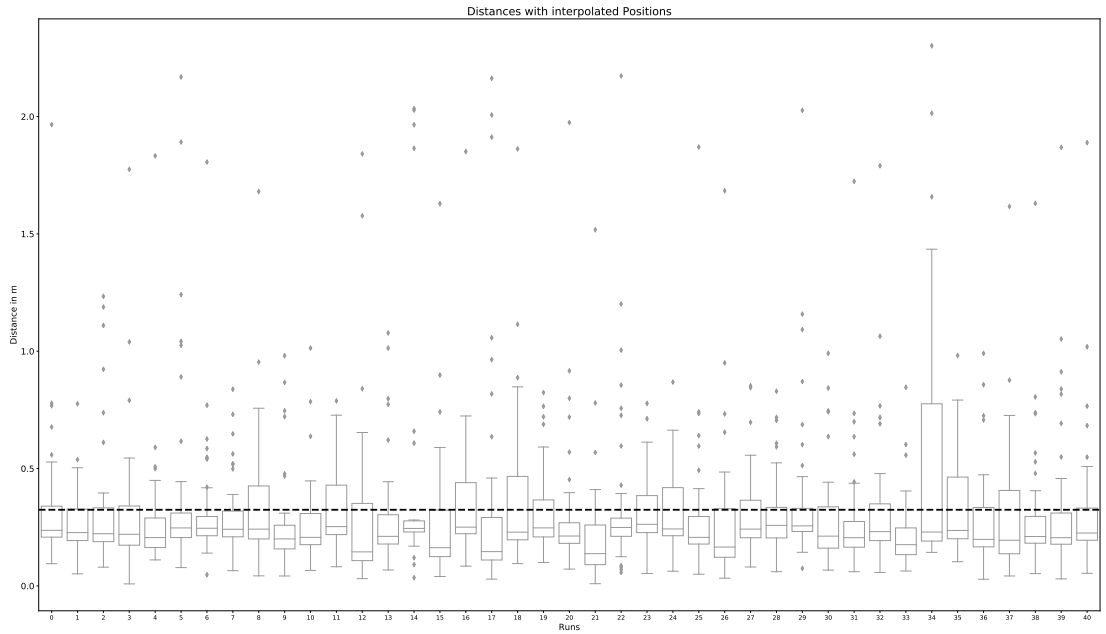

Figure 5.10: Weighted laser area: Absolute distance between calculated position and real robot

50 centimetres, except for 1 out of 41 runs. The distribution in the interquartile

ranges for every run is not symmetrical, but heavily skewed to the left. This means 25% of all values in the interquartile range are crowded in a significantly smaller range from the first quartile to the median, and values above the median are wider distributed. Therefore, more accurate distances gather inside smaller intervals and distances, that are more inaccurate, are located in wider intervals. The mean over all distance values yielded by all experimental runs is 32.45 centimetres and is shown as the horizontal line in the figure.
The former box plots incorporate values yielded over the whole run, including change from line of sight to non line of sight scenarios. Therefore, the experimental results get plotted over time. This is shown in Figure 5.11. Here, the calculated distance

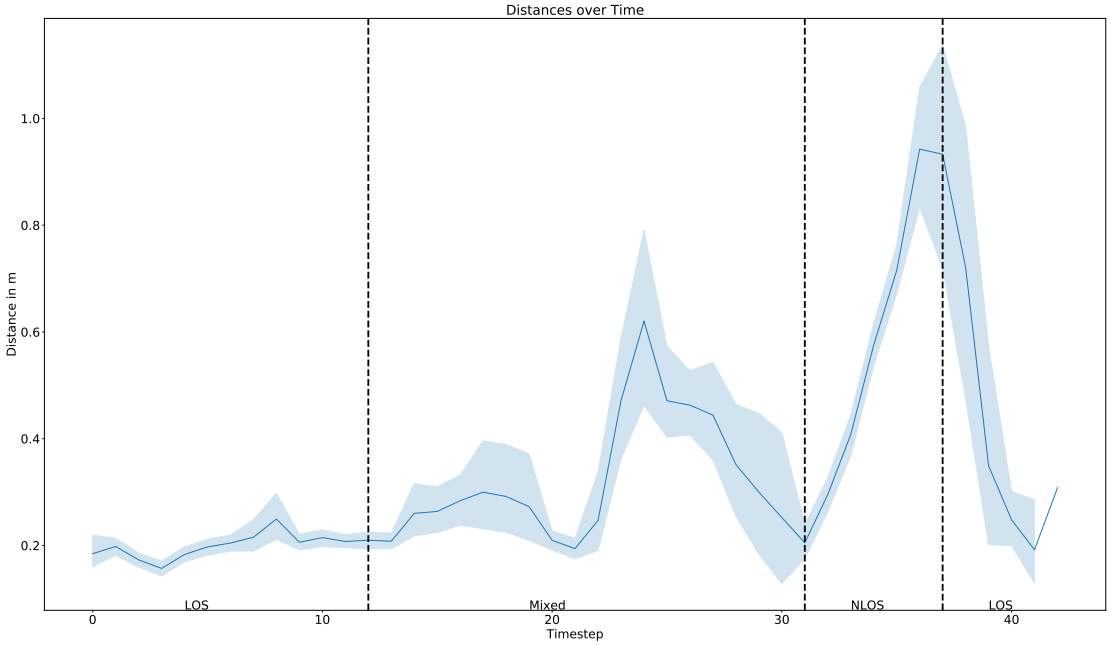

Figure 5.11: Weighted laser area: Absolute distance over time between estimate and real robot

from all experimental runs is plotted over the corresponding measurement time step as an average distance with the 95% confidence interval. The time axis is separated into different sectors according to the line of sight scenario in these time steps. These sectors are line of sight until time step 12, mixed line and non line of sight between time steps 12 and 31 and non line of sight after time step 31 until time step 38. In the first 10 time steps, a distance of slightly below 20 centimetres and slowly increasing to above 20 centimetres resulting in an average distance of about 20 centimetres is achieved, including 2 to 5 centimetres deviation. Then at time step 13, this average distance increases to 30 centimetres at time step 17, with a maximum deviation of about 10 centimetres. At time step 20, the average distance has a small dip to 20 centimetres and then quickly increases to about 60 centimetres, and the corresponding deviation sees an increase to an overall deviation of 20 centimetres at the peak. Between time steps 25 and 32, a short decrease in average distance back to 20 centimetres can be observed, maintaining a deviation of 20 centimetres. This is followed by a steep increase of the average distance to 95 centimetres, with little to no deviation during the increase, resulting in a maximum deviation of 20 centimetres at time step 37. After that in time steps 38 to 41, the average distance

again decreases to 20 centimetres, but keeping the deviation high with values of up to 20 centimetres. Further, the mean and the corresponding standard deviation in metre for each earlier described sector are shown in the following Table 5.4.

LOS mean$\pm\sigma$ in m	Mixed mean$\pm\sigma$ in m	NLOS mean$\pm\sigma$ in m
0.2 $\pm$0.074	0.328 $\pm$0.313	0.559 $\pm$0.376

Table 5.4: Weighted laser area: Mean and standard deviation of sectors

Weight on Range

First, the relative probabilities inside the grid get compared. The mean of average probability over all boxes is numerically shown in Table 5.5. The mean probability

RobotBox	OutsideBox	MaxBox
$9.40 \cdot 10^{-5}$	$6.45 \cdot 10^{-6}$	$2.47 \cdot 10^{-4}$

Table 5.5: Weighted range: Average probabilities in parts of the grid

outside the robot box is significantly smaller than inside the robot box. The MaxBox square equal to the size of the robot box around the maximum likelihood in our grid estimate is still higher than that of the robot box, but with incorporated standard deviation the robot box probability is closer to that of the MaxBox than to that of the OutsideBox. This means the probability inside the robot box is closer to the maximum likelihood than to the minimum likelihood of residence of the teammate. This shows that the average probability inside the robot box exceeds the threshold of at least 50% higher probability values calculated with Equation 5.3. Therefore, the teammate was found in every run and the distance was calculated with Equation 5.4. The absolute distances in metres over every experimental run are shown as box plots in Figure 5.12. Here, the interquartile ranges are mainly located in intervals below 50 centimetres, except for 1 out of 41 runs. The distribution in the interquartile ranges for every run is not symmetrical, but heavily skewed to the left. This means 25% of all values in the interquartile range are crowded in a significantly smaller range from the first quartile to the median, and values above the median are wider distributed. Therefore, more accurate distances gather inside smaller intervals and distances, that are more inaccurate, are located in wider intervals. The mean over all distance values yielded by all experimental runs is 30.82 centimetres and is shown as the horizontal line in the figure.
The former box plots incorporate values yielded over the whole run, including change from line of sight to non line of sight scenarios. Therefore, the experimental results get plotted over time. This is shown in Figure 5.13. Here, the calculated distance from all experimental runs is plotted over the corresponding measurement time step as an average distance with the 95% confidence interval. The time axis is separated into different sectors according to the line of sight scenario in these time steps. These sectors are line of sight until time step 12, mixed line and non line of sight between time steps 12 and 31 and non line of sight after time step 31 until time step 38. In the

