CHALMERS
UNIVERSITY OF TECHNOLOGY

Driver Intention Prediction for Optimal Vehicle Wake-up

A target tracking approach for driver intention prediction using UWB radar technology

Master's thesis in Systems, Control and Mechatronics

DANIEL HÅKANSSON
JOHAN MARTINSSON

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis 2020

Driver Intention Prediction for Optimal Vehicle Wake-up

A target tracking approach for driver intention prediction using UWB radar technology

Daniel Håkansson
Johan Martinsson

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
Gothenburg, Sweden 2020

Driver Intention Prediction for Optimal Vehicle Wake-up
A target tracking approach for driver intention prediction
using UWB radar technology
Daniel Håkansson
Johan Martinsson

Supervisors: Joel Lind, ART Power and Energy, Volvo Cars
Roman Sokolovskii, Department of Electrical Engineering
Examiner: Erik Agrell, Department of Electrical Engineering

Master's Thesis 2020
Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An illustration of the concept for remote vehicle wake-up [1].

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2020

Driver Intention Prediction for Optimal Vehicle Wake-up
A target tracking approach for driver intention prediction
using UWB radar technology
Daniel Håkansson
Johan Martinsson
Department of Electrical Engineering
Chalmers University of Technology

Abstract

With an increasing amount of advanced technology in the car industry, the development of new features that improve user experience is essential in order to stand out against competitors. A team at Volvo Cars has a desire to make their cars adapt to the driver automatically before use and be ready for immediate use with a minimal amount of manual inputs required by the driver. In this thesis, a prototype system is developed where the intention of the driver is predicted by tracking the trajectory of the key assumed to be carried by the driver to activate certain start-up functions in the car in an optimal way in terms of energy consumption. To track the movement of the key, UWB-technology has been used together with an Unscented Kalman Filter to create position, velocity and heading estimates relative to the car. This information is then used together with a loss function that quantifies the intention of the driver and enables to determine which start-up functions that should be activated. A function concept enabling the proposed system to remember the outcome of the trajectories, being whether the driver used the car or not, is also introduced. This concept is discussed to model the usage patterns of the car to predict future driver intentions as the car adapts to the driver over time. The results presented in this thesis are used to evaluate how well this prototype can estimate the trajectory, avoid false positives/negatives — where the vehicle misinterprets the intention of the driver — and also to compare the power consumption of the prototype towards other speculated methods. The evaluation shows that the prototype estimates the trajectory well enough to be used for the intention prediction formulated in this thesis, and that there is potential in the UWB-technology for even more accurate estimations. The intention prediction algorithm performs better in comparison to speculated systems in terms of up-time and minimising false positives, and this minimisation of false positives led to a theoretical reduction in energy consumption. This thesis is also meant as a starting point for Volvo Cars within this research area and also as a proof of concept, where ideas for future improvements are discussed. These improvements include how the inclusion of more data, such as GPS data, time of day and calendar data of the driver can be fused to further predict the driver's intention, as well as further optimisation of UWB-technology within a vehicle.

Keywords: Intention prediction, Loss function, Vehicle wake-up, Ultra-Wideband, Trajectory Estimator, False positives, False negatives, Decision-making method

Acknowledgements

First and foremost, we would like to thank the people at Volvo Cars; our technical supervisor, *Joel Lind*, who has supported us with numerous insights and valuable guidance throughout the thesis and assisted us with practical details such as providing us with a test vehicle. We would also like to thank the people in the teams *Frodo* and *Arwen* of ART Power and Energy who included us as part of their teams, which made us feel welcome as well as our time at the office in Torslanda more enjoyable. We feel that there has been a genuine interest from our colleagues regarding our thesis which has been motivating for us. We also want to thank *Johan Andersson* of ART BF at Volvo Cars who provided us with the UWB system that enabled us to develop a working prototype.

We would also like to thank our supervisor at Chalmers, *Roman Sokolovskii*, who helped us continuously throughout this thesis by giving us ambitious feedback on this report and by providing us guidance through our regular correspondences every other week - as well as when needed through quick replies to our e-mails. We also want to show our gratitude to *Erik Agrell* for being our examiner of this thesis.

As a final remark, the majority of this thesis was carried out during times far from the ordinary with lockdowns, remote meetings, quarantines and much more things that will signify the covid-19 pandemic of 2020. Nevertheless, despite these events, the people mentioned have continued to allocate parts of their time to support us which we are very grateful for. We as thesis workers are proud of the outcome of this thesis and hope that it will serve as a valuable baseline for work within the topic in the future.

Daniel Håkansson, Gothenburg, June 2020
Johan Martinsson, Gothenburg, June 2020

Contents

List of Figures

List of Tables

1
Introduction

Traditionally, many systems have been developed around the pressing of a button — from light switches and phones to ticket machines and cars. This product-centric approach has in return required the user to learn how the system works. Nowadays, as an increasing amount of technology is becoming more advanced, several fields are transitioning into making their products more user-centric. This means that the interactions with them are based on the intuition of the user rather than on the easiest way to design the system. One of these fields is the car industry, where Volvo Cars has a goal of being in the forefront of user experience. Included in their vision is to minimise the amount of manual adjustments required by the user, such as adjusting mirrors, the steering wheel and comfort systems. Instead, the car should adjust to the user automatically.

There are several systems affected by this product transition, and one of them is the *Vehicle Mode Management* (VMM) system in charge of initiating the *Vehicle Wake-up* sequence. In a Volvo car, this sequence generally consists of pressing the unlock-button on the car key and being greeted by flashing indicator lights, while underlying software systems start up and prepare the car for use. Included among the systems are a wide range of sub-systems, i.e. systems handling driver/passenger comfort, battery warm-up for Battery Vehicles and the Vehicle Mode Management system. At Volvo Cars, an important step in the product transition is that the VMM should initiate the Vehicle Wake-up sequence when sensing the user approaching and understand the intention of the user — without the manual input of pressing the unlock-button on the car key. One current method available has already removed the manual input of pressing the unlock-button, and is based on radio-polling between the car and the car key the user is assumed to be carrying. However, it still requires the user to grab the door handle, thus not fully eliminating manual inputs. With the VMM system sensing the user approaching, the ideal case is instead that the car should be ready for immediate driving when the user reaches the car, and therefore unlocking the door without the user grabbing the door handle. However, if the user doesn't intend to use the car, no waking sequence should be initiated, such as if the user simply passes the car when it is parked on the driveway.

In this thesis, the user of the vehicle will further on be referred to as the "driver", since it is considered more likely that the user is the driver of the vehicle. Various devices may serve as a car key, such as a key fob or a mobile phone, and will further on be referred as "key".

1.1 Problem formulation

In this thesis, the problem considered is that of predicting the intention of the driver in order to initiate *optimal* wake-up. An example of when it should be initiated is when the vehicle is parked on the driveway and the driver intends to use the vehicle to get to work in the morning. Here, the vehicle should consider it plausible that the driver will use the vehicle soon, with the help of specific data inputs, and initiate an optimal wake-up before the driver has reached the vehicle. This wake-up consists of processes such as battery warm-up, booting of navigation/infotainment systems and adjusting comfort settings. Another example also concerns the vehicle being parked on the driveway, but now the driver is mowing the lawn next to the driveway. In both cases, the driver is within close range of the vehicle, yet in the latter case, no wake-up sequence should be initiated since the vehicle should predict that the driver does not intend to use the vehicle. One part in the principle of optimal wake-up in the context of this thesis is that the wake-up should be tailored to the intention of the driver. This means that if the vehicle is able to predict the intention of the driver, it should be able to draw conclusions on *how* the wake-up sequence should be initiated, or if it should be initiated at all. By doing this, a minimum amount of energy resources are spent, i.e. through processing power and high-energy processes, which in return reduces the wear on system hardware of the vehicle such as the battery.

A system that tries to predict the intention of the driver will inevitably make mistakes. One type of mistake is when the driver only passes the vehicle, or approaches the vehicle and then walks away. This is referred to as a *false positive*, which is when the system, due to the faulty prediction, wakes the vehicle even though the driver didn't intend to use the vehicle. Two key trajectories showing examples of this are shown to the left in figure 1.1.

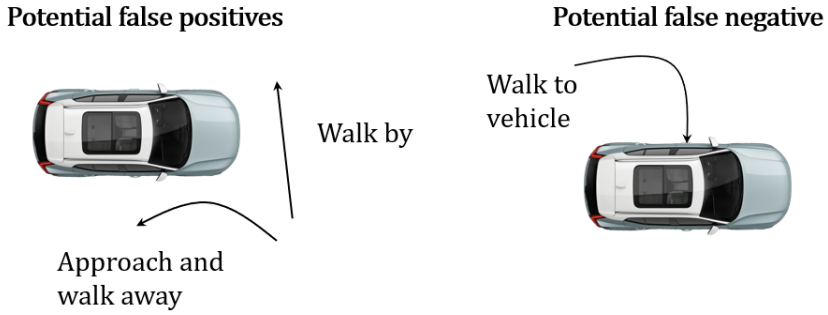


Figure 1.1: Trajectories that might lead to false positives and false negatives respectively.

At the same time, the system must be tuned to avoid *false negatives*, where the vehicle does not wake-up at a comfortable range between the vehicle and the approaching driver who intends to use it. An example of this is seen to the right in figure 1.1. Minimising these false negatives correlates with maximising user comfort.

Relating these two factors to optimal wake-up, the problem can therefore be formulated as *minimising the amount of false positives, while maximising user comfort.*

Extending the topic of maximising user comfort includes that the visible systems should not be confusing for the driver, nor frustrating or annoying. For example, the headlights should not be turning on every time the driver passes the vehicle, a function called ”welcome lights”. This also includes the automatic adjusting of seats, mirrors and other comfort systems. These are important user-related aspects that need to be treated, other than aspects related to power consumption and systems not directly visible to the driver that are handled by the minimisation of false positives. In summary, the problem formulation for this thesis is to predict the intention of the driver while minimising the amount of false positives and maximising the user comfort to achieve optimal wake-up.

1.2 Purpose

The main purpose of investigating the problems formulated in this thesis is to assist Volvo Cars on their progress towards making their products more user-centric within vehicle wake-up, and to propose beneficial technical solutions that can be used in other areas of their products. The reason for Volvo Cars to research into developing more user-centric products is due to their belief that it will improve the overall user experience. The current solution for waking the vehicle requires the user to take actions before being able to drive. Therefore, the purpose is to *minimise manual inputs required*, such as adjusting seats and mirrors, and to *minimise the waiting time* for the driver before all systems are up and running, such as infotainment systems. These are seen as valuable solutions for Volvo Cars in their progress to enhance the user experience for the driver.

Another purpose of this thesis is to research the usage of *ultra-wideband* (UWB) radar technology that Volvo Cars believe they could benefit from. They believe that this technology could be of assistance for minimising manual inputs, as it could be used to sense the driver approaching by polling the key that the driver is assumed to be carrying and thus initiate the vehicle wake-up. By also investigating this technology overall in this thesis, a starting point in the research of it can be established and used in similar applications in current and future products that has considered this type of technology.

1.3 Objectives

The main objective of this project was to research and develop a propositional prototype solution for an optimal wake-up with a minimum of manual interactions through driver intention prediction using UWB technology. This prototype will from here on be referred to as the *Driver Intention System*. This system will then be evaluated in how well it can predict the intention of the driver in relevant scenarios. The intention of the driver is based on probabilities which makes it close to

impossible to completely avoid false positives and false negatives. There is therefore a trade-off between the false positives and false negatives, where a system with zero *false positives* would simply never activate any wake-up functions, and a system with zero *false negatives* would keep the functions up all the time. Therefore, the objective was to develop a method that *finds an optimum* between minimising false positives and minimising false negatives.

Another important objective was to evaluate the performance of the UWB system, and to determine if UWB technology is suitable for the Driver Intention System. Volvo Cars also requested that the inclusion of additional data types to further model the intention of the driver was investigated. This includes data such as GPS position of the vehicle and time of the day for vehicle use.

Formally, these objectives have been divided into four main research questions (RQ) that this thesis answers and discusses.

- **RQ1:** How can the key position measured by UWB sensors be used to optimally estimate the position and trajectory of the key, and how robust is this solution?
- **RQ2:** To what extent can the Driver Intention System be used to identify false positives, that is, when the driver seemingly intends to use the vehicle but decides not to and how can they be minimised in order to minimise power consumption?
- **RQ3:** How does the Driver Intention System perform for trajectories that enter or leave the vehicle, thus trajectories that can induce false positives and false negatives respectively, compared to other proposed solutions?
- **RQ4:** How can the UWB-data and other data types such as GPS position and time of day be used to further model the intention of the driver?

1.4 Scope

The scope of this thesis is limited to the Driver Intention System representing a prototype system that is created entirely in MATLAB. Predicting the intention of the driver is also made from technical where the main baseline is the observed kinematics of the key. The Driver Intention System has a Bayesian framework as core and is divided into three main algorithms: Measurement processing, a key Trajectory Estimator (TE) and a driver Intention Predictor (IP). The Measurement processing handles outlier data in the UWB sensor data. The tracking functionality of the TE is based mainly on nonlinear filtering of UWB sensor data, to estimate the position and trajectory of the key with the use of motion and measurement models. In the IP algorithm, a decision-making function is utilized that decides which functions in the vehicle that should activate, depending on the current information that is produced from the TE algorithm. The possibility of combining data from the TE algorithm with additional data, such as vehicle GPS position and time of day, is also discussed and a proposed method for this using simulated data is introduced.

Due to this area not being extensively researched at Volvo Cars, the more general scope of the thesis is that it will serve as a starting point in their research regarding wake-up with driver intention prediction. Therefore, a solution fulfilling the fundamental requirements of wake-up through intention prediction has been considered as adequate by the company, and no extensive research for a completely integrated solution has been made. This means that the Driver Intention System will only provide the binary outputs for activating certain example functions in the vehicle. This thesis also serves as a proof of concept and as a foundation for further work within similar areas of wake-up, such as "vehicle sleep", being when the driver walks away from the vehicle after using it.

1.5 Previous work

Driver intention prediction is a field that has been widely researched lately when it comes to a vehicle in traffic and when the driver is already using the vehicle which can be read in [2] and [3]. However, there seems to have been little research for waking the vehicle by predicting the intention of the driver. There are some examples of remote wake-up such as "Passive Entry", which is only triggered by the measured distance between the key and vehicle [4], or the received signal strength (RSSI) [5]. The solution proposed in this thesis takes additional factors into account, such as evaluating the trajectory of the key to better estimate the driver's intention.

1.6 Limitations

In order to limit the scope and work during the given time span, the following limitations have been introduced in the project.

- The localisation of the key is simplified to a 2D problem, and the difference in z-axis is assumed to be negligible. With the amount of sensors used in this thesis, a robust 3D localisation is not feasible.
- Both the collected and simulated measurement data in 2D Cartesian coordinates are assumed to be perturbed by zero-mean, white Gaussian noise.
- Computations made within the prototype are made with a computer device that may require high processing power. The computations are therefore assumed to be easily transferred to another device, such as a mobile phone or to a cloud service in a later stage for potential commercial use.
- The wake-up functions in the vehicle are assumed to be activated or deactivated by binary outputs from the prototype. The project will not focus on implementing these functions in a real vehicle.
- The placement and setup of the UWB sensors is only provisional due to a complete integration within the vehicle being infeasible with the acquired UWB equipment.
- The collection of data is made with a UWB module that is connected with a USB cable to a computer to depict the key. This limits the movement of

the key and introduces a potential risk of disturbances from the computer corrupting collected measurements.

1.7 Thesis outline

In chapter 2, the theoretical preliminaries are presented in detail such that anyone with sufficient mathematical foundation can understand the thesis in its entirety. This chapter covers the vehicle wake-up preconditions, presentation of hardware, and basic theory of the implemented methods used in the project.

The methodology for the implementation of the system is described in chapter 3. By following this, it should be possible to understand how the prototype is formed and what decisions were made during the design process.

Chapter 4 covers the experimental setups which were made to produce the results that can be seen in chapter 5.

Lastly, chapter 6 covers the discussion of the presented results and of the project in its entirety. This chapter also provides some propositions regarding how the project can be further improved for future work. The last part of this chapter presents the conclusions of the project conducted in this thesis.

2
Theoretical Preliminaries

The following chapter contains information about the systems discussed at Volvo Cars to locate the key, as well as the current preconditions within that area that have resulted in the ideas for the Driver Intention System. It also presents the hardware as well as the overall theory used for the Driver Intention System. Connected to the implementation of the Driver Intention System, this includes the UWB radar sensor technology and the development board used, followed by how the sensor data is used for estimating the trajectory of the key through Bayesian filtering and finally how decision making is made through a specific Decision Making Method. Connected to the development of the Driver Intention System and the performance evaluation of it, the visualisation tool created in MATLAB and the performance metrics used are presented.

2.1 Vehicle wake-up preconditions

The latest cars developed by Volvo Cars are fitted with multiple sensors capable of sensing the presence of the key, whether inside or outside the car, and future cars are planned to have systems capable of estimating the position of the key. One system that is discussed at Volvo Cars is a multiple-zone system that senses which zone the key is located in, and activates the functions related to that zone. The location of the key is made by measuring the distance to the key as a function of the received signal strength. A visualisation of this with an example of zones can be seen in figure 2.1

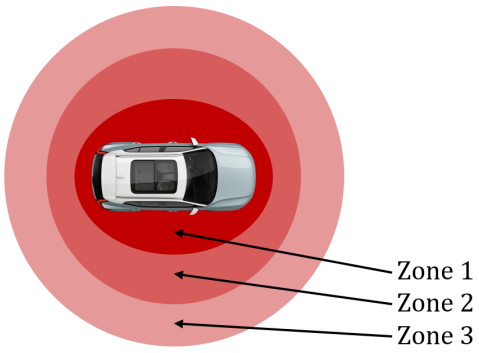


Figure 2.1: Illustration of the discussed zone system with zones connected to functions for vehicle wake-up.

As an example, when the key approaches and enters Zone 3, the wake-up sequence

is initiated. If the key continues approaching the car and enters Zone 2, the next set of systems are activated, and so forth. As the key enters the Zone 1, also being a security zone, the doors are unlocked and the driver can enter the car. If the key exits any of the zones, the wake-up sequences connected to that zone are deactivated. An overview example of the specific functions that each zone could contain can be seen in table 2.1.

Table 2.1: Example description of each zone and corresponding functions.

Zone 1	- Lock/Unlock the vehicle - Open/Close doors - Full welcome lights
Zone 2	- Eject/Retract door handle - Adjustment of comfort systems - Adjusting suspension leveling system
Zone 3	- Discrete welcome light - Infotainment systems

Another system discussed is a key positioning system with multiple sensors located around the vehicle. This system localises the key in 2D Cartesian coordinates $[x \ \ y]^\top$ in relation to a fixed coordinate system of the car. An example of the localisation using two sensors mounted on the left hand side of a car is shown in figure 2.2. The intersection of the two yellow lines, marked with a black arrow, determines the position of the key in relation to the vehicle.

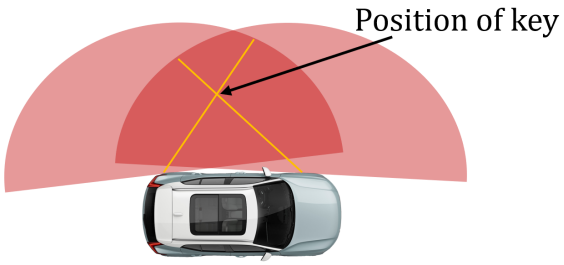


Figure 2.2: Car with multi-sensor localisation system to locate the key.

2.2 Ultra-wideband (UWB)

Based on the preconditions presented in section 2.1, this thesis investigates the use of the UWB technology as a multi-sensor localisation system. The UWB technology is a rising trend for positioning methods, although the concept for the first antenna design was patented as early as 1898 [6]. According to The Federal Communications Commission, the definition of UWB is a radio-frequency (RF) signal occupying a portion of the frequency spectrum that is greater than 20% of the center carrier frequency, or has a bandwidth greater than 500 MHz [7]. Because of the wide frequency band, the UWB transmitters are able to transmit large amount of data while maintaining very low transmit energy consumption. Also, due to its good ability of

travelling through different kinds of materials, it is a very popular choice for indoor positioning where there are many reflecting objects and other physical interference. Because of this, the UWB system is also suitable for being placed inside a vehicle where the signal can penetrate through the chassis.

Other noteworthy characteristics of the UWB system is the low signal-to-noise ratio, high accuracy (5-30 cm) [8], and low interference with other RF signals due to its low spectral density. A figure showing the difference in power spectral density for a UWB signal and a Narrowband (NB) signal can be seen in figure 2.3.

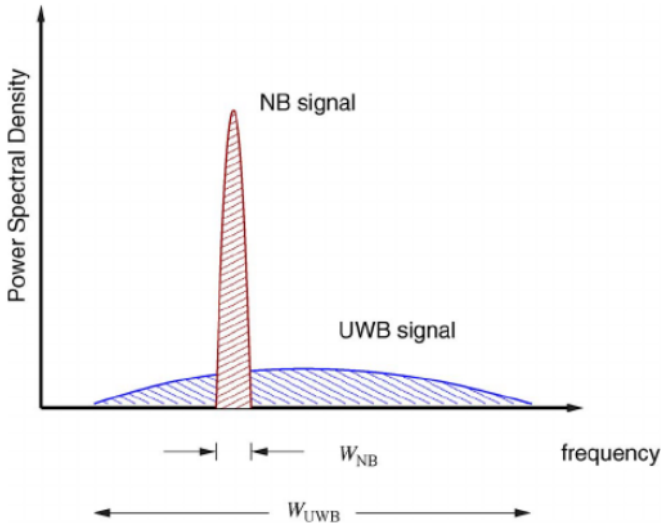


Figure 2.3: Illustration of the difference in spectral density between UWB- and NB signals [7].

2.2.1 Position estimation through trilateration with time of flight

The UWB system consists of anchors and tags where the anchors are antennas with fixed positions, and the tags are the antennas which are to be estimated. Both the anchors and the tags are configured to transmit and receive data. To compute the distance from a single tag to the anchor (r), a method called Time of Flight (ToF) is used, which evaluates the speed of the sent radio waves (speed of light, c), and the time it takes for being transmitted from the tag until it is received back (t_{ToF}),

$$r = \frac{ct_{ToF}}{2}. \tag{2.1}$$

By utilizing several receivers in the system, the distance from the single tag to the anchors can be used with trilateration to estimate the relative position of the transmitter with cartesian coordinates. An illustration of the concept for estimating position through trilateration is shown in figure 2.4.

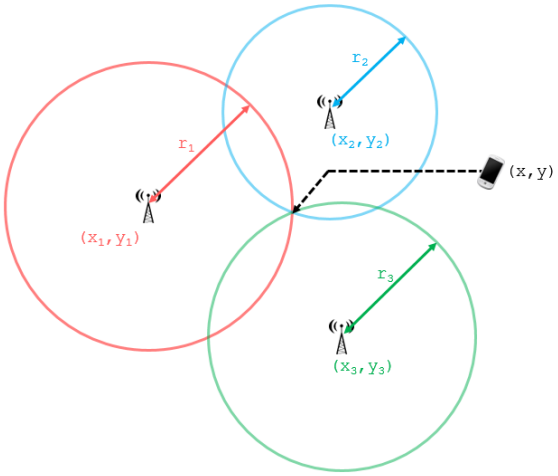


Figure 2.4: Illustration of trilateration for a phone as tag and three anchors [9].

The problem in figure 2.4 can be solved for the position of the tag, (x, y) if the three sensor positions, (x_i, y_i), with $i \in \{1, 2, 3\}$, are fixed and known, and each radius, r_i, is well estimated [9]. The position of the tag is derived from the equation for each of the circles, which is defined as

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = r_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = r_2^2 \\ (x - x_3)^2 + (y - y_3)^2 = r_3^2 \end{cases} \tag{2.2}$$

By expanding the squares for each equation, they can be rewritten as

$$\begin{cases} x^2 - 2x_1x + x_1^2 + y^2 - 2y_1y + y_1^2 = r_1^2 \\ x^2 - 2x_2x + x_2^2 + y^2 - 2y_2y + y_2^2 = r_2^2 \\ x^2 - 2x_3x + x_3^2 + y^2 - 2y_3y + y_3^2 = r_3^2 \end{cases} \tag{2.3}$$

By subtracting the second row in equation 2.3 with the first row, then subtracting the second row in the same equation with the third row, the problem can be written as the two equations below

$$(-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y = r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \tag{2.4}$$

$$(-2x_2 + 2x_3)x + (-2y_2 + 2y_3)y = r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \tag{2.5}$$

To simplify this problem, the known variables can be grouped together to get a system of the form A, B, C, D, E, F. The new system of equations can then be written as

$$\begin{cases} Ax + By = C \\ Dx + Ey = F \end{cases} \tag{2.6}$$

The solution to this equation system with two equations and two variables can then be solved

$$x = \frac{CE - FB}{EA - BD} \tag{2.7}$$

$$y = \frac{CD - AF}{BD - AE} \tag{2.8}$$

2.3 DWM1001: Two-Way-Ranging Real-Time Location System

The UWB sensor used in this thesis is called a Decawave DWM1001 module [10] which contains a Bluetooth Low Energy (BLE) module and a UWB module, and can thus communicate through both protocols. DWM1001 also includes a 3-axis accelerometer which further improves the position estimation. A schematic of the module is shown in figure 2.5.

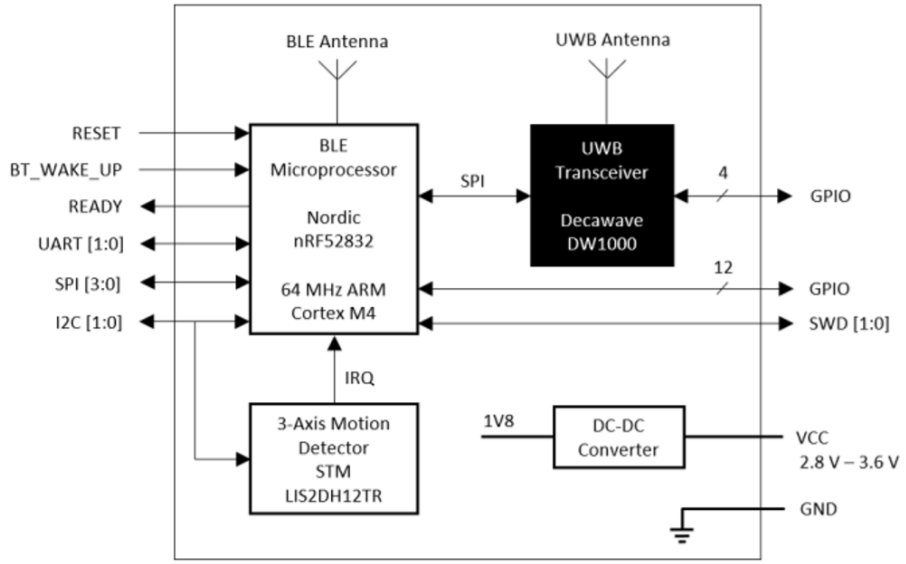


Figure 2.5: Schematic overview for the DWM1001 module containing a BLE module, UWB-module, and 3-Axis accelerometer [10].