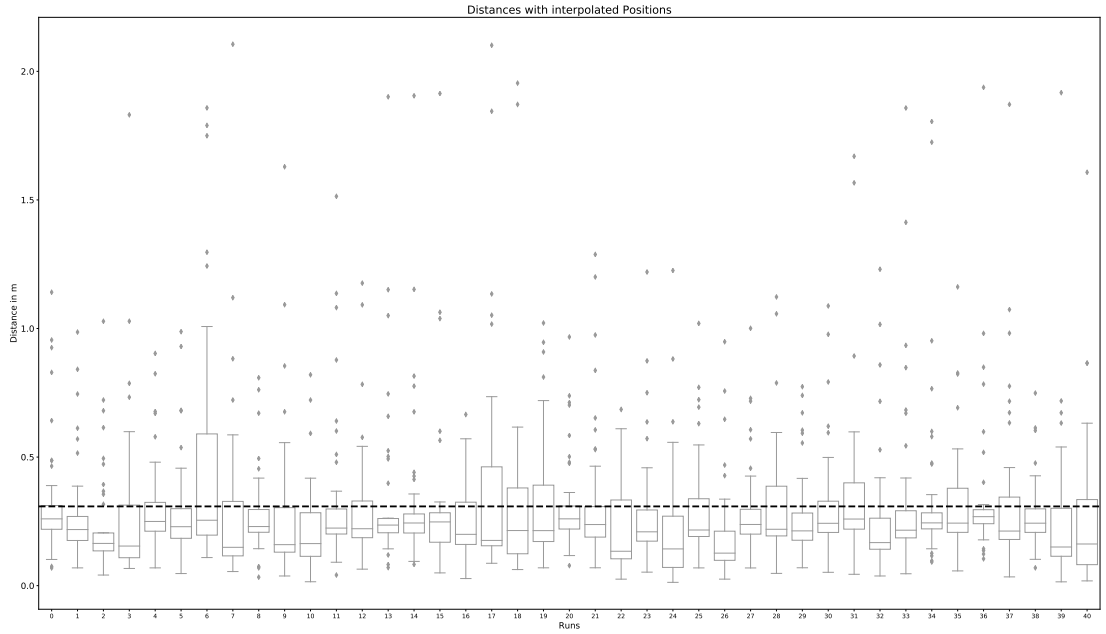

Figure 5.12: Weighted range: Absolute distance between calculated position and real robot

first 15 time steps, an average distance of about 20 centimetres is achieved, including 2 to 5 centimetres deviation. Then, this average distance increases by 10 centimetres, with a deviation of about 10 centimetres. At time step 20, the average distance shortly decreases to 20 centimetres before a quick increase to about 50 centimetres. Hereby, the corresponding deviation stays the same. Between time steps 24 and 31, a decrease in average distance back to 20 centimetres can be observed, but partly raising the deviation to 20 centimetres. From time step 31, this is followed by a steep increase of the average distance to 1 metre. During this increase, the deviation decreases to 5 centimetres, and results at 15 centimetres at the end of it. After that in time steps 38 to 41, the average distance again decreases to below 20 centimetres, and the deviation decreases to 10 centimetres at the end. Further, the mean and the corresponding standard deviation in metre for each earlier described sector are shown in the following Table 5.6.

LOS mean$\pm\sigma$ in m	Mixed mean$\pm\sigma$ in m	NLOS mean$\pm\sigma$ in m
0.188 $\pm$0.083	0.308 $\pm$0.287	0.548 $\pm$0.352

Table 5.6: Weighted range: Mean and standard deviation of sectors

Weight on Old Estimate

First, the relative probabilities inside the grid get compared. The mean of average probability over all boxes is numerically shown in Table 5.7. The mean probability outside the robot box is significantly smaller than inside the robot box. The MaxBox square equal to the size of the robot box around the maximum likelihood in our grid estimate is still higher than that of the robot box, but with incorporated standard deviatio the robot box probability is closer to that of the MaxBox than to that of

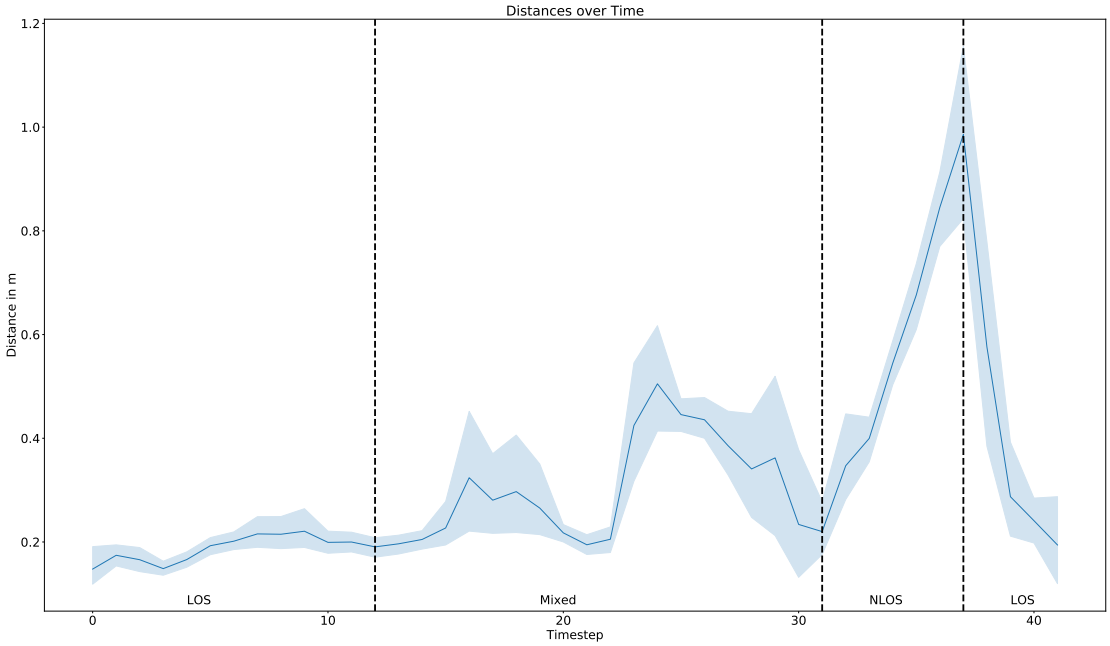

Figure 5.13: Weighted range: Absolute distance over time between estimate and real robot

RobotBox	OutsideBox	MaxBox
$9.10 \cdot 10^{-5}$	$6.45 \cdot 10^{-6}$	$2.37 \cdot 10^{-4}$

Table 5.7: Weighted old estimate: Average probabilities in parts of the grid

the OutsideBox. This means the probability inside the robot box is closer to the maximum likelihood than to the minimum likelihood of residence of the teammate. This shows that the average probability inside the robot box exceeds the threshold of at least 50% higher probability values calculated with Equation 5.3. Therefore, the teammate was found in every run and the distance was calculated with Equation 5.4. The absolute distances in metres over every experimental run are shown as box plots in Figure 5.14. Here, the interquartile ranges are mainly located in intervals below 50 centimetres, except for 1 out of 41 runs. The distribution in the interquartile ranges for every run is not symmetrical, but heavily skewed to the left. This means 25% of all values in the interquartile range are crowded in a significantly smaller range from the first quartile to the median, and values above the median are wider distributed. Therefore, more accurate distances gather inside smaller intervals and distances, that are more inaccurate, are located in wider intervals. The mean over all distance values yielded by all experimental runs is 31.72 centimetres and is shown as the horizontal line in the figure.

The former box plots incorporate values yielded over the whole run, including change from line of sight to non line of sight scenarios. Therefore, the experimental results get plotted over time. This is shown in Figure 5.15. Here, the calculated distance from all experimental runs is plotted over the corresponding measurement time step as an average distance with the 95% confidence interval. The time axis is separated into different sectors according to the line of sight scenario in these time steps. These sectors are line of sight until time step 12, mixed line and non line of sight between

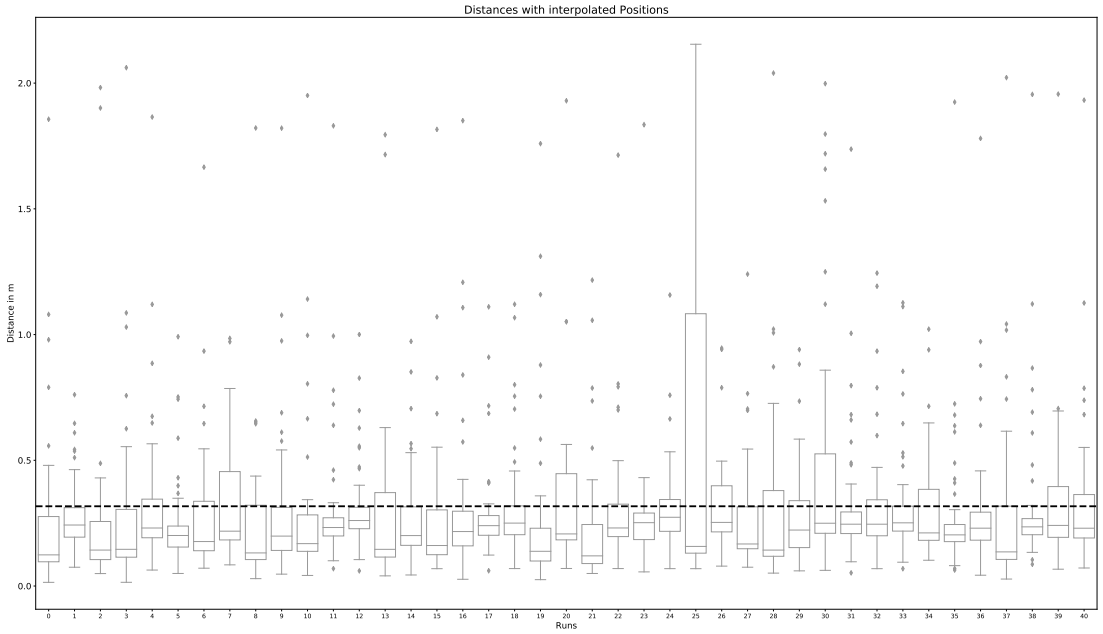

Figure 5.14: Weighted old estimate: Absolute distance between calculated position and real robot

time steps 12 and 31 and non line of sight after time step 31 until time step 38. In the first 17 time steps, an average distance of about 20 centimetres is achieved, including little to no deviation. Then, the average distance increases shortly to 30 centimetres with a deviation of about 10 centimetres, but immediately decreases again to 20 centimetres with minimal deviation. At time step 22, the average distance quickly increases to about 60 centimetres, and the corresponding deviation sees an increase to an overall deviation of 15 centimetres. Between time steps 25 and 31, a short decrease in average distance back to 20 centimetres can be observed, maintaining a deviation of 15 centimetres. This is followed by an increase in the average distance to 1 metre and an increase in deviation to 30 centimetres at the top. After that in time steps 38 to 41, the average distance again decreases to below 20 centimetres, but simultaneously decreasing the deviation to 5 to 10 centimetres. Further, the mean and the corresponding standard deviation in metre for each earlier described sector are shown in the following Table 5.8.