The UWB module is a transceiver which means that it can both receive and transmit data. The BLE modules have protocols for connectivity to a smartphone or tablet (requires a 6.0 or more recent Android device) with the purpose of visualisation and configuration of the UWB system. The UWB antenna has a centre frequency of 6.5 GHz and a point-to-point range up to 60m within line of sight.

Each of the modules can be configured as an anchor, tag, or listener, where the position of the anchors can be set and fixed with cartesian coordinates. The UWB system has the potential of supporting up to thousands of tags being tracked simultaneously, whereas this project focuses on using only one or two tags to be tracked. To remotely track the position of the tag, the DWM1001 module can be configured as a listener to communicate over USB to a computer for real-time logs. The listener appears as invisible in the UWB system and serves only the purpose of evaluating the system. A visualisation for an example UWB system with four anchors, several tags and a listener can be seen in figure 2.6.

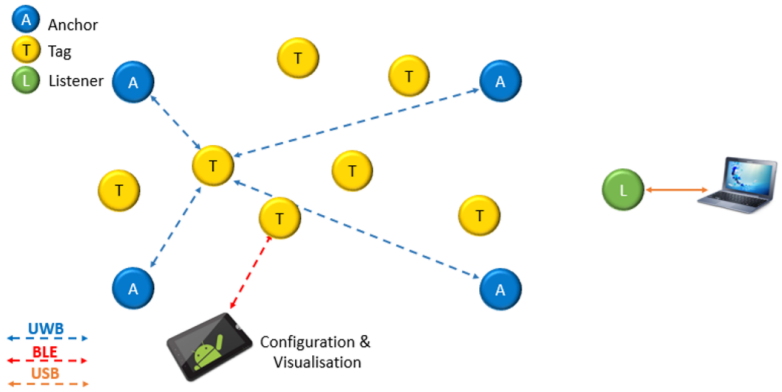
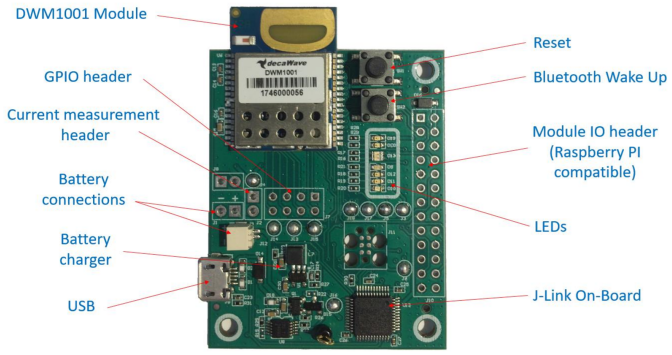


Figure 2.6: An example UWB system and the different communication protocols being in use [11].

2.3.1 Development board, DWM1001-Dev

The Development board DWM1001-Dev [12], see figure 2.7a, is an extension to the DWM1001 module with the purpose of further evaluation and programming of the module. This board can be inserted into a casing from Decaware, see figure 2.7b, which is the unit being used for this project. This casing comes with a battery slot for Lithium 123 batteries, which can act as power source for the sensor instead of the alternative of using a mini-USB cable.



(a) DWM1001-Dev board [12].

(b) Casing and battery slot for DWM1001-Dev [11].

Figure 2.7: Development board and casing for Decawave DWM1001.

2.4 Bayesian statistics, inference and filtering

With the hardware and theory of the UWB technology used in this thesis now explained, and therefore how the Driver Intention System observes its surroundings, the following section regards the processing of the gathered observations made by the system to create position estimates of the key through Bayesian filtering. The reason to why Bayesian filtering has been used is due to the observations being noisy

and that the measurements alone would produce inaccurate data of the position of the key.

The term Bayesian filtering refers to using Bayesian statistics when formulating optimal filtering, whereas optimal filtering stems from a set of methods that can be used for estimating the state of a system that is indirectly observed through noisy measurements [13]. The state of the system, denoted as $\mathbf{x}_k$ at a discrete-time instance k, can consist of dynamic variables such as position, velocity, orientation and angular velocity.

Observing a system through a noisy measurement sequence $\mathbf{y}_{1:T} = \{\mathbf{y}_1, ..., \mathbf{y}_T\}$ results in uncertainties of the system state sequence $\mathbf{x}_{0:T} = \{\mathbf{x}_0, ..., \mathbf{x}_T\}$. To mitigate this, one can use Bayesian inference where the state of the system is modelled as a probability density, p, rather than by a fixed deterministic value. This is achieved by using Bayes' Theorem where the *posterior density*, $p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T})$, is derived using a *prior*, $p(\mathbf{x}_{0:T})$ and a *likelihood*, $p(\mathbf{y}_{1:T}|\mathbf{x}_{0:T})$. The prior distribution describes all the state densities up until discrete-time instance T and the likelihood describes the probable evolution of the measurements given the prior. Given the prior distribution and the likelihood, the posterior distribution of all the states given the history of measurements can be summarised through Bayes' theorem as

$$p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) = \frac{p(\mathbf{y}_{1:T}|\mathbf{x}_{0:T})p(\mathbf{x}_{0:T})}{p(\mathbf{y}_{1:T})} \propto p(\mathbf{y}_{1:T}|\mathbf{x}_{0:T})p(\mathbf{x}_{0:T}). \tag{2.9}$$

However, working with full sequences ranging from 0 to T and computing posterior distributions of all the states quickly becomes problematic as soon as additional measurements are obtained, due to the set of states growing larger. This needs to be addressed for real-time applications regardless of computational resources. The solution to this is to model the system states as *marginal distributions of possible state variables* and restricting the class of the system models as Markovian [13]. This produces a more general probabilistic state space model with *three main components*:

1. **A prior distribution**, $\mathbf{x}_0 \sim p(\mathbf{x}_0)$, a distribution describing the initial state distribution at discrete time step $k = 0$ or the previous state distribution at discrete step $k > 0$.
2. **A system model**, $\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{x}_{k-1})$, a transition probability distribution specifying the system dynamics and its uncertainty as a Markov sequence. With the system model, it is possible to make a *prediction* of what the next state might be given the history of measurements up until $k - 1$.
3. **A measurement model**, $\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}_k)$, also called *likelihood*, a conditional probability distribution of the measurement given the state, specifying how the measurements $\mathbf{y}_k$ might evolve based on the current state, $\mathbf{x}_k$. With the measurement model, it is possible to *update* the prediction made by the system model using an observation made at discrete-time instance k.

With the probabilistic state space model above, theorem 4.1 in [13] shows that the computation of the posterior distribution up until a measurement made at discrete

time step k now can be summarised as:

$$p\left(\mathbf{x}_k|\mathbf{y}_{1:k}\right) \propto p\left(\mathbf{y}_k|\mathbf{x}_k\right) p\left(\mathbf{x}_0\right), \quad \text{for } k = 0 \tag{2.10}$$

$$p\left(\mathbf{x}_k|\mathbf{y}_{1:k}\right) \propto p\left(\mathbf{y}_k|\mathbf{x}_k\right) p\left(\mathbf{x}_k|\mathbf{y}_{1:k-1}\right), \quad \text{for } k > 0 \tag{2.11}$$

where

$$p\left(\mathbf{x}_k|\mathbf{y}_{1:k-1}\right) = p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right) p\left(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}\right). \tag{2.12}$$

What has now been obtained are the basic building blocks of a *Bayesian model*, a crucial step in the context of *Bayesian filtering*, the recursive methodology to compute the posterior density. In order to practically make use of these building blocks provided by Bayesian inference, numerical methods are necessary for computing the posterior distributions. In this thesis, *Gaussian approximators* are used to approximate the posterior distributions as Gaussian distributions, meaning that the posterior distribution is formulated in the following way

$$p\left(\mathbf{x}_k|\mathbf{y}_{1:k}\right) \simeq \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}). \tag{2.13}$$

where $\hat{\mathbf{x}}_{k|k}$ is the posterior mean, also being the state estimation, and $\mathbf{P}_{k|k}$ is the posterior covariance matrix, both at discrete-time instance k given all the information up until k. How these are calculated will be specifically dependent on which Bayesian filtering method is applied, which further determines the performance of the filtering. This will be covered further in section 2.4.1.

As mentioned earlier, two of the three main components in a Bayesian model is the system model and the measurement model. When it comes to target tracking applications, these can be modelled in a wide variety of ways depending on the system and measurement properties. The system model, denoted as f, describes the evolution of the system states, such as the movement of a target in 2D. The uncertainty of this evolution, or *process noise*, is defined by the *process noise covariance matrix* $\mathbf{Q}_k$. The measurement model, denoted as h, describes the likely evolution of a measurement given the state $\mathbf{x}_k$. The uncertainty of this evolution, or *measurement noise*, is defined by the *measurement noise covariance matrix* $\mathbf{R}_k$. The matrix $\mathbf{Q}_k$ together with the chosen filtering method defines the predicted covariance matrix $\mathbf{P}_{k|k-1}$, which is the uncertainty of the predictions made by the system model. Similar reasoning can be done for the measurement model, where the matrix $\mathbf{R}_k$ together with the chosen filtering method defines the posterior covariance matrix $\mathbf{P}_{k|k}$, and therefore the uncertainty, or covariance, of each state estimation. How these covariance matrices are computed in detail depends on the choice of filtering method used in this thesis, which will be covered in 2.4.1. Concerning the models used in this thesis, a Gaussian 2D Coordinated Turn model with polar velocity components is used as the system model. This model has the discrete-time state vector

$$\mathbf{x}_k = \begin{bmatrix} p_{xk} & p_{yk} & v_k & \alpha_k & \omega_k \end{bmatrix}^\top \tag{2.14}$$

where (p_{xk}, p_{yk}) is the target position in Cartesian coordinates, v_k velocity, α_k the heading and ω_k the turn-rate to model the movement of the key as seen in figure 2.8.

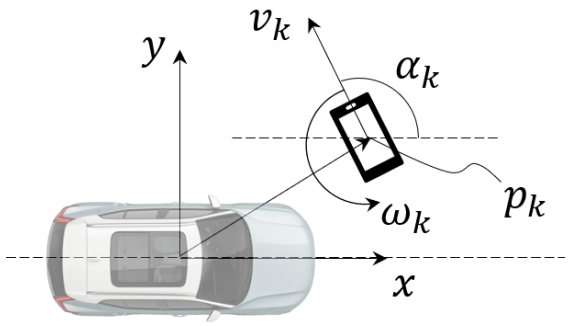


Figure 2.8: Illustration of the rotational kinematics of the target, in this case a smartphone, using the Coordinated Turn Motion Model with Polar Velocity.

In figure 2.8, the target heading $\alpha_k \in [-\pi, \pi]$, where the velocity, v_k, takes positive values for $\alpha_k > 0$ and negative values for $\alpha_k < 0$. The values of the turn rate ω_k depends on the discrete time derivative of α_k. Since the model considers the motion of a target, it can further on be considered as a motion model. As described in [14], this model assumes that the target is driven by linear rotational acceleration inputs v and ω, which in this thesis are decided by the additive Gaussian noise parameters σ_v and σ_ω. The discretised model is of the form

$$\mathbf{x}_{k+1} = f(\mathbf{x_k}) + G(\mathbf{x_k})\psi_k, \quad \psi_k = \begin{bmatrix} \sigma_v^2 & \sigma_\omega^2 \end{bmatrix}^\top \tag{2.15}$$

where

$$f(\mathbf{x_k}) = \begin{bmatrix} p_{x_k} + \frac{2v_k}{\omega_k}\sin(\frac{\omega_k T}{2})\cos(\alpha_k + \frac{\omega_k T}{2}) \\ p_{y_k} + \frac{2v_k}{\omega_k}\sin(\frac{\omega_k T}{2})\sin(\alpha_k + \frac{\omega_k T}{2}) \\ v_k \\ \alpha_k + \omega_k T \\ \omega_k \end{bmatrix}, \; G(\mathbf{x}_k) = \begin{bmatrix} \frac{T^2}{2}\cos(\alpha_k) & 0 \\ \frac{T^2}{2}\sin(\alpha_k) & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \tag{2.16}$$

and T being the sample time. This model setup has shown to be a sufficient choice in terms of tracking performance as shown in [14] and [15]. By using the matrix G, a zero-order-hold behaviour is established on the inputs creating a smoother input sequence rather than having impulses for each sample instance. With G and ψ_k, the process noise covariance matrix $\mathbf{Q}_k$, consisting of the two input parameters σ_v and σ_ω, is modelled as

$$\mathbf{Q}_k = G(\mathbf{x}_k)\,\mathrm{diag}(\sigma_v^2, \sigma_\omega^2)G(\mathbf{x}_k)^\top. \tag{2.17}$$

The measurement model used in this thesis is a linear model. This is due to the Bayesian filtering utilising the trilaterated measurements from the UWB-sensors being in 2D Cartesian coordinates $[x \;\; y]^\top$ while the Bayesian filter outputs estimates in the same form. Therefore, the measurement model is as follows:

$$h(\mathbf{x}_k) = \mathrm{diag}(p_{x_k}, p_{y_k}, 0, 0, 0). \tag{2.18}$$

The measurement model covariance matrix $\mathbf{R}_k$ depends on the type of sensors used and can be modelled through experiments. In this thesis, this has been done through experiments with the UWB-sensors. The chosen covariance matrix will be explained further on in chapter 3.

2.4.1 Gaussian approximations and the Unscented Kalman Filter

In the context of target tracking, Bayesian filtering is a common and powerful type of method to determine the state of a target with nonlinear properties, for example, movement. This is often realised with a nonlinear Kalman Filter, such as one that estimates the system states through Gaussian approximations which has shown to be an effective way to estimate state evolutions of a nonlinear system [13], [15]. Traditionally, the Kalman Filter is a common choice for estimating the state of a system with linear properties. To handle systems that are nonlinear, many different types of extended version of the Kalman Filter has been developed, such as the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF).

The filter used in this thesis is the Unscented Kalman Filter, which is a relevant choice of filter for applications similar to the one treated in this thesis, as shown in [15] where it was used for a tracking application. Other work, such as [16], has also shown that the traditional Kalman Filter performs well together with UWB-sensors. In [16], the author suggests working with a nonlinear Kalman Filter to achieve better performance, such as the Extended Kalman Filter, to better handle disturbances caused by an obstructed line of sight. In this thesis, the Unscented Kalman Filter is applied instead of the Extended Kalman Filter, due to its better performance according to the results shown in [15] as well as its more straightforward implementation and less heavy computations as discussed in [14].

There are different ways of implementing a nonlinear Kalman Filter constructed as a Gaussian approximator. However, all Gaussian approximators rely on a prior distribution of the state estimate that initialises the filtering and that this is followed by two main steps computed recursively:

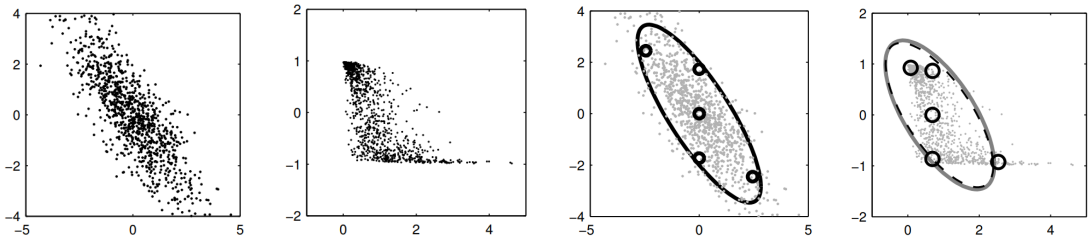
1. **A prediction step**: $\hat{\mathbf{x}}_{k|k-1} \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})$, where given the prior state estimation $\hat{\mathbf{x}}_{k-1|k-1}$, a prediction of the next state $\hat{\mathbf{x}}_{k|k-1}$ is based on the system model f and predicted covariance matrix $\mathbf{P}_{k|k-1}$.
2. **An update step**: $\hat{\mathbf{x}}_{k|k} \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$, where the posterior $\hat{\mathbf{x}}_{k|k}$ is computed by combining the prior and measurement likelihood, as first introduced in Bayes' theorem in equation 2.9.

The way the Unscented Kalman Filter conducts these two tasks is by relying on the Unscented Transform for the approximation of filtering distributions. It does this by approximating the mean and covariance of the nonlinear function directly by generating a set of sampled *sigma points* using the prior mean $\hat{\mathbf{x}}_{k-1|k-1}$ and covariance $\mathbf{P}_{k-1|k-1}$. As explained in chapter 5.5 of [13], the Unscented Transform is a numerical method that can be used for approximating the joint distribution of random variables $\mathbf{x}$ and $\mathbf{y}$ that are defined as

$$\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{P}), \quad \mathbf{y} = \mathbf{g}(\mathbf{x}) \tag{2.19}$$

It is also mentioned that the idea of the Unscented Transform is that it is easier to

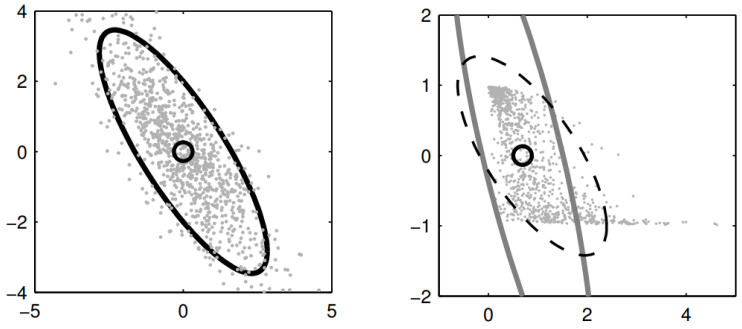
approximate a probability distribution than it is to approximate the nonlinear function **g**, therefore a set of sigma points are chosen to capture the mean and covariance of the original distribution of **x** directly. These sigma points are then propagated through the non-linearity of **g** and the mean and covariance of the transformed variable are estimated directly from them. As can be seen in [13], the Unscented Transform forms the sigma points and conducts the Gaussian approximation as illustrated in figure 2.9.



(a) Original **(b)** Transformed **(c)** Original with sigma points **(d)** Transformed with sigma points

Figure 2.9: A non-linear transformation applied on a random variable, shown in 2.9a, resulting in the random variable shown in 2.9b. The Unscented Transform then approximates the mean and covariance of the transformed random variable by computing sigma points (black circles) in the original random variable in 2.9c and then propagates them through the nonlinear function **g** resulting in what can be seen in 2.9b. It is from 2.9b that the mean and covariance then are directly estimated from. The dashed line represents the true distribution and the solid line is the approximation. All figures are adapted from [13].

To show how the performance using the Unscented Transform differs from the more traditional Extended Kalman Filter, [13] also illustrates what the corresponding results of the linearisation-based approximation of the EKF would look like in figure 2.10b.



(a) Original, EKF **(b)** Transformed, EKF

Figure 2.10: Illustration of the linearisation-based transformation made by the Extended Kalman Filter. Both figures are adapted from [13].

As one can see when comparing 2.9d and 2.10b, the approximated covariance com-

puted by the Unscented Transform is far more accurate than the one made by the EKF, which further results in more confident state estimations.

In figure 2.9c, a total of five sigma points are used and the placement of these are directly related to the prior mean and covariance and weights W_0 and W_i. In this thesis, due to $\hat{\mathbf{x}}_{k-1|k-1}$ is assumed Gaussian, the weights are assigned as

$$W_0 = 1 - \frac{n}{3}\ ,\ W_i = \frac{1 - W_0}{2n} \tag{2.20}$$

where n is the amount of state variables in $\hat{\mathbf{x}}_{k-1|k-1}$. To connect to the state representations used in this thesis, the variables $\mathbf{x}$, $\mathbf{P}$ and $\mathbf{g}$ introduced in 2.19 can be substituted with $\hat{\mathbf{x}}_{k-1|k-1}$, $\mathbf{P}_{k-1|k-1}$ and f for the prediction step and $\hat{\mathbf{x}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$ and h for the update step. With reference to the detailed theory in [13], the Unscented Kalman Filter computes the posterior mean and covariance the following way:

- **Prediction step:**
  1. Form a set of $2n + 1$ sigma points using the prior mean and covariance $\hat{\mathbf{x}}_{k-1|k-1}$ and $\mathbf{P}_{k-1|k-1}$

$$\mathcal{X}_{k-1}^{(0)} = \hat{\mathbf{x}}_{k-1|k-1}, \tag{2.21}$$

$$\mathcal{X}_{k-1}^{(i)} = \hat{\mathbf{x}}_{k-1|k-1} + \sqrt{\frac{n}{1 - W_0}}(\mathbf{P}_{k-1|k-1}^{1/2})_i \tag{2.22}$$

$$\mathcal{X}_{k-1}^{(i+n)} = \hat{\mathbf{x}}_{k-1|k-1} - \sqrt{\frac{n}{1 - W_0}}(\mathbf{P}_{k-1|k-1}^{1/2})_i \tag{2.23}$$

  2. Propagate the sigma points through the system model f

$$\hat{\mathcal{X}}_k^{(i)} = f(\mathcal{X}_{k-1}^{(i)}) \tag{2.24}$$

  3. Compute the predicted mean $\hat{\mathbf{x}}_{k|k-1}$ and the predicted covariance $\mathbf{P}_{k|k-1}$:

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2n} W_i \hat{\mathcal{X}}_k^{(i)} \tag{2.25}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=0}^{2n} W_i \left(\hat{\mathcal{X}}_k^{(i)} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\hat{\mathcal{X}}_k^{(i)} - \hat{\mathbf{x}}_{k|k-1}\right)^\top + \mathbf{Q}_{k-1} \tag{2.26}$$

- **Update step:**
  1. Form a new set of $2n + 1$ sigma points using the predicted mean and covariance $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$, where n is the amount of state variables in $\hat{\mathbf{x}}_{k|k-1}$.

$$\mathcal{X}_k^{(0)} = \hat{\mathbf{x}}_{k|k-1}, \tag{2.27}$$

$$\mathcal{X}_k^{(i)} = \hat{\mathbf{x}}_{k|k-1} + \sqrt{\frac{n}{1 - W_0}}(\mathbf{P}_{k|k-1}^{1/2})_i \tag{2.28}$$

$$\mathcal{X}_k^{(i+n)} = \hat{\mathbf{x}}_{k|k-1} - \sqrt{\frac{n}{1 - W_0}}(\mathbf{P}_{k|k-1}^{1/2})_i \tag{2.29}$$

2. Propagate the sigma points through the measurement model h

$$\hat{\mathcal{Y}}_k^{(i)} = h(\mathcal{X}_k^{(i)}) \tag{2.30}$$

3. Compute the measurement likelihood $\hat{\mathbf{y}}_{k|k-1}$, the predicted measurement covariance $\mathbf{S}_k$ and the cross-covariance of the state and the measurement $\mathbf{C}_k$:

$$\hat{\mathbf{y}}_{k|k-1} = \sum_{i=0}^{2n} W_i \hat{\mathcal{Y}}_k^{(i)} \tag{2.31}$$

$$\mathbf{S}_k = \sum_{i=0}^{2n} W_i \left(\hat{\mathcal{Y}}_k^{(i)} - \hat{\mathbf{y}}_{k|k-1}\right)\left(\hat{\mathcal{Y}}_k^{(i)} - \hat{\mathbf{y}}_{k|k-1}\right)^\top + \mathbf{R}_k \tag{2.32}$$

$$\mathbf{C}_k = \sum_{i=0}^{2n} W_i \left(\mathcal{X}_k^{(i)} - \hat{\mathbf{x}}_{k|k-1}\right)\left(\hat{\mathcal{Y}}_k^{(i)} - \hat{\mathbf{y}}_{k|k-1}\right)^\top \tag{2.33}$$

4. Compute the Kalman Filter gain $\mathbf{K}_k$, the posterior state mean $\hat{\mathbf{x}}_{k|k}$ and posterior covariance $\mathbf{P}_{k|k}$ conditional on the obtained measurement $\mathbf{y}_k$:

$$\mathbf{K}_k = \mathbf{C}_k \mathbf{S}_k^{-1} \tag{2.34}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left[\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}\right] \tag{2.35}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top \tag{2.36}$$

With the posterior mean and covariance obtained, the state estimation $\hat{\mathbf{x}}_{k|k} \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$ has been computed using the Unscented Kalman Filter.

2.5 Decision-making method

There are many different methods for making decisions, and to know if a chosen method is suitable or not depends on the given environment and accessible types of data. Initially, Bayesian Decision Theory was planned to be implemented as the decision-making method, which incorporates the uncertainty of the states to find optimal decisions [17]. However, a lack of prior knowledge for the covariance of the chosen states led to the choice of evaluating *deterministic values*, rather than probability distributions. Aside from this difference, the method that was developed shares much of the same fundamental structure as the Bayesian Decision Theory, which is presented in this section. Further explanation of the design choices behind the decision-making method of this thesis can be found in chapter 3.4.

To summarize the objective of this decision-making method, the goal was to develop a method that quantifies the intention of the driver through a loss function, mainly from evaluating the movement of the key. This value should then be evaluated with a decision function to decides which functions to activate in the vehicle.

2.5.1 Decision space

The decisions, $a \in \mathcal{A}$, can be formulated in various ways depending on the problem or situation. It can for example be expressed as a real value ($a \in \mathcal{R}$), multidimensional

value ($a \in \mathcal{R}^n$), or binary value ($a \in \{0, 1\}$). For this thesis, the decision will be binary values for three actions in a vector, $\mathbf{a} = [a_1, a_2, a_3]^\top$, which are listed below. The functions for each zone can be seen in table 2.1.

a_1 : Start/Turn off Zone 1 functionality
a_2 : Start/Turn off Zone 2 functionality
a_3 : Start/Turn off Zone 3 functionality

2.5.2 Information space

The elements of the information space $\theta \in \Theta$ can also be expressed in several different ways, similarly to the decisions. The information for this project is based on the distance and movement of the key which are expressed as real values (see section 3.4). More information can easily be added to the information space even if they are not expressed as real values, such as time of the day, GPS position of vehicle, events, etc.

2.5.3 Loss function

The loss function is used to evaluate how good a decision is depending on given elements of the information space. The result from a loss function is always a real value, and a simplified expression can be seen in equation 2.37.

$$\mathcal{L} : \Theta \to \mathcal{R} \tag{2.37}$$

The choice of loss function is subjective, but the general purpose is to evaluate the loss of each decision depending on the current information to quantify which decision is most suitable for the current situation. This project focuses on achieving a low return value from the loss function when the probability of using the vehicle is high. In essence, this return value captures the intention of the driver, which is used for making the decision. A more in-depth description of the loss function for the project can be seen in section 3.4.

2.5.4 Decision function

Unlike the standard use of Bayesian Decision Theory, a simple decision function was created to convert the output value from the loss function into the actions that activate/deactivate the functions in the vehicle;

$$\mathcal{D}(\mathcal{L}) : \mathcal{R} \to \mathbf{a} \tag{2.38}$$

The decision function consists of threshold values for Zone 2 and Zone 3, that are denoted by τ_{z2} and τ_{z3}. The function outputs $a_i = 0$ for each zone ($i \in \{2, 3\}$) if the loss value is higher than the respective threshold, and $a_i = 1$ if the loss value is lower. Due to security reasons, Zone 1 was designed to be activated only by the distance to the vehicle and thus not included as a decision to be made from a loss value threshold. Further description of the decision function can be found in section 3.4.