LOS mean$\pm\boldsymbol{\sigma}$ in m	Mixed mean$\pm\boldsymbol{\sigma}$ in m	NLOS mean$\pm\boldsymbol{\sigma}$ in m
0.177 $\pm$0.066	0.308 $\pm$0.314	0.576 $\pm$0.358

Table 5.8: Weighted old estimate: Mean and standard deviation of sectors

5.4.2 Evaluation

In the following, the results gathered before will be evaluated and conclusions about this will be drawn. At first, the probabilities of each box get compared and are therefore summarized and shown in Table 5.9. As you can see here, the robot box means over every run just differ slightly for each weighting because the changing line of sight scenarios are incorporated inside these values and therefore compensate

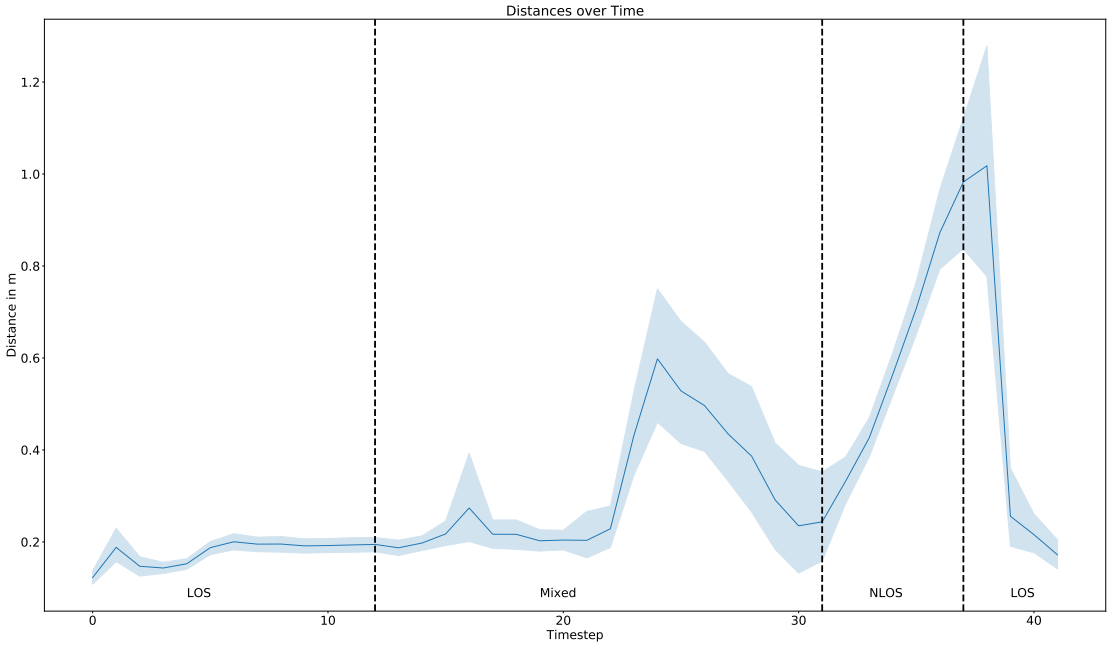

Figure 5.15: Weighted old estimate: Absolute distance over time between estimate and real robot

Weight on	RobotBox mean(10^{-5})	OutsideBox mean(10^{-6})	MaxBox mean(10^{-4})
Laser Points	8.28	6.47	2.39
Laser Area	9.12	6.51	2.27
Range	9.40	6.45	2.47
Old Estimate	9.10	6.45	2.37

Table 5.9: Influence of weighting on average probabilities in boxes

the expected strengths and weaknesses of each sensor input. Notable here is the average laser points probability inside the robot box, which is the lowest of all weighting experiments and therefore spans the biggest interval between the robot box probability and the maximum box probability. This increased gap might be caused by the measurement offset of the laser range detector between the edge and centre of an object, described in Section 4.2.2. It is more significant because of more weight on those affected measurements.

Further, it can be seen, that the yielded mean probabilities of the robot box for all weightings always exceed the threshold of 50% higher values than the probabilities outside the robot box and are closer to the maximum probability of the grid. Hence, for all weightings, the teammate was found, and the distances were calculated.
Table 5.10 shows the mean, the standard deviation and the maximum value of the 99% confidence interval of the yielded distance values grouped by line of sight scenarios. Here, the effects of every weighting of the inputs on the yielded distances can be seen and analysed.
The experiment with weights on the laser points yields best results in a line of sight scenario as expected, but falls off as soon as the line of sight is lost. It performs worst for the mixed and the non line of sight scenario of all experiments, and only achieves

Weight on	LOS		Mixed		NLOS	
	mean $\pm\sigma$ (m)	max 99%	mean $\pm\sigma$ (m)	max 99%	mean $\pm\sigma$ (m)	max 99%
Laser Points	0.195 $\pm$0.064	0.203 m	0.355 $\pm$0.374	0.392 m	0.619 $\pm$0.392	0.691 m
Laser Area	0.2 $\pm$0.074	0.208 m	0.328 $\pm$0.313	0.358 m	0.559 $\pm$0.376	0.625 m
Range	0.188 $\pm$0.083	0.197 m	0.308 $\pm$0.287	0.335 m	0.548 $\pm$0.352	0.607 m
Old Estimate	0.177 $\pm$0.066	0.184 m	0.308 $\pm$0.314	0.338 m	0.576 $\pm$0.358	0.634 m

Table 5.10: Influence of weighting on distances

a maximum value of 39.2 and 69.1 centimetres with 99% confidence. Contradictory to the expectation, it performs second worst in a line of sight scenario as well. Only the standard deviation in the line of sight scenario outperforms all other experiments, which indicates accurate results with a kind of offset error. Here, as mentioned before, the measurements are influenced by the offset of the sensor readings and yield increased distances. Therefore, this experiment will perform better in a line of sight environment than what can be shown with the results of the evaluation metric here up to half the size of tracked teammate, in this setup 10 centimetres.
The weight on laser area experiment yields the worst results in line of sight and is subject to the sensor reading offset as well. But instead of pinpointing the location of the teammate like the laser sensor points input, it only reduces the uncertainty of residence of the teammate on the whole grid, even if no line of sight on the teammate is given. Therefore, the distances yielded in the mixed and non line of sight scenario are more accurate than with weight on the laser points input. In a mixed environment, it performs only slightly worse than the range and old estimate experiment, and in non line of sight it yields even better mean and maximum values with 99% confidence than the old estimate experiment. Hence, the uncertainty reduction is of value over all scenarios by excluding possible locations and therefore increasing the certainty in other parts of the grid.
With weight on the range input, the yielded distances excel mainly in a non line of sight scenario as expected, with a 99% confidence maximum value of slightly above 60 centimetres. But also in mixed and line of sight, the results are best to second best among the experiments, with values yielded with 99% confidence of 33.5 centimetres at max in mixed and 19.7 centimetres at max in line of sight. The range sensor is accurate but is not able to locate the teammate on its own and therefore needs another sensor to locate it exactly. For line of sight this is done by the laser points input and for non line of sight the possible residence area is limited by the old estimate. Hence, with help of other inputs, the universally good range input can be accurate on all sight scenarios.
With weights on the old estimate input, overall good results are yielded. Here, the result from the previous time step get emphasized and so all other inputs are used with equal weight and the results are based on the measurements of those but with

a delay, which can be seen in the changes in distance measurements some time after the sight scenario changes in Figure 5.15. Therefore, it smooths pinpointed changes of the environment for more precision, which is given within the line of sight scenario, and gets worse if the location can not be pinpointed like in a non line of sight environment as the grid is blurred to a point of equal probability. The accuracy of the result in such a case scales with the size of the blur, which is linked to the velocity of change and the mentioned delay.
In summary, it is shown, that all inputs have different strengths and weaknesses are so susceptible to different environments like line of sight and non line of sight. To maximize the effectiveness of the fusion and each input, it is important to adapt different inputs and weightings to the current environment. For example, it is a good practice to emphasize the range input if more objects are present, which interrupt the line of sight. Hence, for each application, it is best to adjust the inputs to the needs of the corresponding environment. Therefore, the requirement of adaptability described in Section 5.1.4 was successfully verified.

5.5 Experiments simulating a fault on the sensors

This experiment is conducted similar to the experiment on Bayesian Fusion, but with simulated faults or errors on the sensor inputs, which are likely to happen in a real scenario. This is done by adding more Gaussian noise on each sensor measurement as a value-correlated error, completely neglect an input source for failure or overwrite random cells in the result with high values as non value-correlated error. For all failure scenarios, the failure occurrence probability and the related duration will be determined as described in Section 4.2.4. Hereby, the influence of these faults and errors on the robustness and accuracy of the result can be examined. And furthermore, it can be checked how big the faults can get while a feasible level of accuracy is still achieved. First, the impact of a neglect of input sources will be examined. Hereby, input sources will be selected to be neglected for every measurement time step. Second, a value-correlated error on the input sources will be generated for every time step as well. Third, a non value-correlated error will be generated on the inputs. The results can be compared against other types of faults and against the results of the experiment, which evaluates the overall accuracy of the fusion with no added faults.

5.5.1 Results

Missing Source

In the following, the results of the conducted experiment with missing sensor inputs are shown. First, the relative probabilities inside the grid get compared. This is shown in Figure 5.16. This figure contains box plots showing the average likelihood in the result grid on a log-scale over every experimental run. The left box plots show the average probabilities inside the robot box per run, and the dashed line marks the mean over all data. Here, the mean of the probabilities is $9.88 \cdot 10^{-5}$. Whereby, the associated standard deviation is $7.61 \cdot 10^{-5}$. The box plots in the middle show the average probability outside the pre-defined robot box per run, with a dashed line as mean. The overall mean has a value of $6.36 \cdot 10^{-6}$ and a standard