2.6 Performance evaluation

With the technology for how the Driver Intention Systems observes its surroundings presented, as well as the theory for how these observations are used, this final sub section explains the development tool and performance evaluation metrics used. For the development of the Driver Intention System and to obtain a visual way of studying its performance, a visualisation tool was developed in MATLAB. With this tool, the performance of the accuracy and precision of the position estimates made by the system could be studied using specific metrics such as Cumulative Distribution Functions (CDF) and Root Mean Square Error (RMSE). These are common metrics studied in the performance of tracking algorithms such as UWB-radar systems [18]. Metrics to study the overall performance of the Driver Intention System have been established as well. This has been done by studying how the Driver Intention System handles false positives/false negatives in comparison to other similar systems, such as the discussed solution by Volvo Cars using zone-based systems based on distance introduced in section 2.1. Doing these comparisons have also allowed the studying of the performance of the Driver Intention System in terms of energy consumption. The performance for optimally waking the vehicle is also evaluated, where the distance from the driver to the vehicle is of interest and the performance of finding an optimum between minimising false positives and false negatives for the system has been evaluated.

2.6.1 Visualisation tool in MATLAB

The first step to study the performance of the Driver Intention System was to develop a visualisation tool using MATLAB. This tool visualises the trajectory of the key, and is also made for evaluating the functionality of the TE and IP algorithms. The plot of this visualisation tool can be seen in figure 2.11.

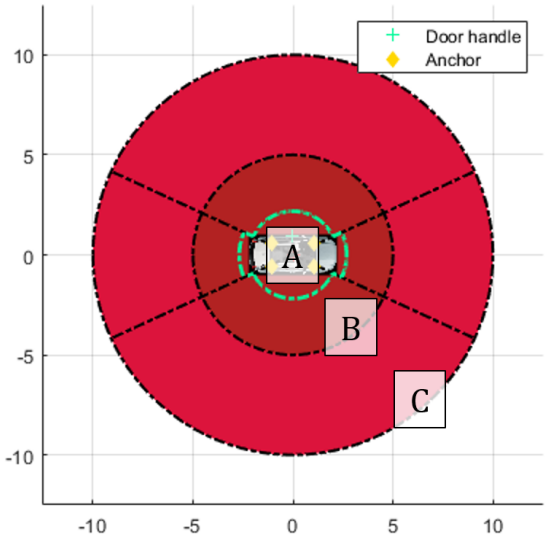


Figure 2.11: A plot made by the visualisation tool in MATLAB showing a car surrounded by three static zones A, B and C

The vehicle seen in the center of figure 2.11 is a Volvo XC40, which is the car model used during the experimental setups in section 4. Surrounding the vehicle are three static zones, where the zone closest to the vehicle, Static Zone A, is the area used for unlocking the security zone, namely Zone 1. This static zone is based on the distance from the centre of the vehicle to its corner points, $l_c = 2.400$ m, which is calculated from the vehicle width, $l_w = 1.863$ m and vehicle length, $l_l = 4.425$ m. The other two zones are defined as circles with the radius of 5 m and 10 m, and are included for the viewer to easily identify the distance of the measurements to the centre of the vehicle. The exact measurements for each zone are presented in table 2.2.

Table 2.2: Radius values in metres of the static zones A, B and C.

Static zone A	Radius $\in [0,\ \ l_c - 0.25] \approx [0,\ \ 2.15]$ front and rear of car Radius $\in [0,\ \ l_c + 0.25] \approx [0,\ \ 2.65]$ left and right of car
Static zone B	Radius $\in [\sim 2.15,\ \ 5]$ front and rear of car Radius $\in [\sim 2.65,\ \ 5]$ left and right of car
Static zone C	Radius $\in [5,\ \ 10]$

2.6.2 Accuracy and precision

The accuracy is calculated by evaluating the Euclidean distance between the estimated coordinates, $\hat{p}_{x,k}, \hat{p}_{y,k}$ and the true coordinates, $p_{x,k}^{true}, p_{y,k}^{true}$ of the key through the following equation

$$TE_{accuracy} = \sqrt{(\hat{p}_{x,k} - p_{x,k}^{true})^2 + (\hat{p}_{y,k} - p_{y,k}^{true})^2} \tag{2.39}$$

Evaluating the precision produces metrics that can be used to assess the robustness of the position estimates produced by the TE as it reveals the variation of position estimates over several sequences of filtered measurement data, each with a sequence length N, also referred as "runs". To compute the precision of the position estimates during one run requires the median of all state estimates made in that run, $\hat{p}_{y,1:N}^{median}$, followed by the Euclidean distance to each estimated position, which is summarised in the following equation

$$TE_{precision} = \sqrt{(\hat{p}_{x,k} - \hat{p}_{x,1:N}^{median})^2 + (\hat{p}_{y,k} - \hat{p}_{y,1:N}^{median})^2}, \quad N = \text{Sequence length} \tag{2.40}$$

Combining accuracy and precision, the accuracy of the estimations can be expressed as a Cumulative Distribution Function (CDF) which gives a metric for the performance of the position estimates that is often easier to understand, i.e. that an estimate has a precision of 1 metre with a 90 percent accuracy.

2.6.3 Root Mean Square Error, RMSE

This metric presents the error between the true and of any estimated state class ν, and is for each estimate at discrete-time instance k calculated by

$$RMSE_\nu = \sqrt{\frac{\sum(\nu_k^{true} - \hat{\nu}_k)^2}{N}}, \quad N = \text{Sample size} \tag{2.41}$$

3

Driver Intention System

The following chapter explains how the Driver Intention System has been implemented using the theoretical preliminaries introduced in the previous chapter. It also explains further methods used to gather and make use of the measurements from the UWB-sensors, as well as motivations for the specific design choices made.

3.1 System outline

As introduced in the first chapter of this thesis, the implemented system consists mainly of three function blocks; Measurement processing, the Trajectory Estimator (TE) and the Intention Predictor (IP). In this section, their purpose and functionality is explained as well as motivations behind some of the main design decisions that were made during their development. Figure 3.1 shows an overview of the system as a block diagram.

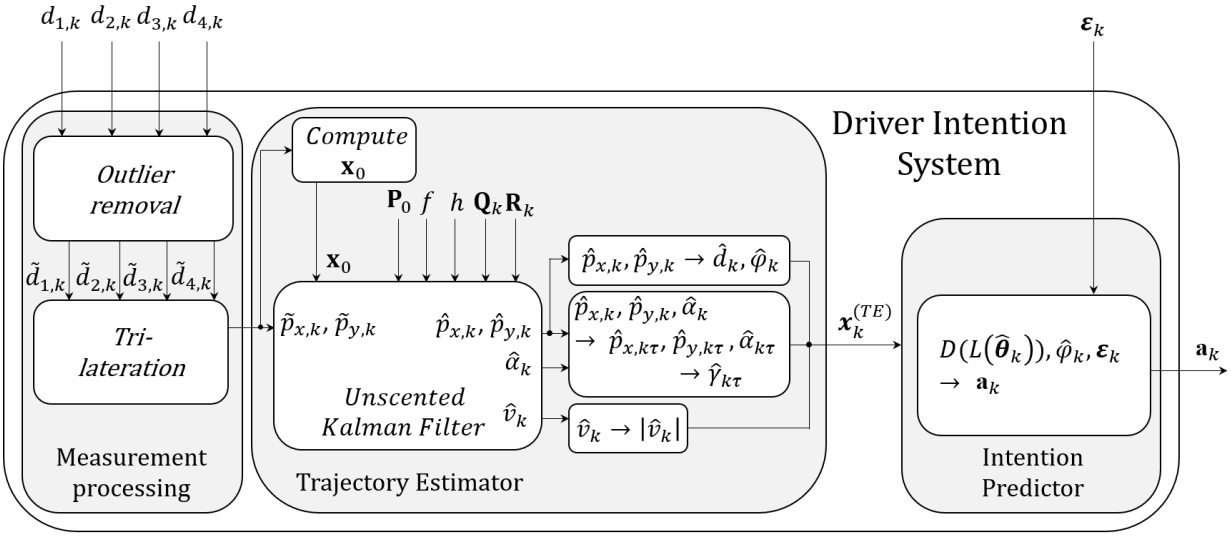


Figure 3.1: An overview of the Driver Intention System where the measured distances to the key $d_{i,k}$ with $i = 1...4$ for each anchor are converted into the vector $\mathbf{x}_k^{(TE)}$ used in the Intention Predictor to create the boolean action vector $\mathbf{a}_k$ describing which actions to be performed.

The variables shown in figure 3.1 are presented with explanations in table 3.1.

Table 3.1: Explanations to the variables used in the Driver Intention System seen in figure 3.1.

Alg.	Variable	Explanation
MP	$d_{i,k}$	Measured distance from anchor i at instance k to the key
	$\tilde{d}_{i,k}$	Processed measurement from anchor i at instance k to the key
	$\tilde{p}_{x,k}, \tilde{p}_{y,k}$	Key coordinates based on processed measurements
TE	$\mathbf{x}_0$	Prior state mean vector
	$\mathbf{P}_0$	Prior state covariance matrix
	f	Motion model function
	h	Measurement model function
	$\mathbf{Q}_k$	Process noise covariance matrix
	$\mathbf{R}_k$	Measurement noise covariance matrix
	$\hat{p}_{xk}, \hat{p}_{yk}$	Estimated key coordinate
	$\hat{\alpha}_k$	Estimated heading of the key w.r.t. axis parallel to the car
	$\hat{v}_k$	Estimated velocity of the key
	$\hat{d}_k$	Estimated distance to the key
	$\hat{\gamma}_{k\tau}$	τ:th estimated heading of the key w.r.t. center of car
	$\lvert\hat{v}_k\rvert$	Absolute value of estimated velocity of the key
	$\hat{\varphi}_k$	Estimated angle of arrival of the key
	$\mathbf{x}_k^{(TE)}$	TE-state vector consisting of $\hat{\boldsymbol{\theta}}_k$ and $\hat{\varphi}_k$
IP	$\boldsymbol{\varepsilon}_k$	Parameters such as GPS position and time of day
	$\hat{\boldsymbol{\theta}}_k$	IP-states consisting of $\hat{d}_k$, $\hat{\gamma}_{k\tau}$, $\lvert\hat{v}_k\rvert$
	$\mathcal{L}(\hat{\boldsymbol{\theta}}_k)$	Loss value, describing the probability of driver using vehicle.
	$\mathcal{D}(\mathcal{L}(\hat{\boldsymbol{\theta}}_k))$	Decision function, activates zones depending on the loss value
	$\mathbf{a}_k$	Action describing which zone/zones has been activated

3.1.1 Choice of system design

The development of the Driver Intention System required several design choices, but the main design choice was based on the point of view from which the modelling of driver intention was made. In this thesis, modelling the intention of the driver is made from a kinematic point of view, which narrowed down to the three state variables used in the loss function, namely the distance d,relative heading γ and velocity v. The relation between these variables and the centre of the vehicle can be seen in figure 3.2.

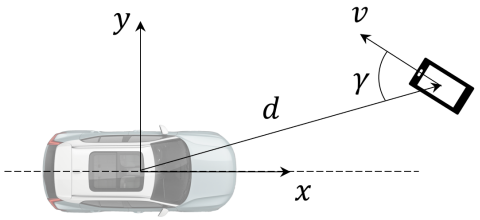


Figure 3.2: Relation between the variables d, γ and v and the centre of the vehicle. Here illustrated with smartphone assumed to be carried by the driver.

For the sake of notation simplicity, the chosen variables can be combined into a single vector $\boldsymbol{\theta}$. The relationship between the chosen variables, either a low or high intention of using the vehicle and either a low or high loss value can be seen in table 3.2.

Table 3.2: Extreme-values of the IP-states modelling the intention of the driver

d	γ	v	Intention of using the vehicle	$\mathcal{L}(\boldsymbol{\theta})$
Low	Low	High	High	Low
High	High	Low	Low	High

This means that a low distance value d, a low relative angle value γ and a high velocity value v – all together corresponding to the driver being close to vehicle, facing the vehicle and travelling at a high velocity – implies high intention. Contrarily, a high distance value d, a high relative angle value γ and a low velocity value v – all together corresponding to the driver being far away from the vehicle, facing away from the vehicle and travelling at a low velocity – implies a low intention. To make the relationship between the intention and the loss function more intuitive, one can think of the intention of using the vehicle as always being high, since it is the goal of the driver in the vehicle approach context of this thesis, and that the loss function decides how much this intention is reduced. Therefore, a low loss means a high intention, and a high loss means a low intention. Since this thesis focuses on the driver approaching the vehicle, the main problem became constructing a system that could convert the data from the UWB-system into the variables d, γ and v, and through these, mathematically model the intention of the driver.

The first block seen in figure 3.1, measurement processing, gathers raw measurements from the UWB system and pre-processes and refines these by removing outliers. The pre-processed measurements are then trilaterated into 2D Cartesian coordinates that are then used as inputs to the next block, the TE algorithm. Here, the choice of filtering method was based on the desire that the information regarding the movement of the key is highly accurate as well as the assumed properties of the obtained trilaterated measurements being zero-mean and Gaussian. Having detailed information about the movement of the key was considered fundamental for the accuracy of the intention prediction. This therefore required modelling the movement of the key with a nonlinear model since the model produces such adequate range of information.

With the choice of using a nonlinear model to model the movement of the key, this led to the choice of using a nonlinear Kalman Filter for position estimation. Since the measurements output from the measurement processing are assumed as Gaussian, this lead to the choice of a Gaussian approximator as Kalman Filter. The Unscented Kalman Filter was chosen due to it being a well-performing and common type of Gaussian approximator used in tracking applications. With a high-performing filter producing position estimates, these estimates are converted into states that are used by the loss function to quantify the intention of the driver. Finally, a decision function uses the loss values produced by the loss function and decides which functions

in the vehicle that should be activated. The detailed explanation of these functionalities and the design choices made for each function block are further presented in the following subsections of this chapter.