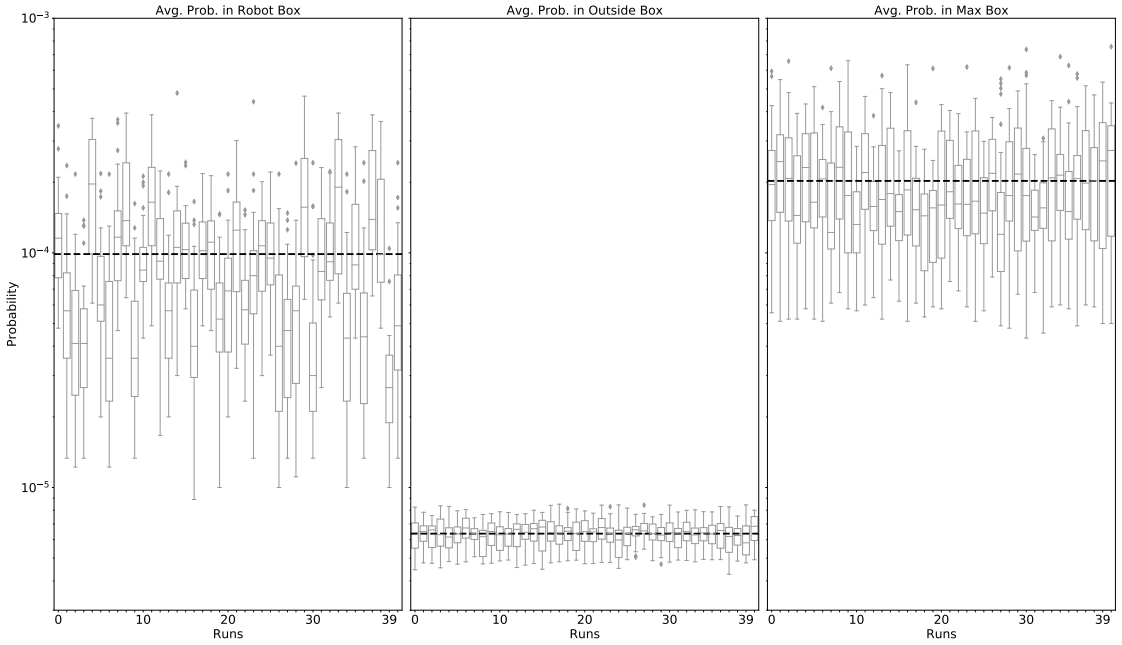

Figure 5.16: Missing Sensor: Average probability in robot box, outside robot box, and around max probability

deviation of $9.01 \cdot 10^{-7}$. The overall deviation and thus also the interquartile ranges are significantly smaller than inside the robot box and are located far below in a way that the yielded values do not overlap once. Here, the calculated mean over all average probabilities inside the robot box is over 15 times larger than the mean calculated outside the robot box. The box plot on the right shows the average probability in a square equal to the size of the robot box with the maximum average probability in our grid estimate. The average probabilities in the maximum box have a mean value of $2.03 \cdot 10^{-4}$ with an associated standard deviation of $1.19 \cdot 10^{-4}$. It can be seen, that the overall mean is still higher than that of the robot box, but the intervals of the yielded values are overlapping at some points. This means, the average probability inside the robot box is closer to the maximum likelihood than to the minimum likelihood of residence.

These plots show that the mean of average probabilities inside the robot box exceeds the threshold of at least 50% higher probability values than outside the robot box calculated with Equation 5.3. Therefore, the teammate was found in every run and the distance was calculated with Equation 5.4. The absolute distances in metres over every experimental run are shown as box plots in Figure 5.17. Here, only 13 out of 41 interquartile ranges are not fully located below 50 centimetres. The interquartile ranges of those runs differ in value and the third quartile can reach up to about 1.3 metres at maximum. The distribution in the interquartile ranges is not symmetrical for all runs, but skewed to the left. This means 25% of all values in the interquartile range are crowded in a significantly smaller range from the first quartile to the median, and values above the median are wider distributed. Therefore, more accurate distances gather inside smaller intervals and distances. Distances, that are more inaccurate, are located in wider intervals. The mean over all distance values

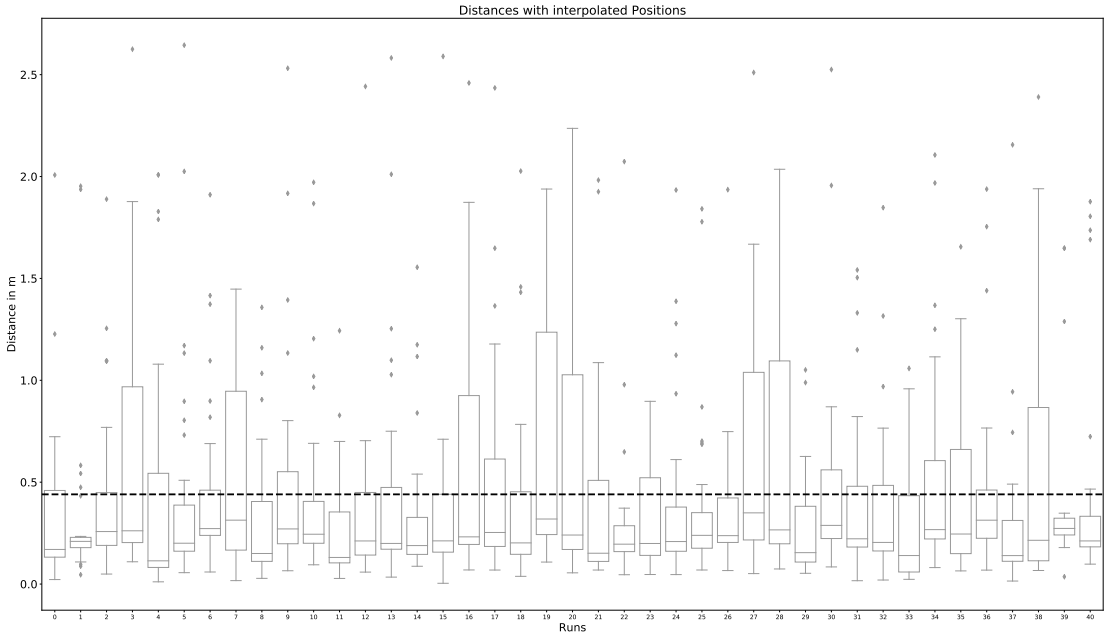

Figure 5.17: Missing Sensor: Absolute distance between calculated position and real robot

yielded by all experimental runs is 0.44 metres with a standard deviation of 0.494 metres overall. The mean is shown as the dashed horizontal line in the figure.

A latency measurement, shown in Figure 5.18, can be used to examine the correlation between the latency behaviour of the input sources and the distances per run and in this case shows the neglection of input sources through the Weibull distribution as described in Section 4.2.4. The standard deviations of latency range from 0 to about 2.4 seconds. Here, most values are located around in the interval from 0.5 to 1.5 seconds. Therefore, the latency is below a time step in this experiment, which is about 2 seconds. This means the failure occurrence as well as the failure duration on the inputs chosen by the Weibull distribution were successful. The failures were distributed among the different input sources. Hereby, inputs are alternating to fail or not, which indicates the failure of at least 1 input in every time step and hence the possibility of failure of multiple inputs at the same time step and most likely on run 18 with all input latencies above 2 seconds. Run 5, 12 and 15 are the ones to likely not contain multiple failures at the same time, because one or two inputs have a value of approximately 0. Here, run 18 with all input latency deviations above 2 seconds yields no significant increased distance values. Run 5, 12 and 15 yield distances with interquartile ranges below the overall mean, but are only slightly better than run 18.

Value-correlated Faults

In the following, the results of the conducted experiment with value-correlated faults are shown. First, the relative probabilities inside the grid get compared. This is shown in Figure 5.19. This figure contains box plots showing the average likelihood in the result grid on a log-scale over every experimental run. The left box plots show the average probabilities inside the robot box per run, and the dashed line marks

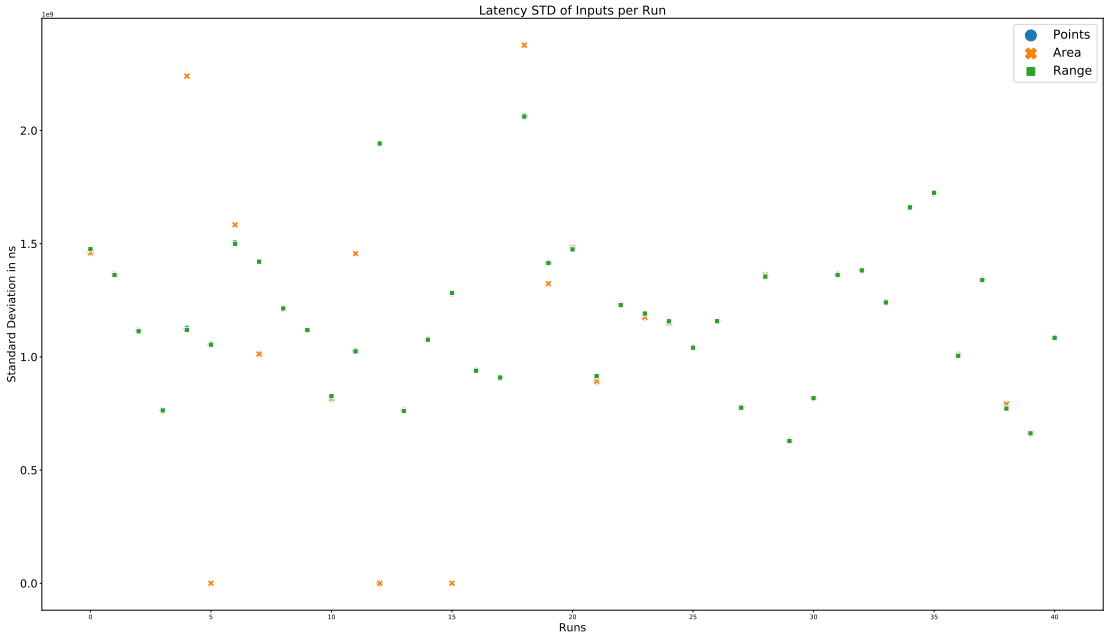

Figure 5.18: Missing Sensor: Standard Deviation of Latency per Run

the mean over all data. Here, the mean of the probabilities is $7.48 \cdot 10^{-5}$. Whereby, the associated standard deviation is $6.02 \cdot 10^{-5}$. The box plots in the middle show the average probability outside the pre-defined robot box per run, with a dashed line as mean. The overall mean has a value of $6.35 \cdot 10^{-6}$ and a standard deviation of $8.6 \cdot 10^{-7}$. The overall deviation and thus also the interquartile ranges are significantly smaller than inside the robot box and are located far below in a way that the yielded values do not overlap once. Here, the calculated mean over all average probabilities inside the robot box is over 11 times larger than the mean calculated outside the robot box. The box plot on the right shows the average probability in a square equal to the size of the robot box with the maximum average probability in our grid estimate. The average probabilities in the maximum box have a mean value of $1.86 \cdot 10^{-4}$ with an associated standard deviation of $1.04 \cdot 10^{-4}$. It can be seen, that the overall mean is still higher than that of the robot box, but the intervals of the yielded values are overlapping. This means, the average probability inside the robot box is closer to the maximum likelihood than to the minimum likelihood of residence.