3.2 Measurement processing

This section describes the chain of events from extracting the distance for each sensor to how the trilaterated measurements $\tilde{p}_{x,k}, \tilde{p}_{y,k}$ are processed to the form which is used as input for the Unscented Kalman Filter. By connecting the tag to a computer with USB, the raw measured distances from the anchors to the tag can be logged to the computer device with a sample rate of 10 Hz. There are several ways of logging the measurements, both for real-time and batch processing. This section will only cover the method for batch processing which is used for the results in this report, whereas the real-time solution is only used for live demonstration. The chosen method for batch processing is to use the inbuilt UART shell through a terminal program, Tera Term. To connect and communicate with the UART shell, the COM port for the tag must be selected, and the terminal baud rate needs to be set to 115200 bits per second. The information is then logged to a file that is used as input to a MATLAB script. More details of the setup can be read in the MDEK1001 System User Manual [19]. The measurements from the four anchors; *DW8A0C*, *DW1591*, *DW408F* and *DW4338*, for an example trajectory can be seen below in figure 3.3.

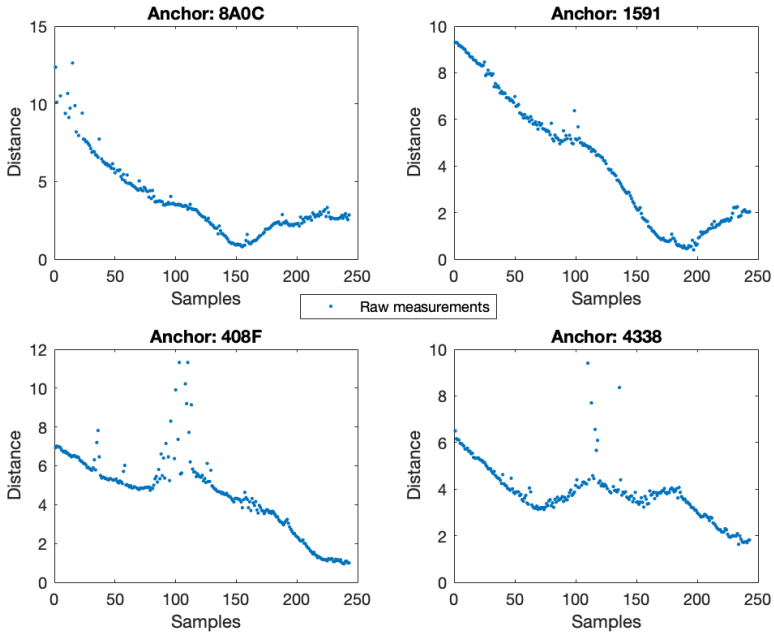


Figure 3.3: Raw measurements from four anchors for an example trajectory.

The measurements suffers from noise which occurs when the signal quality of the anchor decreases. This is often realised as an outlier. A visible example for this

is around samples 90-120 for the anchor 408F. The accuracy of the measurements also decreases heavily when the distance is further than 12 m, which can be seen in figure 3.4, extracted from another trajectory that is moving straight from the rear and outwards from the vehicle.

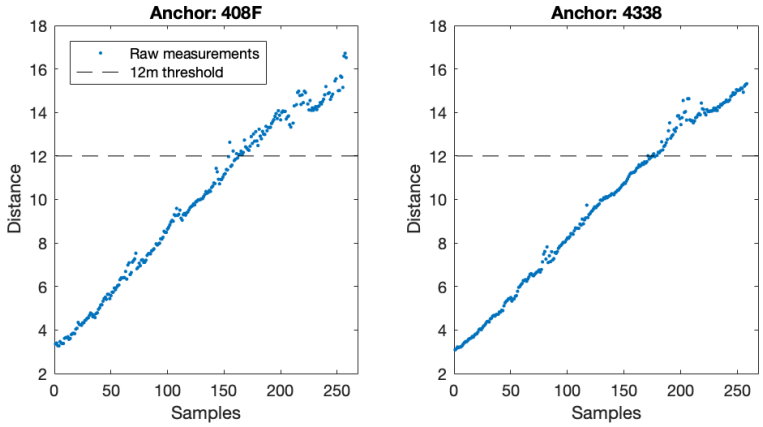


Figure 3.4: Raw measurements from two anchors for moving straight from the vehicle. Showing the difference in accuracy when measurements are over 12m.

The outliers caused by these disturbances heavily influence the trilateration in a negative way, leading to a deterioration in the accuracy of the sensors as the distances grows. This can be assessed by preprocesing the measurements to remove the most obvious outliers.

3.2.1 Outlier removal

The preprocessing of measurements contain three implemented methods for removing outliers. Removing outliers in this context corresponds to replacing the measurement value with a value of NaN (Not a Number).

Method 1: Removing raw anchor measurements that increase with 0.8m, or larger, than the last non-NaN measurement.

Method 2: Removing any raw anchor measurements above 12 m.

Method 3: Removing any trilaterated measurements that lies outside the radius of 10 m from the vehicle.

The first method is implemented to handle measurement outliers caused by lower signal quality, shown earlier in figure 3.3. An increase in the measured distance of 0.8 m between a sample at instance k and $k+1$ with a sample rate of 10 Hz corresponds to a velocity of at least 8 m/s over one sample instance, which classifies it as an outlier. The results from implementing this method on the measurements seen in figure 3.3 can be seen in figure 3.5.

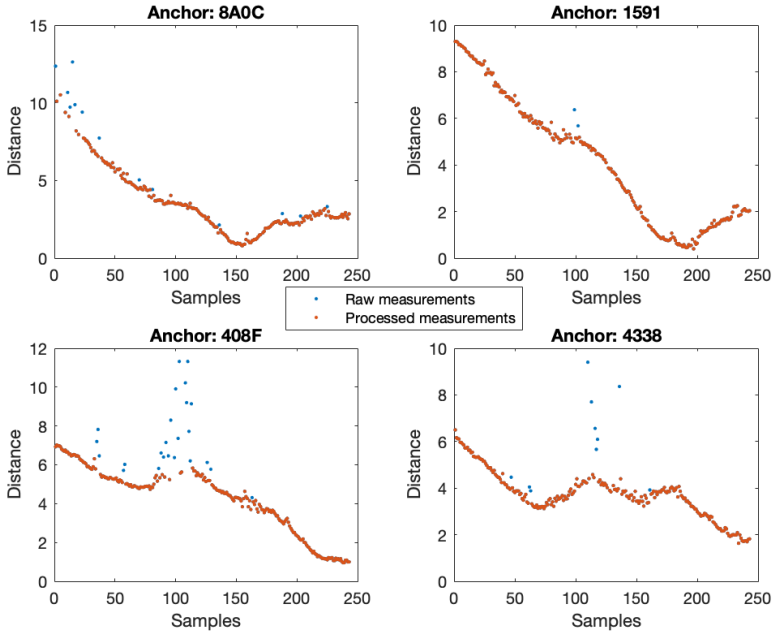
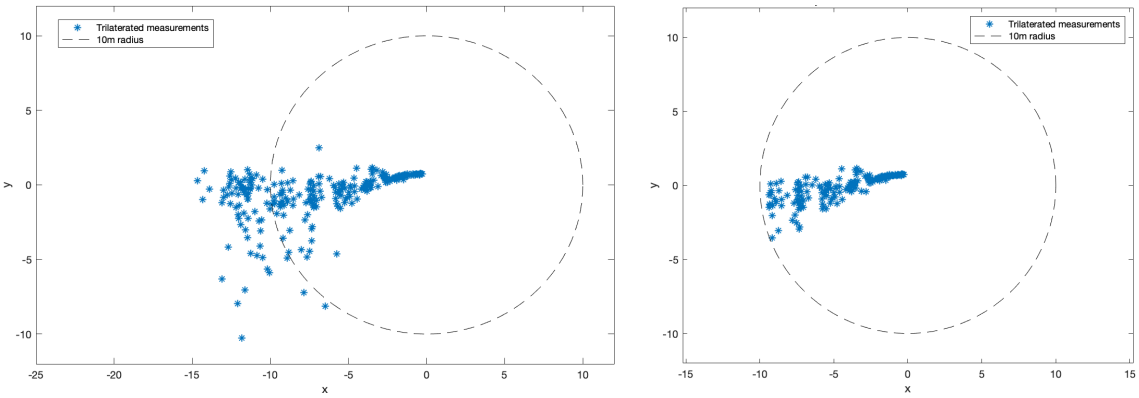


Figure 3.5: Comparison of raw and processed measurements from four anchors, gathered from an example trajectory.

The two remaining methods were implemented to counter the decreasing accuracy for a longer distance, which led to a constraint of only tracking the tag within a 10 m radius from the centre of the vehicle. The measurements for distances further than 10 m from the vehicle are considered too uncertain, resulting in trilaterated measurements that are not feasible for filtering. The three closest anchors are chosen for the trilateration, which further improves the accuracy since a shorter range is more reliable. Optimally, the signal quality of each anchor could be utilized such that the three anchors with best quality are chosen; however, this information is not available through the chosen implementation with the UART shell. The resulting difference of using raw measurements and combining all three methods of preprocessing measurements for the trajectory presented in figure 3.4, can be seen in figures 3.6a, 3.6b.



(a) Raw trilaterated measurements.

(b) Preprocessed trilaterated measurements.

Figure 3.6: Comparison in trilaterated measurements before and after being preprocessed.

It is shown that the preprocessing of measurements removes the outliers both outside the 10 m radius but also in this case some minor outliers within. The preprocessed trilaterated measurements are also shown to inherit a more stable variance that decreases as the distance between the center of the vehicle and the key decreses, thus making it suitable for being the input to the Unscented Kalman Filter of the Trajectory Estimator.

3.3 Trajectory Estimator

The Trajectory Estimator estimates the distance $\hat{d}_k$, a heading $\hat{\gamma}_{k\tau}$, absolute velocity $|\hat{v}_k|$ and angle of arrival $\hat{\varphi}_k$ of the key. As one can see, only every τ:th estimate of $\hat{\gamma}_k$ is used, which will be explained further in section 3.3.2. Here, it is important to distinguish the difference between $\hat{\alpha}_k$ produced by the Unscented Kalman Filter and the heading $\hat{\gamma}_{k\tau}$, where $\hat{\alpha}_k$ is the heading of the key with respect to the axis parallel to the car and that $\hat{\gamma}_{k\tau}$ is the heading with respect to the center of the car. The two angles can be seen in figure 3.7.

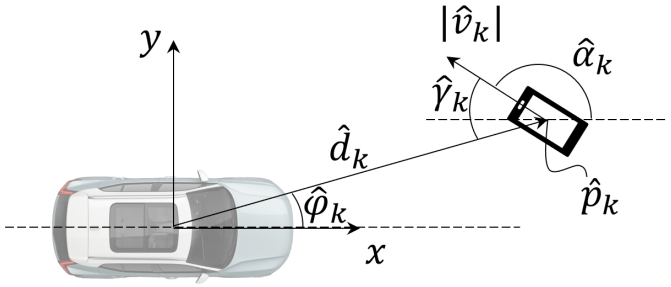


Figure 3.7: Illustration of the kinematic variables of the key relative the center of the car. The key is depicted as a smartphone at position $\hat{p}_k$.

As mentioned earlier, the Driver Intention System relies on the kinematic properties and trajectory of the key, and therefore the output of the TE is of high importance for the functionality of the system as a whole. The TE outputs four states in a vector $\mathbf{x}_k^{(TE)}$ seen in equation 3.1 that are then fed into the IP-algorithm.

$$\mathbf{x}_k^{(TE)} = \begin{bmatrix} \hat{d}_k & \hat{\gamma}_k & |\hat{v}_k| & \hat{\varphi}_k \end{bmatrix}^\top = \begin{bmatrix} \hat{\boldsymbol{\theta}}_k & \hat{\varphi}_k \end{bmatrix}^\top \tag{3.1}$$

Again, the first three state variables in $\mathbf{x}_k^{(TE)}$ is denoted as the vector $\hat{\boldsymbol{\theta}}_k$ and are essential for the loss function of the IP-algorithm. Further on in this thesis, these will be referred as to as the *IP-states*. The relationship between these variables and the loss function will be further covered in section 3.4. The fourth state variable, $\hat{\varphi}_k$, is mainly used to determine if and at which angle of arrival the driver has entered Zone 1, which is further covered in section 3.4.1.