These plots show that the mean of average probabilities inside the robot box exceeds the threshold of at least 50% higher probability values than outside the robot box calculated with Equation 5.3. Therefore, the teammate was found in every run and the distance was calculated with Equation 5.4. The absolute distances in metres over every experimental run are shown as box plots in Figure 5.20. Here, only 11 out of 41 runs have interquartile ranges ranging above 50 centimetres. The interquartile ranges of those runs differ in value, and the third quartile can reach up to about 1.4 metres at maximum. The distribution in the interquartile ranges is not symmetrical for all runs, but skewed to the left, except for the last conducted run, which is skewed to the right. This means for almost all runs, 25% of all values in the interquartile range are crowded in a significantly smaller range from the first quartile to the median, and

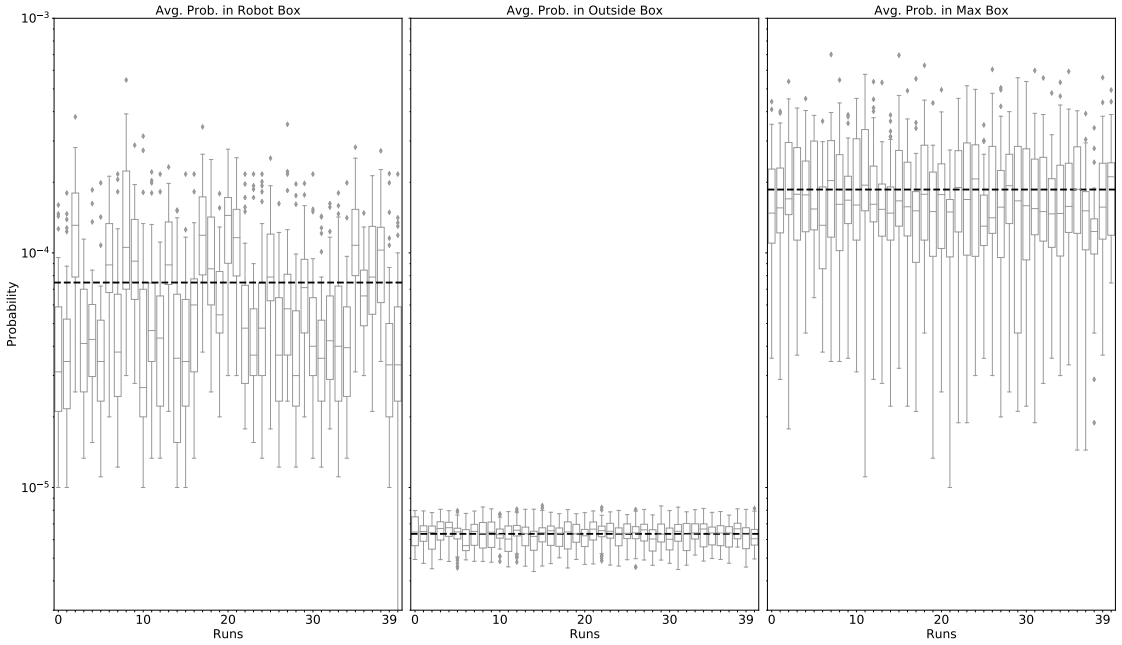

Figure 5.19: Value-correlated Fault: Average probability in robot box, outside robot box, and around max probability

values above the median are wider distributed. Therefore, more accurate distances gather inside smaller intervals and distances. Distances, that are more inaccurate, are located in wider intervals. The mean over all distance values yielded by all experimental runs is 0.426 metres with a standard deviation of 0.447 metres overall. The mean is shown as the dashed horizontal line in the figure.
The corresponding latency measurement per run, shown in Figure 5.21, can be used to examine correlations between the latency behaviour of the input sources and the distances per run. In this case, only small latencies should occur. The standard deviations of latencies range from mostly 0 to about 1 second, except for one outlier on run 23. Therefore, there are only small latencies to the input sources, which have no significant influence on the yielded distances, even for run 23 with the outlier.

Non Value-correlated Faults

In the following, the results of the conducted experiment with non value-correlated faults are shown. First, the relative probabilities inside the grid get compared. This is shown in Figure 5.22. This figure contains box plots showing the average likelihood in the result grid on a log-scale over every experimental run. The left box plots show the average probabilities inside the robot box per run, and the dashed line marks the mean over all data. Here, the mean of the probabilities is $9.68 \cdot 10^{-5}$. Whereby, the associated standard deviation is $7.99 \cdot 10^{-5}$. The box plots in the middle show the average probability outside the pre-defined robot box per run, with a dashed line as mean. The overall mean has a value of $6.39 \cdot 10^{-6}$ and a standard deviation of $9.58 \cdot 10^{-7}$. The overall deviation and thus also the interquartile ranges are significantly smaller than inside the robot box and are located far below in a way that the yielded values do not overlap once. Here, the calculated mean over all average probabilities inside the robot box is over 15 times larger than the mean

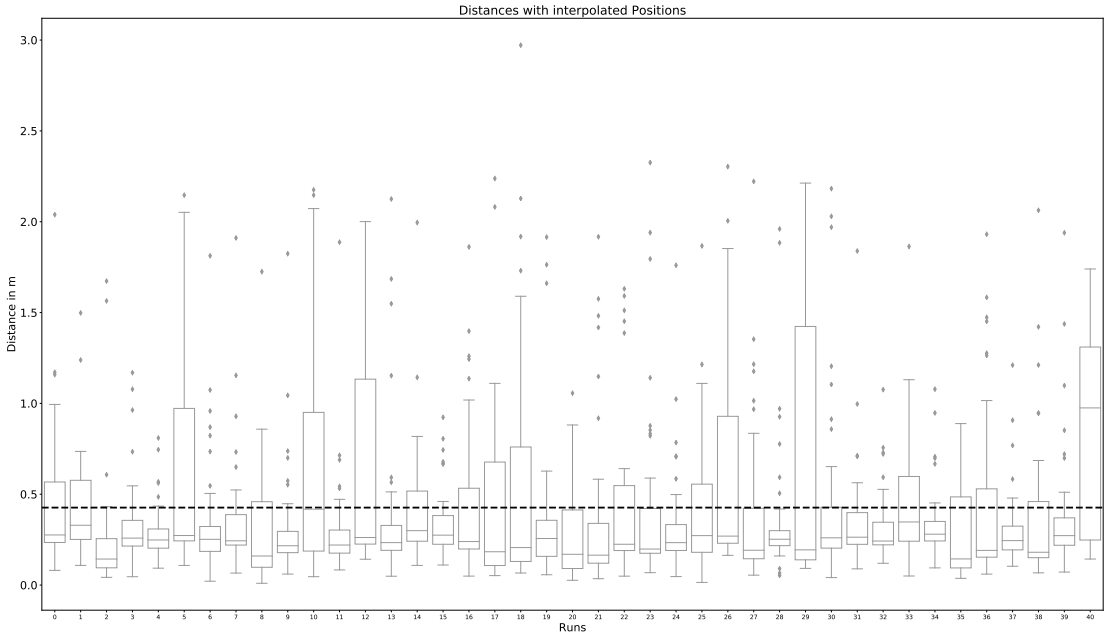

Figure 5.20: Value-correlated Fault: Absolute distance between calculated position and real robot

calculated outside the robot box. The box plot on the right shows the average probability in a square equal to the size of the robot box with the maximum average probability in our grid estimate. The average probabilities in the maximum box have a mean value of $2.36 \cdot 10^{-4}$ with an associated standard deviation of $1.04 \cdot 10^{-4}$. It can be seen, that the overall mean is still higher than that of the robot box, but the intervals of the yielded values are overlapping. This means, the average probability inside the robot box is closer to the maximum likelihood than to the minimum likelihood of residence.
These plots show that the mean of average probabilities inside the robot box exceeds the threshold of at least 50% higher probability values than outside the robot box calculated with Equation 5.3. Therefore, the teammate was found, and the distances were calculated for every run with Equation 5.4. The absolute distances in metres over every experimental run are shown as box plots in Figure 5.23. Here, only 5 out of 41 runs have interquartile ranges ranging above 50 centimetres. The interquartile ranges of those runs differ in value. 4 of those runs have the third quartile located below 1 metre, and only run 22 reaches up to 2.5 metres. The distribution in the interquartile ranges is not symmetrical for all runs, but heavily skewed to the left, except for run 22 and 36, which are skewed to the right. This means for almost all runs, 25% of all values in the interquartile range are crowded in a significantly smaller range from the first quartile to the median, and values above the median are wider distributed. Therefore, more accurate distances gather inside smaller intervals and distances. Distances, that are more inaccurate, are located in wider intervals. The mean over all distance values yielded by all experimental runs is 0.371 metres with a standard deviation of 0.442 metres overall. The mean is shown as the dashed horizontal line in the figure.
The corresponding latency measurement per run, shown in Figure 5.24, can be used to examine correlations between the latency behaviour of the input sources and the distances per run. In this case, only small latencies resulting from some sort of errors

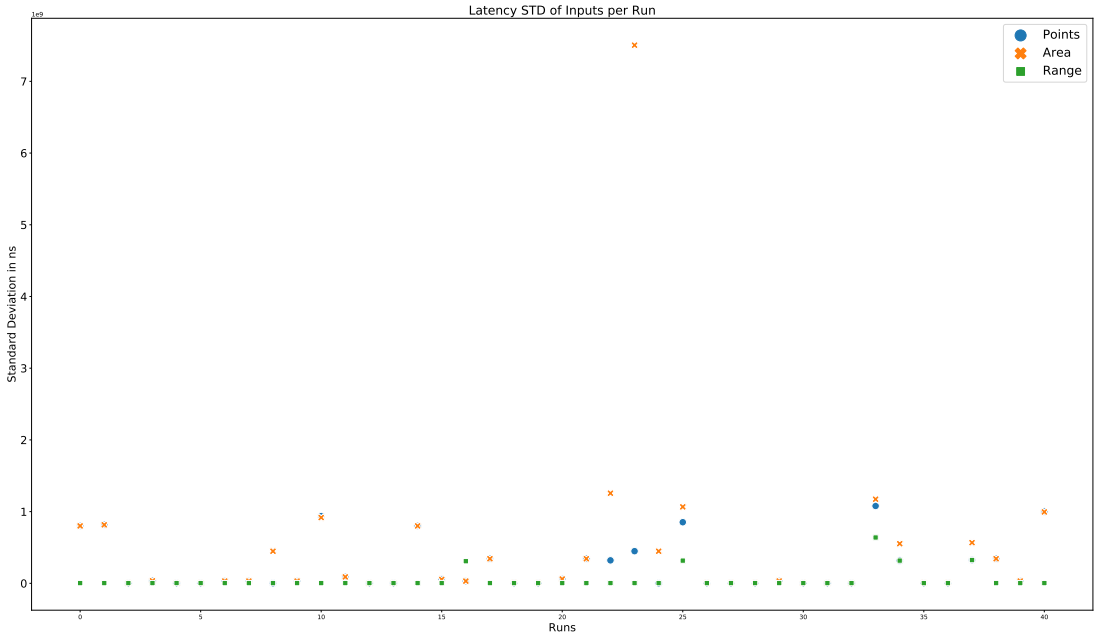

Figure 5.21: Value-correlated Fault: Standard Deviation of Latency per Run

should occur, as well. The standard deviations of latencies range from mostly 0 to about 1 second, except for one outlier on run 1, 2, 3 and 6. Therefore, there are mainly small latencies to the input sources, which have no significant or visible influence on the yielded distances, even for the outlier runs.

5.5.2 Evaluation

In the following, the gathered results will be evaluated and conclusions about this will be drawn. At first, the probabilities of each box get compared and are therefore summarized and shown in Table 5.11. Here, for each fault type the mean of the

Fault Type	RobotBox mean(10^{-5})	OutsideBox mean(10^{-6})	MaxBox mean(10^{-4})
Missing	9.88	6.36	2.03
Value-correlated	7.48	6.35	1.86
Non Value-correlated	9.68	6.39	2.36

Table 5.11: Influence of fault types on average probabilities in boxes

average probabilities inside the robot box, outside the robot box and inside the maximum box is shown. It can be seen, that the mean outside the robot box is almost the same for every fault type. Though, the mean inside the robot box and the maximum box differ significantly in value for each fault type. Value-correlated faults yield the lowest mean value inside the robot box and also inside the correlated maximum box. With a difference of over $2 \cdot 10^{-5}$ to the lowest, the mean inside the robot box yielded by non value-correlated faults is the second highest of the 3 fault types, but has the highest mean inside the maximum box of them. The mean inside the robot box yielded while inputs are missing has the highest value of all fault

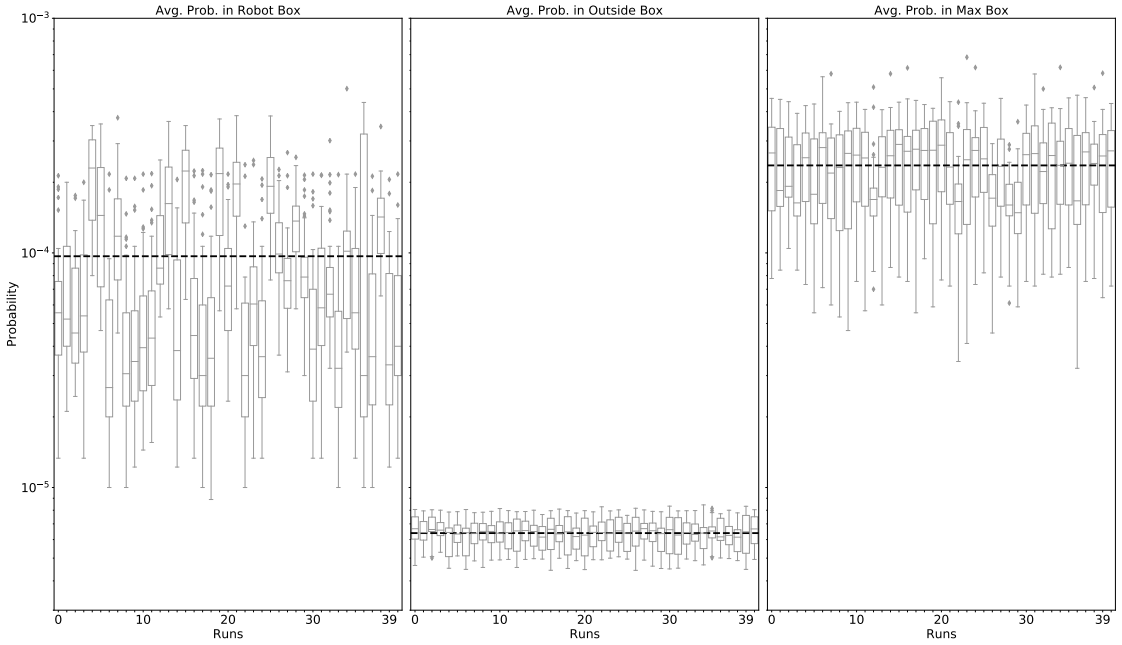

Figure 5.22: Non Value-correlated Fault: Average probability in robot box, outside robot box, and around max probability

types. This shows that despite each type of fault, the probabilities inside the grid still have a significant gradient. Even the smallest mean value inside the robot box yielded by the value-correlated fault type is 11 times higher than the mean of average probabilities outside of it. Therefore, the uncertainty inside the whole certainty grid is greatly reduced for all tested fault types.
With all values exceeding the threshold of 50%, the absolute distances were calculated and summarized in Table 5.12. The mean distance value yielded while missing some

Fault Type	mean (m)	$\pm\sigma$ (m)
Missing	0.44	0.494
Value-correlated	0.426	0.447
Non Value-correlated	0.371	0.442

Table 5.12: Influence of fault types on average distances

inputs is the highest value among the 3 fault types, with 44 centimetres. The associated standard deviation is also the highest among them. Also, the highest number of runs have an interquartile range exceeding a distance of 0.5 metres. Those values can be explained through the distribution of failure shown in the results section. By neglecting an input, the result need to be calculated using the remaining inputs, even if the most accurate and informative input at that time step is missing. Further, as the distribution shows, it is possible for two or more sensors to fail during the same time step, leaving one or even zero inputs for the calculation of the result. This counteracts the benefits gained by a fusion of different sensors and renders the result more inaccurate with each missing input. The remaining inputs still locate the teammate inside the grid, resulting in a high mean of average probability values, but lack accuracy. Therefore, the missing of an input source has the most impact on

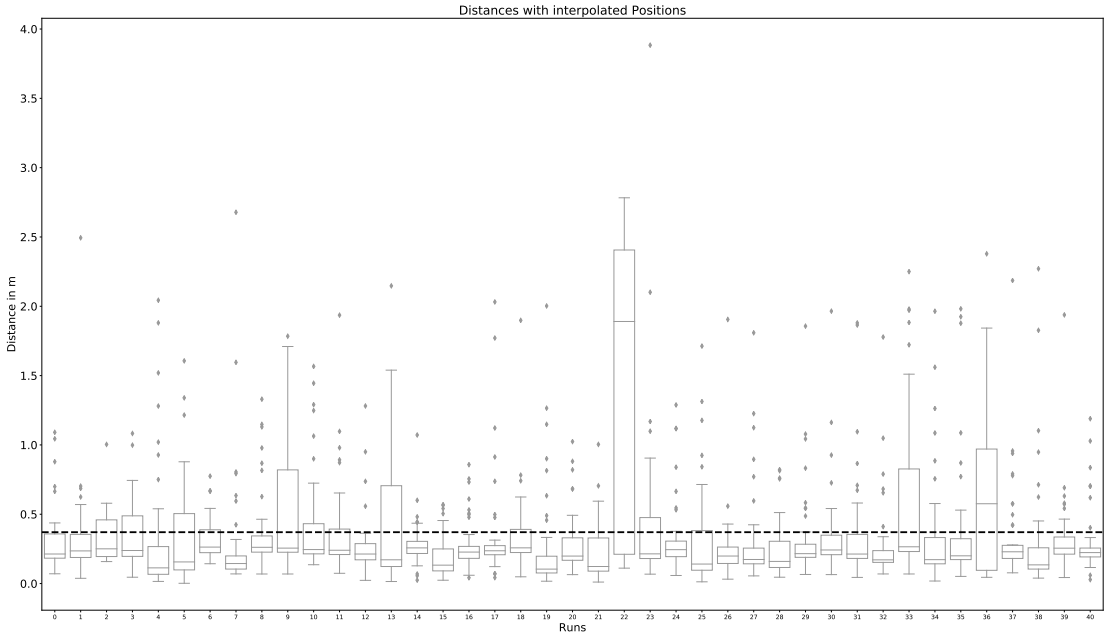

Figure 5.23: Non Value-correlated Fault: Absolute distance between calculated position and real robot

the result accuracy out of the 3 fault types combining low accuracy, with still high confidence in knowledge of residence.
Value-correlated faults yield the second-best mean distance of 42.6 centimetres and associated standard deviation of 44.7 centimetres. Here, the faults imposed on the different input sources impact the result not quite as much as missing inputs. This can be explained as follows. The value-correlated faults impose quite high errors on the inputs, but the fusion of multiple inputs, dampens the effects of those errors. The benefits gained from a sensor fusion are still applicable, and thus the mean distance is more accurate despite high errors than with missing inputs. The low mean of average probabilities inside the robot box results from the application of the fault. With the increase in the Gaussian error distribution, more cells in the certainty grid inherit higher probability values, spreading the probability above the grid. As the error increases, more cells of the grid get affected and the probabilities in each cell shrink, resulting in a lower mean of average probabilities inside the robot box. Hence, value-correlated faults decreases the accuracy of the yielded result, while simultaneously reducing the confidence in knowledge of the residence of the teammate, which is of advantage over a high confidence with bad accuracy.
The best mean distance value of 37.1 centimetres is yielded with non value-correlated faults. The associated standard deviation of 44.2 centimetres is also the best of all fault types. This fault type yields a significant better value than the other types, with a difference of over 5 centimetres in mean distance value, while also maintaining a high mean average probability value inside the robot box. This means non value-correlated faults decrease the accuracy the least among all tested faults and contain a high confidence in knowledge of the teammate's residence. The benefits of the sensor fusion can be seen here. The error, that was added on the inputs, have only low influence on the yielded result because for every fused input, the influence of this error is reduced to no significance and the non-faulty measurements get emphasized.