3.3.1 The Unscented Kalman Filter

Exiting the Measurement processing as seen in 3.1, the processed measurements enter the Unscented Kalman Filter (UKF). Here it estimates the state of the key as previously introduced in 2.4, but the TE-states only make use of the Cartesian coordinates, the heading $\hat{\alpha}_k$ and the velocity $\hat{v}_k$ from the filter. To initialise the filter, it requires knowledge of the prior state mean, $\mathbf{x}_0$, and state covariance, $\mathbf{P}_0$ as well as the process noise and measurement noise covariance matrices, $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively. These parameters were set and tuned in the developed simulation tool in MATLAB. This was done by simulating the TE on a generated set of measurements in the form of Cartesian coordinates based on chosen true key trajectories. The output of the TE was later evaluated in terms of RMSE and through visual comparisons between the set of true trajectories and the trajectories produced by the UKF. The parameters for the prior mean and covariance matrix in the UKF were set to

$$\mathbf{x}_0 = \begin{bmatrix} p_{x,0} \\ p_{y,0} \\ 0.5 \\ \alpha_0 \\ 10e^{-9} \end{bmatrix}, \ \mathbf{P}_0 = \begin{bmatrix} 0.2 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 & 0 \\ 0 & 0 & 0 & 10^{-9} & 0 \\ 0 & 0 & 0 & 0 & \sigma_\omega^2 \end{bmatrix} \tag{3.2}$$

For the prior mean, early experiments showed that using a small sequence with length $k = \delta$ of measurement data was necessary in order to define the priors of the Cartesian coordinates $p_{x,0}, p_{y,0}$. When tuning the filter during simulations, a sequence length of $\delta = 10$ was used. This δ was used with the measurements from the UWB-system as well. Therefore, the filter uses the first 10 trilaterated measurements $\tilde{p}_{1:10}$ to calculate the prior state mean variables the following way:

$$p_{x,0} = \frac{1}{\delta} \sum_{i=1}^{\delta} \tilde{p}_{x,i}, \quad p_{y,0} = \frac{1}{\delta} \sum_{i=1}^{\delta} \tilde{p}_{y,i}. \tag{3.3}$$

Practically, this means that the filter does not start the filtering until around 1 second after the TE has started gathering measurements. However, the negative impact

of this was deemed negligible in terms of filter and TE performance. Initialising the key heading prior α_0 was done with the assumption that the initial observation of the key would be that it is facing the car, and therefore done the following way:

$$\alpha_0 = \arctan\left(\frac{-\tilde{p}_{y,1}}{-\tilde{p}_{x,1}}\right). \tag{3.4}$$

Following setting the parameters for the priors, the next step involved setting the parameters σ_v and σ_ω for the process noise covariance matrix $\mathbf{Q}_k$. When constructing a Kalman Filter, the tuning and choice of these parameters is central for the performance of the filter as the uncertainty of the process modelled by $\mathbf{Q}_k$ is commonly the unknown parameter [13]. Therefore, before tuning the matrix $\mathbf{Q}_k$, it is important to first define what is usually known, in this case it was the measurement noise covariance matrix $\mathbf{R}_k$, and specifically the parameters σ_x and σ_y that it contains. Early experiments with the UWB-sensors revealed σ_x and σ_y increasing proportionally to the previously estimated distance $\hat{d}_{k-1}$ by a factor $\lambda = 0.15$ as a reasonable assumption. This means that the measurement noise covariance matrix $\mathbf{R}_k$ was modelled as

$$\mathbf{R}_k = \lambda\hat{d}_{k-1}\begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} = 0.15\hat{d}_{k-1}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.5}$$

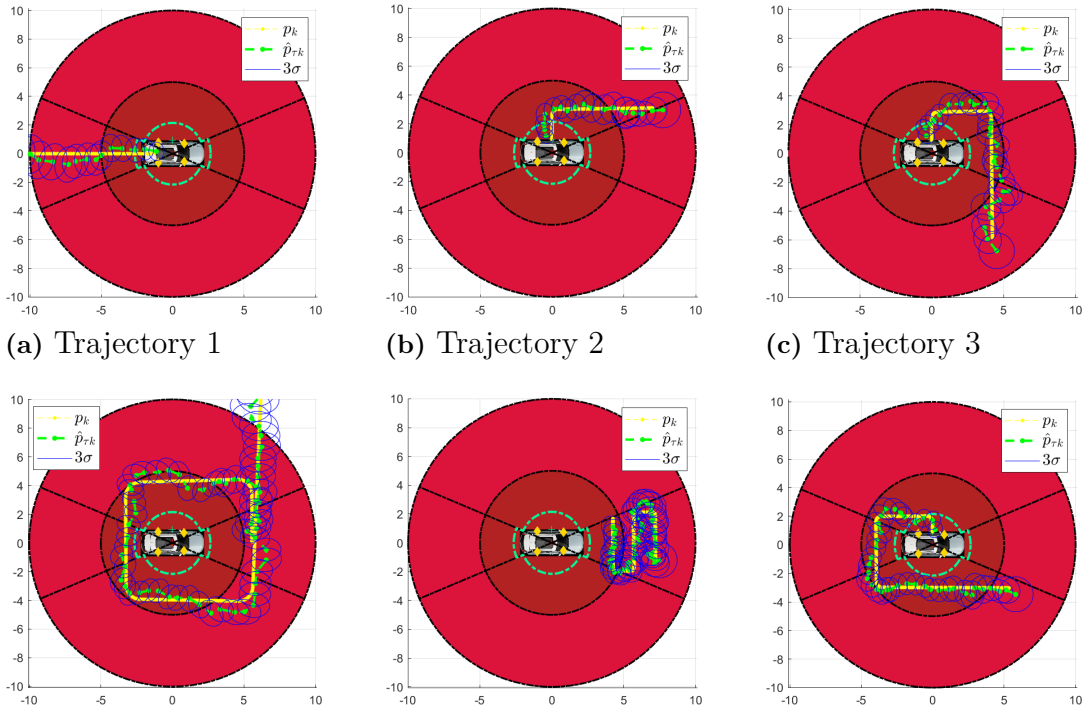
With this measurement noise covariance matrix, the Signal-to-noise-ratio (SNR) will decrease as the distance between the car and the key decreases, leading to the UKF trusting the information from the measurements more as the key comes closer to the vehicle. Contrarily, the UKF will trust its estimates, and thereby the motion model, more the further away the key is from the vehicle. Further trust is also put in the motion model if the measurements from the Measurement processing are set to NaN which sometimes is the case as explained in section 3.2.1. Here, if the UKF receives a NaN as a measurement, no update step is performed and the prediction is set as the estimate.

To tune the process noise covariance matrix $\mathbf{Q}_k$, six different key trajectories close to the vehicle were chosen as example trajectories. These trajectories were chosen with supervision from Volvo Cars as representative trajectories of how the key might move when near the car. The simulated measurements in Cartesian coordinates of the key travelling these trajectories were generated by augmenting the data of the true key trajectories with an additive zero-mean Gaussian noise with the covariance $\mathbf{R}_k$. While tuning the inputs parameters σ_v and σ_ω, each trajectory was simulated over a total of 1000 Monte-Carlo simulations. After each of these 1000 Monte-Carlo simulations, the process noise covariance matrix $\mathbf{Q}_k$ was tuned until a low enough average RMSE of the TE-states over all of the six trajectories had been reached. At the same time, plots of the position estimates were produced together with their 3σ uncertainty regions to visually confirm when most of the true trajectories were within these regions. When most of the true trajectories would be within the uncertainty regions, this would mean that the estimates are following the 99.7-percent rule meaning that the estimates are within that region with a 99.7 % certainty,

indicating a well-performing UKF. The final parameters for σ_v and σ_ω for $\mathbf{Q}_k$ that were used in the UKF are presented in equation 3.6

$$\mathbf{Q}_k = G(\mathbf{x}_k)\,\mathrm{diag}(\sigma_v^2, \sigma_\omega^2)G(\mathbf{x}_k)^\top = G(\mathbf{x}_k)\,\mathrm{diag}(0.35^2, 0.3^2)G(\mathbf{x}_k)^\top \tag{3.6}$$

where G was presented earlier in equation 2.16. These parameters are similar to those chosen in [14] where an Unscented Kalman Filter was used as well; however, in an application of tracking aeroplanes. The position estimates and 3σ-regions produced for test trajectories 1-6 can be seen in figure 3.8.



(a) Trajectory 1 **(b)** Trajectory 2 **(c)** Trajectory 3

(d) Trajectory 4 **(e)** Trajectory 5 **(f)** Trajectory 6

Figure 3.8: True and estimated trajectories together with 3σ-regions generated using simulated measurements.

One can see that much of the true trajectory is within three standard deviations of the estimates, meaning that the estimates are within this region with a 99.7 % probability and that the UKF is therefore performing well with the chosen parameters.

3.3.2 Calculating the TE-state vector $\mathbf{x}_k^{(TE)}$

With the Cartesian coordinates $(\hat{p}_{x,k}, \hat{p}_{y,k})$ calculated by the UKF, the TE-state variables $\hat{d}_k$, $\hat{\gamma}_k$ and $\hat{\varphi}_k$ can be calculated. For the estimated velocity $\hat{v}_k$, only the absolute value is of interest since only the magnitude of the velocity is relevant for the IP-algorithm as the heading $\hat{\gamma}_k$ is treated separately. By setting the center of the vehicle as origin, as previously seen in figure 3.7, $\hat{d}_k$ and $\hat{\varphi}_k$ are computed with

the following equations:

$$\hat{d}_k = \sqrt{\hat{p}_{x,k}^2 + \hat{p}_{y,k}^2} \ , \quad \hat{\varphi}_k = \arctan\left(\frac{\hat{p}_{y,k}}{\hat{p}_{x,k}}\right). \tag{3.7}$$

To calculate the estimated heading of the key with respect to origin, $\hat{\gamma}_k$, the four variables $\hat{p}_{xk}, \hat{p}_{yk}, \hat{\alpha}_k, \hat{d}_k$ are used together with the cosine rule. Figure 3.9 illustrates the necessary components.

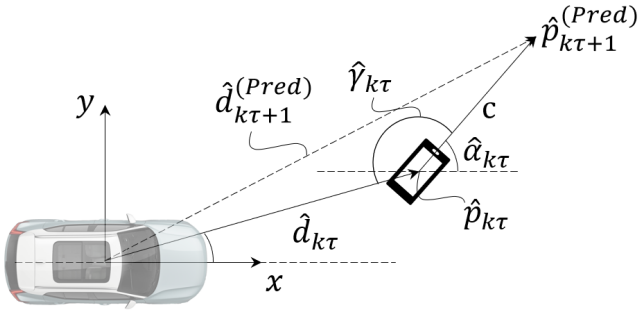


Figure 3.9: Illustration of how the angle $\hat{\gamma}_{k\tau}$ is calculated using the cosine rule.

The heading $\hat{\gamma}_k$ has a large influence on the loss function of the IP-algorithm and large changes in this variable can potentially impact the loss function negatively. Therefore, only every τ:th is used of that variable. This leads to $\hat{\gamma}_{k\tau}$ being calculated in the following way:

1. Using every τ:th estimate of the variable $\hat{\alpha}_{k\tau}$ and the Cartesian coordinate estimates $\hat{p}_{x,k\tau}, \hat{p}_{y,k\tau}$, predict the "virtual" Cartesian coordinates $\hat{p}_{x,k\tau+1}^{(Pred)}, \hat{p}_{y,k\tau+1}^{(Pred)}$ at the next sample instance, regardless the velocity:

$$\hat{p}_{x,k\tau+1}^{(Pred)} = \hat{p}_{x,k\tau} + cos(\hat{\alpha}_{k\tau}) \tag{3.8}$$
$$\hat{p}_{y,k\tau+1}^{(Pred)} = \hat{p}_{y,k\tau} + sin(\hat{\alpha}_{k\tau}) \tag{3.9}$$

2. Calculate the predicted absolute traveled distance c between the two Cartesian points $\hat{p_{k\tau}}$ and $\hat{p}_{k\tau+1}^{(Pred)}$ using Pythagoras theorem

$$c = \sqrt{(\hat{p}_{x,k\tau+1}^{(Pred)} - \hat{p}_{x,k\tau})^2 + (\hat{p}_{y,k\tau+1}^{(Pred)} - \hat{p}_{y,k\tau})^2} \tag{3.10}$$

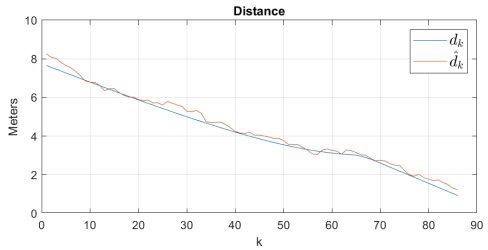
3. Calculate the "virtual" distance $\hat{d}_{k\tau+1}^{(Pred)}$ to the origin at the next instance

$$\hat{d}_{k\tau+1}^{(Pred)} = \sqrt{\left(\hat{p}_{x,k\tau+1}^{(Pred)}\right)^2 + \left(\hat{p}_{y,k\tau+1}^{(Pred)}\right)^2} \tag{3.11}$$

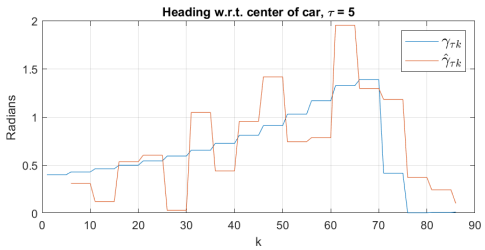
4. Calculate the estimated heading of the key at sample instance $k\tau$ w.r.t. origin using the cosine rule

$$\hat{\gamma}_{k\tau} = \arccos\left(\frac{\left(\hat{d}_{k\tau+1}^{(Pred)}\right)^2 - \hat{d}_{k\tau}^2 - c^2}{-2(\hat{d}_{k\tau}c)}\right) \tag{3.12}$$

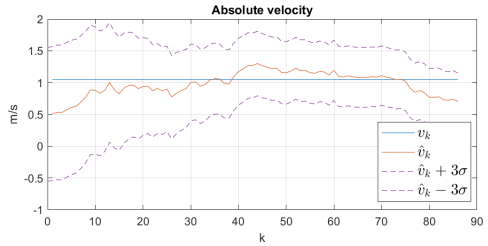
The selection of τ causes a delay and needs to be chosen carefully in order to avoid a too large delay for real-time systems. For the Driver Intention System with a sample rate of 10Hz, τ was set to 5 samples, which corresponds to a 0.5 seconds delay. This was considered as acceptable with supervision from Volvo Cars. The TE-states produced with the set parameters for the UKF can be seen in figure 3.10c where the estimations for the velocity is also within the 3σ regions which further implies that the filter is performing well.