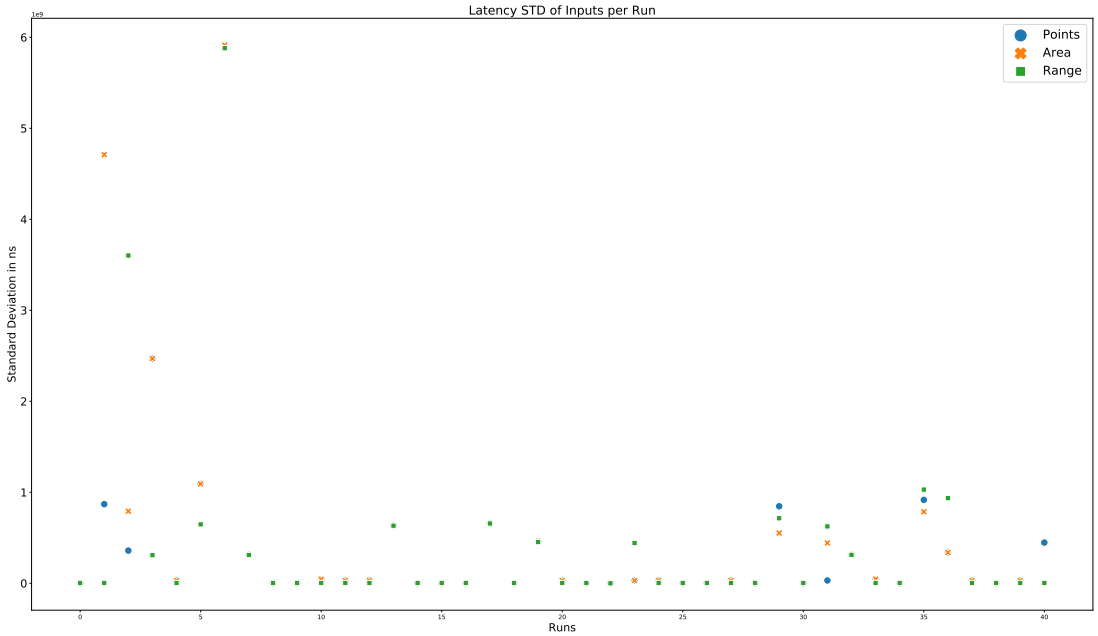

Figure 5.24: Non Value-correlated Fault: Standard Deviation of Latency per Run

Outlying distance values for all fault types in this experiment, may be due to other failures in the simulation as well as through localization errors of the robot leading to failures in the measurements. Unusual high latency values are possibly due to an input missing for multiple sequential time steps.
In summary, this experiment clearly shows, that faults have different influence on the yielded results as described above. Every fault type yields values worse than non-faulty ones from the overall accuracy experiment, with 10.2 centimetres to 3.3 centimetres difference in mean distance. Nevertheless, the mean distances of the fault experiments are still kept in a feasible range, and for each applied fault type, a significant gradient of probabilities in the certainty grid existed. Therefore, it can be seen, that the fusion approach is to a certain degree robust against errors. Hereby, the susceptibility to faults varies from fault type to fault type, with being most prone to the complete failure of inputs and least prone to non value-correlated faults.

5.6 Summary

The conducted experiments were done to examine the performance of the sensor fusion approach using the Bayes Theorem.
In this context, first, an experiment was done investigating the precondition of stochastic independence for using the Bayes Theorem. Here, all input sources are linear independent to each other and additionally one can be clearly seen as stochastic independent. The other two do not fulfil the requirement for stochastic independence. Nevertheless, this supports the usage of naive Bayes assumptions for the use with the Bayes theorem because in a robotic use case mutual deviation from stochastic independence are not preventable and are likely not to largely interfere with the results.
Second, further experiments were conducted to provide insight on the performance and specifically on, whether the requirements for successful teammate tracking, established

for this use case in Section 5.1.4, can be fulfilled. The accuracy requirement of 30 centimetres, with a deviation of 15 centimetres, is evaluated in experiment Section 5.3. Overall, the requirement can be seen as fulfilled. In line of sight scenarios, a significantly lower distance and standard deviation is measured. Combined, with the whole experimental run, which is dependent on used inputs and environment, a slightly higher distance mean of 33.8 centimetres is measured. But, it is shown in this experiment, that by combining inputs using this approach the accuracy can be highly improved and by improving some experimental designs the mean distance would exceed the requirement.
The requirement of adaptability to different use case can also be seen as fulfilled. Here, different weightings on different input sources, shift the yielded values in favour of certain environmental properties. Optical sensors favour more line of sight scenarios, whereas ultra-wideband sensors are more versatile in obstacle rich environments, and additionally emphasizing the results from old time steps smooth the current result in a dynamic environment. It is clearly shown by the conducted experiment in Section 5.4 that weighting on the inputs adapt the approach to different use cases.
The last requirement of robustness is evaluated in experiment Section 5.5 by the application of different faults. It is shown that, despite different types of faults, a gradient in certainty and thus a reduction in uncertainty exists for all tested fault types. The overall measured accuracy is decreased by applying different faults, but the value differs for each type of fault. The missing of inputs yields the highest decrease, value-correlated faults have a medium decrease on accuracy but the highest decrease in certainty, and non value-correlated faults yield values with the lowest decrease in accuracy and are well compensated by the fusion.
In summary, the Bayes-Hempel Method achieves feasible results and is able to answer the research questions. First, question RQ1 about a meaningful evaluation metric to measure the result quality gets answered. Here, the metric consists of a relative and an absolute part. The relative part evaluates the certainty in knowledge about the position of the teammate. The absolute part then evaluates the distance error between the calculated position and the actual one of the teammate. Hence, accuracy and certainty in this accuracy gets combined in this metric. The metric suited to evaluate the results, but it is not perfect as described in experiment 5.3 and can be further developed to incorporate the effects of sensor properties, which leads to even better evaluation results. Second, question RQ3 regarding the influence of parameters is answered. The question is reflected by the requirement of adaptability. Therefore, it can be said, that changing parameters greatly influences the approach and increases the usability of the Bayes-Hempel Method in different environmental settings. Third, question RQ4 concerning the robustness is also answered with the requirement analysis. Hence, with usual occurring faults, the certainty, and the accuracy is maintained to a certain degree. Hereby, it being most prone to the failure of inputs, and least prone to non-value correlated faults. With the analysis of the requirements and the answered questions, the overall question RQ2 classifying the quantitative advantage over other state-of-the-art fusion approaches can be answered. Though, no significant improvement in performance could be achieved, the Bayes-Hempel Method is able to perform similarly to other approaches but with certain benefits. On best conditions, the Bayes-Hempel Method yields translation errors of around 0.2 metres. This can compete with results yielded by Kalman sensor fusions

of 0.05 and respectively 0.14 metres, and are on an equal level with values yielded by a deep learning fusion approach. Both other approaches are formerly described in Section 3.2. The mentioned benefits of the Bayes-Hempel Method consist of a fast adaptability to environmental changes, the ability to maintain spatial orientations for each teammate during changes by mapping each teammate in a single certainty grid, and an enhancement of the ability to combine sensors of different modalities.

The next chapter gives an overlook over this thesis and concludes the most important findings and results. Further, an outlook on possible future work and improvements is given.

6. Conclusion and Future Work

In this work, the question is answered, whether the application of a Bayesian sensor fusion with modified certainty grids brings qualitative advantages and improvements compared to latest works on other sensor fusion approaches. These approaches are used for teammate tracking and localization in a robotic swarm environment.

After the clarification of the basics, the implementation of the introduced approach is described. The modifications to basic certainty grids are presented. The grid represents a discrete density function, which is used to estimate the position of a teammate. The processing of the grid data is done by the Bayes-Hempel Method, which utilizes the Bayes theorem to combine different MCGs produced by several sensors. To validate this approach, different experiments are conducted. With those, the quality of measurements, the robustness, and the adaptivity to different scenarios considering preset requirements are investigated.

In summary, the approach yields feasible results satisfying the requirements and the Research Questions 1, 2, 3 and 4 can be answered. On best conditions, a mean distance error of 20 centimetres far below the requirement of 30 centimetres and on average a mean distance error of 33.8 centimetres is achieved. This mainly depends on the used sensors and the properties of the environment. With parameter changes, an adaptability to this different environment properties and therefore to different use cases can be achieved. In contrast to other state-of-the-art fusion approaches like Kalman or deep learning, no significant improvement in performance, but similar results, could be achieved. Nevertheless, the presented approach achieves a faster adaptability to environmental changes and adds the ability to maintain spatial orientations for each teammate during changes by mapping each teammate in a single certainty grid. Furthermore, our sensor fusion enhances the ability to combine sensors of different modalities.

Future Work

During the evaluation, different parts with potential for improvement were found.

We already introduce a compensation of data delays to get more accurate results. Similar approaches could be used for other calculable inaccuracies, for example the

deviation between the real robot position and the measured outer boundaries. Those occur because the LIDAR sensor only senses these outer boundaries. As in swarms usually only one type of robot is used, information like the size and form of the robots are known and could be used to eliminate this offset.

Additionally, in swarms, most of the time more than 2 robots are used. Thus, experiments with more than 2 robots would be beneficial as an extension of this work.

Furthermore, a more accurate movement prediction of the teammate could be added to concretize the probability distribution of the possible teammate position. By analysing multiple old estimates yielded over a longer time, a movement path can be reconstructed and extrapolated.

For all the experiments we assumed, that the robot know their position. In real-world scenarios this is not often the case and different localization algorithms would be used. This applies an additional error to the teammate tracking as it depends on the own position. Further experiments on the interplay between localization and our teammate tracking approach can provide insight for more realistic use cases.

In these use cases containing different environments, the Bayes-Hempel method achieves good results, but the parameters need to be tuned manually. By using an optimization algorithm, that adapts the parameters to the current environment, a better usability and probably an increased quality of results would be achieved. In relation to this, a sensitivity analysis on sensor weightings can be conducted to gain further insights into dependencies.

Bibliography

[1] S. Ghorpade, M. Zennaro, and B. Chaudhari, "Survey of localization for internet of things nodes: approaches, challenges and open issues," *Future Internet*, vol. 13, no. 8, p. 210, 2021.

[2] B. Vergouw, H. Nagel, G. Bondt, and B. Custers, "Drone technology: Types, payloads, applications, frequency spectrum issues and future developments," in *The future of drone use*, pp. 21–45, Springer, 2016.

[3] J. N. Yasin, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Formation maintenance and collision avoidance in a swarm of drones," in *Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control*, pp. 1–6, 2019.

[4] S. Hackwood and G. Beni, "Self-organization of sensors for swarm intelligence," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 819–820, IEEE Computer Society, 1992.

[5] W. Elmenreich, "An introduction to sensor fusion," *Vienna University of Technology, Austria*, vol. 502, pp. 1–28, 2002.

[6] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830–39846, 2020.

[7] P. Cano and J. Ruiz-del Solar, "Robust tracking of soccer robots using random finite sets," *IEEE Intelligent Systems*, vol. 32, no. 6, pp. 22–29, 2017.

[8] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, *et al.*, "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots*, vol. 17, no. 2, pp. 223–245, 2004.

[9] D. Marco, T. Elio, G. Roderich, *et al.*, "The swarm-bot project," *Swarm Robotics. Lecture Notes in Computer Science*, vol. 3342, 2005.

[10] T. L. Anderson and M. Donath, "Synthesis of reflexive behavior for a mobile robot based upon a stimulus-response paradigm," in *Mobile Robots III*, vol. 1007, pp. 198–211, SPIE, 1989.

[11] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," in *Sensor Devices and Systems for Robotics* (A. Casals, ed.), (Berlin, Heidelberg), pp. 253–276, Springer Berlin Heidelberg, 1989.

[12] F. White, "Data fusion lexicon," 1991.

[13] K. Nagla, M. Uddin, and D. Singh, "Multisensor data fusion and integration for mobile robots: A review," *IAES International Journal of Robotics and Automation*, vol. 3, no. 2, p. 131, 2014.

[14] F. Castanedo, "A review of data fusion techniques," *The scientific world journal*, vol. 2013, 2013.

[15] M. Kam, X. Zhu, and P. Kalata, "Sensor fusion for mobile robot navigation," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 108–119, 1997.

[16] H. F. Durrant-Whyte, "Sensor models and multisensor integration," in *Autonomous robot vehicles*, pp. 73–89, Springer, 1990.

[17] R. C. Luo and M. G. Kay, "Multisensor integration and fusion in intelligent systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 901–931, 1989.

[18] A. Visser and F. Groen, "Organisation and design of autonomous systems. textbook, faculty of mathematics," *Computer Science, Physics and Astronomy, University of Amsterdam, Kruislaan*, vol. 403, 1999.

[19] R. R. Brooks and S. S. Iyengar, *Multi-Sensor Fusion: Fundamentals and Applications with Software*. USA: Prentice-Hall, Inc., 1998.

[20] J. Joyce, "Bayes' theorem," 2003.

[21] A. Gut and A. Gut, *Probability: a graduate course*, vol. 200. Springer, 2005.

[22] D. A. Schum, *The evidential foundations of probabilistic reasoning*. Northwestern University Press, 2001.

[23] M. P. Allen, "Covariance and linear independence," *Understanding regression analysis*, pp. 31–35, 1997.

[24] R. D'AGOSTINO and E. S. Pearson, "Tests for departure from normality. empirical results for the distributions of b 2 and b," *Biometrika*, vol. 60, no. 3, pp. 613–622, 1973.

[25] R. B. d'Agostino, "An omnibus test of normality for moderate and large size samples," *Biometrika*, vol. 58, no. 2, pp. 341–348, 1971.

[26] N. Henze and B. Zirkler, "A class of invariant consistent tests for multivariate normality," *Communications in statistics-Theory and Methods*, vol. 19, no. 10, pp. 3595–3617, 1990.

[27] E. L. Melnick and A. Tenenbein, "Misspecifications of the normal distribution," *The American Statistician*, vol. 36, no. 4, pp. 372–373, 1982.

[28] D. J. Hand and K. Yu, "Idiot's bayes—not so stupid after all?," *International statistical review*, vol. 69, no. 3, pp. 385–398, 2001.

[29] M. Hempel, "Programming and evaluation of an ultra-wideband distance measurement system for mobile robots," *University of Magdeburg, Magdeburg*, 2019.

[30] S. I. Popoola, A. A. Atayero, O. D. Arausi, and V. O. Matthews, "Path loss dataset for modeling radio wave propagation in smart campus environment," *Data in brief*, vol. 17, pp. 1062–1073, 2018.

[31] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using rssi value for distance estimation in wireless sensor networks based on zigbee," in *2008 15th international conference on systems, signals and image processing*, pp. 303–306, IEEE, 2008.

[32] L. Pei, R. Chen, J. Liu, H. Kuusniemi, T. Tenhunen, and Y. Chen, "Using inquiry-based bluetooth rssi probability distributions for indoor positioning," *Journal of Global Positioning Systems*, vol. 9, no. 2, pp. 122–130, 2010.

[33] C.-B. Lim, S.-H. Kang, H.-H. Cho, S.-W. Park, and J.-G. Park, "An enhanced indoor localization algorithm based on ieee 802.11 wlan using rssi and multiple parameters," in *Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on*, pp. 238–242, IEEE, 2010.

[34] K. Kaemarungsi, "Distribution of wlan received signal strength indication for indoor location determination," in *Wireless Pervasive Computing, 2006 1st International Symposium on*, pp. 6–pp, IEEE, 2006.

[35] Q. Fu and G. Retscher, "Active rfid trilateration and location fingerprinting based on rssi for pedestrian navigation," *The Journal of Navigation*, vol. 62, no. 2, pp. 323–340, 2009.

[36] "Digikey - at86rf233-zur." https://www.digikey.com/product-detail/en/microchip-technology/AT86RF233-ZUR/AT86RF233-ZURCT-ND/3775085. Accessed: 19.01.2019.

[37] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 41–52, 2007.

[38] E. Elnahrawy, X. Li, and R. P. Martin, "The limits of localization using signal strength: A comparative study," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 406–414, IEEE, 2004.

[39] S. Stavrou and S. Saunders, "Factors influencing outdoor to indoor radio wave propagation," 2003.

[40] R. Paschotta, "phase shift method for distance measurements." https://www.rp-photonics.com/phase_shift_method_for_distance_measurements.html. Accessed: 17.01.2019.

[41] L. Qiu, Z. Huang, S. Zhang, C. Jing, H. Li, and S. Li, "Multifrequency phase difference of arrival range measurement: principle, implementation, and evaluation," *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, p. 715307, 2015.

[42] "Speed of light." https://en.wikipedia.org/wiki/Speed_of_light. Accessed: 21.01.2019.

[43] F. Michler, H. Deniz, F. Lurz, R. Weigel, and A. Koelpin, "Performance analysis of an ultra wideband transceiver for real-time localization," in *2018 48th European Microwave Conference (EuMC)*, pp. 1141–1144, IEEE, 2018.

[44] R. Cardinali, L. De Nardis, M.-G. Di Benedetto, and P. Lombardo, "Uwb ranging accuracy in high-and low-data-rate applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 4, pp. 1865–1875, 2006.

[45] "Future electronics - decawave dwm1000." https://www.futureelectronics.com/p/wireless-rf--rf-modules-solutions--proprietary/dwm1000-decawave-7090231. Accessed: 19.01.2019.

[46] T. Raj, F. Hanim Hashim, A. Baseri Huddin, M. F. Ibrahim, and A. Hussain, "A survey on lidar scanning mechanisms," *Electronics*, vol. 9, no. 5, p. 741, 2020.

[47] Hitachi-LG Data Storage, *LDS1.5 Specifications Model HLS-LFCD2*, 2 2014.

[48] R. Mautz, "Overview of current indoor positioning systems," *Geodezija ir kartografija*, vol. 35, no. 1, pp. 18–22, 2009.

[49] W. H. Strickland and R. H. King, *Characteristics of ultrasonic ranging sensors in an underground environment.* US Department of the Interior, Bureau of Mines, 1993.

[50] J. D. N. Cheeke, *Fundamentals and applications of ultrasonic waves.* CRC press, 2016.

[51] S. Mai, "Wireless ranging in swarm robotics," 2015/2016.

[52] "Robotshop - ultrasonic range finders." https://www.robotshop.com/en/ultrasonic-range-finders.html. Accessed: 19.01.2019.

[53] "Migatroncorp - applications of ultrasonic sensors." https://www.migatron.com/ultrasonic-detections-and-control-applications/. Accessed: 04.01.2019.

[54] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[55] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," 1961.

[56] S. Thrun, W. Burgard, and D. Fox, "Probalistic robotics," *Kybernetes*, 2006.

[57] V. Sarma and S. Raju, "Multisensor data fusion and decision support for airborne target identification," *IEEE Transactions on systems, man, and cybernetics*, vol. 21, no. 5, pp. 1224–1230, 1991.

[58] R. E. Kalman, "When is a linear control system optimal?," 1964.

[59] G. Bishop, G. Welch, *et al.*, "An introduction to the kalman filter," *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-23175, p. 41, 2001.

[60] A. Karol and M.-A. Williams, "Distributed sensor fusion for object tracking," in *Robot Soccer World Cup*, pp. 504–511, Springer, 2005.

[61] J. A. Corrales, F. A. Candelas, and F. Torres, "Hybrid tracking of human operators using imu/uwb data fusion by a kalman filter," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 193–200, IEEE, 2008.

[62] K. Marzullo, "Tolerating failures of continuous-valued sensors," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 4, pp. 284–304, 1990.

[63] P. Chew and K. Marzullo, "Masking failures of multidimensional sensors," tech. rep., CORNELL UNIV ITHACA NY DEPT OF COMPUTER SCIENCE, 1991.

[64] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *ICML*, 2011.

[65] S. Bohez, T. Verbelen, E. De Coninck, B. Vankeirsbilck, P. Simoens, and B. Dhoedt, "Sensor fusion for robot control through deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2365–2370, 2017.

[66] "Robotis turtlebot3." https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/. Accessed: 30.06.2022.

[67] "Robot operating system." https://www.ros.org/. Accessed: 30.06.2022.

[68] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[69] "python.org." https://www.python.org/.

[70] "Gazebo sim." https://gazebosim.org/home. Accessed: 30.06.2022.

[71] "rosbag ros package." http://wiki.ros.org/rosbag.

[72] DecaWave Ltd, *DW1000 User Manual*, 6 2015. Revision 2.05.

[73] DecaWave Ltd, *DWM1000 Datasheet*, 6 2016. Revision V1.4.

[74] S. Shafer, A. Stentz, and C. Thorpe, "An architecture for sensor fusion in a mobile robot," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2002–2011, 1986.

[75] "Occupancy grid ros message." http://docs.ros.org/en/lunar/api/nav_msgs/html/msg/OccupancyGrid.html.

[76] J. Lienig and H. Bruemmer, *Fundamentals of electronic systems design*. Springer, 2017.

[77] G. S. Mudholkar and D. K. Srivastava, "Exponentiated weibull family for analyzing bathtub failure-rate data," *IEEE transactions on reliability*, vol. 42, no. 2, pp. 299–302, 1993.

[78] D. Weikert, C. Steup, and S. Mostaghim, "Availability-aware multiobjective task allocation algorithm for internet of things networks," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 12945–12953, 2022.

[79] S. Zug, A. Dietrich, and J. Kaiser, "An architecture for a dependable distributed sensor system," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, pp. 408–419, Feb 2011.

[80] S. Macenski and I. Jambrecic, "Slam toolbox: Slam for the dynamic world," *Journal of Open Source Software*, vol. 6, no. 61, p. 2783, 2021.

I herewith assure that I wrote the present thesis independently, that the thesis has not been partially or fully submitted as graded academic work and that I have used no other means than the ones indicated. I have indicated all parts of the work in which sources are used according to their wording or to their meaning.

Magdeburg, October 21, 2022