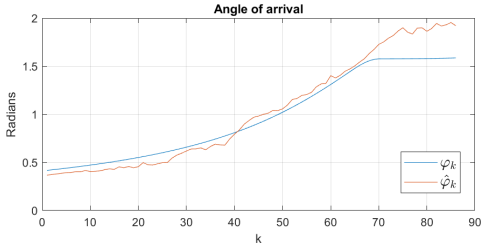
(a) True and estimated distance



(b) True and estimated heading w.r.t. center of car



(c) True and estimated velocity with 3σ estimates.



(d) True and estimated angle of arrival

Figure 3.10: Comparison of TE-states for true trajectory and estimated trajectory. The comparison is made with Trajectory 2 shown in figure 3.8b

With most of the true trajectories being within the 3σ regions of the estimates, the filter is performing adequately with the chosen parameters for $\mathbf{Q}_k$ and $\mathbf{R}_k$. The RMSE values for each of the TE-states of all the trajectories averaged over 1000 Monte-Carlo simulations can be seen in table 3.3.

Table 3.3: RMSE values for each of the TE-states from a run of 1000 Monte-Carlo simulations. Each sequence has the length N with a sample time of $T = 0.1$ s.

	Traj. 1	**Traj. 2**	**Traj. 3**	**Traj. 4**	**Traj. 5**	**Traj. 6**
N	296	86	133	379	253	189
d_{RMSE}	0.543	0.413	0.396	0.503	0.374	0.362
γ_{RMSE}	0.586	0.476	0.488	0.457	0.731	0.510
v_{RMSE}	0.249	0.226	0.236	0.261	0.245	0.173
$\cos(\varphi_{RMSE})$	0.003	0.100	0.068	0.041	0.019	0.062

Due to the switch between π and $-\pi$ for the angle of arrival φ_k as the key passes the x-axis to the rear of the car, occurring for trajectory 1, 4 and 6, and together

with how sensitive RMSE is to large changes, the cosine for φ_k was used instead to get a fair numeric of the error, since $\cos(\pi) = \cos(-\pi)$. The RMSE values in 3.3 served as benchmark values for the TE when further on using measurements from the UWB system.

3.4 Intention Predictor

The next function block in the flow chart, seen in figure 3.1, is the Intention Predictor algorithm which, described on a high level, takes the TE-states, $\mathbf{x}_k^{(TE)}$, as input and converts them into the actions a_i, with $i \in \{1, 2, 3\}$, that activates/deactivates zone functionality. These zone functionalities could for instance be those listed in table 2.1. The IP-algorithm includes a decision-making method, presented briefly in chapter 2.5, where an action is decided from the results of a loss function that depends on the information from an information space.

3.4.1 Loss function

The purpose of the loss function in the IP algorithm is to evaluate the intention of the driver in a metric value, where a low expected loss value corresponds to a higher intention of using the vehicle. The inputs to the loss function are the TE-states, which are deterministic values of the estimated mean of the distance, the heading relative the center of the vehicle, and the velocity.

To form the loss function, some conditions of the TE states have to be defined which either increase the loss value or decrease it. The loss value for the intention of the driver should decrease for a lower distance to the vehicle, a heading angle relative to the vehicle center, and for a higher velocity towards the vehicle. Additionally, the loss value should increase if the velocity is high for a heading that is facing away from the vehicle, as previously introduced in table 3.2 in section 3.1.1. These conditions together with the weight factors, $\kappa_1, \kappa_2, \kappa_3$, defining the relation between the IP-states, form the choice of the loss function for the IP-algorithm, which is presented in equation (3.13). The variables with "(n)" as superscript correspond to normalized values, which is further described in section 3.4.3.

$$\mathcal{L}(\boldsymbol{\theta}_k) = \kappa_1 \hat{d}_k^{(n)} + \kappa_2 \sin(\hat{\gamma}_k^{(n)}) - \kappa_3 \cos(\hat{\gamma}_k^{(n)}) \hat{v}_k^{(n)} \tag{3.13}$$

The design choice of incorporating deterministic values to the loss function, rather than probability distributions, provides a deterministic output from the loss function. In return, this simplified the process of tuning the weight factors such that a desired behaviour of the loss function could be achieved.

3.4.2 Decision function

The biggest difference from the conventional method of Bayesian Decision Theory, where the loss value for each action is calculated, is that the chosen loss function returns a single loss value regardless of the available actions. To adapt to

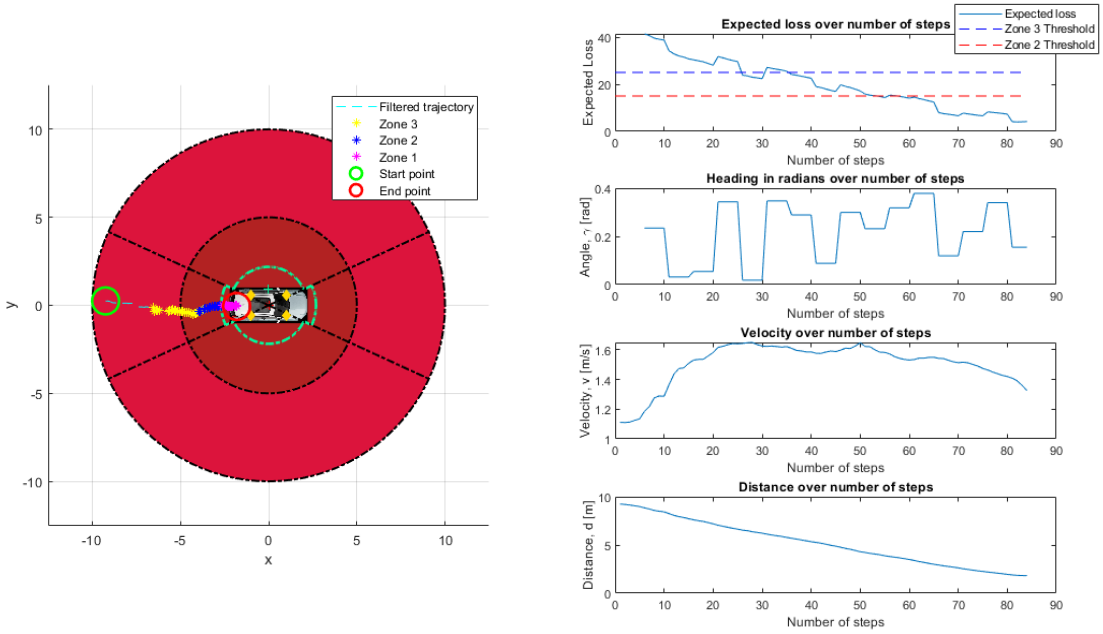
this difference, a decision function was implemented which determines if the intention, expressed as the loss value, is high enough to enter Zone 3 or Zone 2. As mentioned in chapter 2.5, the zone functionalities for Zone 2 and Zone 3 are activated/deactivated if the loss value passes the corresponding threshold set for each zone respectively. The initial conditions for these thresholds were arbitrarily set to $\tau_{z_2} = 15$ and $\tau_{z_3} = 25$. The magnitude of the thresholds are co-dependent to the magnitude of the weight factors, thus a decision was made to only focus on tuning the weight factors to match with an arbitrary choice of zone thresholds. The thresholds for Zone 2 and Zone 3 were also designed with the ability to dynamically change throughout the monitoring process, and are further explained in section 3.4.4.

The reason for not declaring a loss value threshold for triggering Zone 1 is due to security reasons constrained by Volvo Cars, since this zone includes the functionality of unlocking the car. This led to the decision of only activating Zone 1 by evaluating the measured distance between the key and the vehicle. The innermost static zone, seen in figure 3.11a with a green border, corresponds to the threshold for activating Zone 1. This zone is made up by four sub-zones that are separated by the estimated angle of arrival, $\hat{\varphi}_k$. The zone in the rear and in front of the vehicle is set with the same threshold distance, and a shorter distance is used for the driver's side and passenger side. Each sub-zone unlocks the vehicle and also introduces the possibility of unlocking only the doors in that sub-zone. Table 3.4 describes the setting for Zone 1, where the threshold distances depend on geometrical properties of the vehicle. In this case, the length to the corner of the vehicle from its center-point, $l_c = 2.400$ m. The angles used for each zone can be seen directly in figure 3.11a, and are computed from the x-axis, going through the center of the vehicle, to each corner, $\hat{\varphi}_c = 0.4309$ rad.

Table 3.4: Settings for the sub-zones of Zone 1

Subzone	**Angle of arrival** [rad]	**Threshold distance** [m]
Front	$-0.4309 < \hat{\varphi}_k < 0.4309$	$l_c + 0.25$
Driver's side	$0.4309 < \hat{\varphi}_k < 2.7107$	$l_c - 0.25$
Passenger side	$-2.7107 < \hat{\varphi}_k < -0.4309$	$l_c - 0.25$
Rear	$2.7107 < \hat{\varphi}_k$ OR $\hat{\varphi}_k < -2.7107$	$l_c + 0.25$

A visualisation of the complete system in action, including the IP states and indications of zone triggers for an estimated trajectory can be seen in figure 3.11. Here, the estimated trajectory of the key approaching the vehicle directly is shown together with the corresponding loss value during that trajectory with $\kappa_1 = 50, \kappa_2 = 27, \kappa_3 = 50$. An indication for when each zone is triggered can be seen in the left figure, where positions that cross Zone 3 are marked with yellow, Zone 2 with blue, and Zone 1 with pink. Once a zone has been triggered, the functions for that zone are active until the loss has been higher than the respective threshold for a continuous time of 5 s. This feature makes sure that the zones are not too frequently deactivated, and is further explained in section 3.4.4.



(a) Trajectory and zones being triggered **(b)** Corresponding loss value and IP states

Figure 3.11: An illustration of a scenario where the driver is walking towards the vehicle, and the corresponding IP states and expected loss.

3.4.3 Parameter tuning

As previously mentioned, by setting arbitrary values to the zone thresholds, the process of tuning the IP algorithm can be simplified to focus on only tuning the weight factors. To further simplify the process of tuning the IP algorithm, the IP-states was decided to be normalized linearly. By normalizing the IP-states, the weight factors can be set to comparable values, and the heading angle can be constrained to only yield positive values when used with sine and cosine. The target heading relative the center of the vehicle can only take values of $\hat{\gamma} \in [0, \pi]$, the maximum distance was set to 10 m, according to the outermost radius seen in figure 3.11a, and the likely maximum velocity was set to 5 m/s, equivalent to an average running speed. The conversion of these states can be seen below.

$$\hat{d} \in [0, 10] : \hat{d}^{(n)} = \frac{\hat{d}}{10} \rightarrow \hat{d}^{(n)} \in [0, 1] \tag{3.14}$$

$$\hat{\gamma} \in [0, \pi] : \hat{\gamma}^{(n)} = \frac{\hat{\gamma}}{2} \rightarrow \hat{\gamma}^{(n)} \in [0, \frac{\pi}{2}] \tag{3.15}$$

$$\hat{v} \in [0, 5] : \hat{v}^{(n)} = \frac{\hat{v}}{5} \rightarrow \hat{v}^{(n)} \in [0, 1] \tag{3.16}$$

This choice of loss function allows the three different factors, κ_1, κ_2 and κ_3, for tuning. The weight for the distance between the key and the vehicle is tuned by κ_1,

the weight of the key heading is tuned by κ_2, and the weight of the key velocity depending on the target heading is tuned by κ_3. By taking the sine-component of the normalized heading angle, the additive loss value of the heading angle increases exponentially for non-normalized angles between 0 and $\frac{\pi}{2}$. The same can be seen for the last term in the loss function, which includes the negative cosine of the normalized heading angle. This was implemented to ensure that heading angles close to zero affect the loss similarly to an angle of zero, but also to penalize higher angles, corresponding to the key facing away from the vehicle, exponentially.

In order to tune the weight factors, five different conditions with desired constraints were formed in order to tune one factor at a time. Note that this tuning is made for the default zone thresholds of $\tau_{z2} = 15$ and $\tau_{z3} = 25$, where lower thresholds will yield different constraints which is further covered in section 3.4.4. By setting the heading and velocity with constant values, it was possible to evaluate when, and at what distance, the zones were triggered. It was then possible to tune the weight factors to increase or decrease the influence of a state and thus find a setting that fulfills all the set conditions. The five conditions are presented below in table 3.5. Note that the blank threshold values in table 3.5, noted as "-", means that the zone should never be triggered during these conditions, even for very short range.

Table 3.5: The setting of the five conditions and their corresponding desired distances for triggering zone 2 and zone 3.

Condition	Heading	Velocity	Zone 3 triggered	Zone 2 triggered
1	$\hat{\gamma}_k = 0$	$v = 1.5$	6.5m	4.5m
2	$\hat{\gamma}_k = 0$	$v = 0.5$	5.5m	3.5m
3	$\hat{\gamma}_k = \pi/2$	$v = 1.5$	-	-
4	$\hat{\gamma}_k = \pi/4$	$v = 1.5$	4.5m	-
5	$\hat{\gamma}_k = \pi/8$	$v = 1.5$	5.5m	3.5m

The five conditions were decided accordingly for different settings of the heading and velocity in order to tune the weight factors. The goal was to find a tuning where both zones should trigger at longer range for conditions of low heading angle and high velocity, and avoid false positives for opposite conditions. To begin tuning, the conditions 1 and 2 were used, where the goal was to tune the weight factors κ_1 and κ_3, since the heading was set to zero and did not influence the loss. Secondly, the conditions $3, 4, 5$ were set to tune the last weight factor, κ_2. An example of how the thresholds were found for each condition can be seen in figure 3.12, showing the fixed setting of the first condition.

This process was performed for all five conditions which gave the results seen in table 3.6. The parameter tuning that were found and used for these results is, $\kappa_1 = 50, \kappa_2 = 27, \kappa_3 = 50$. The results show that the loss function is sufficiently tuned for the desired thresholds, and these parameters were used throughout the thesis.

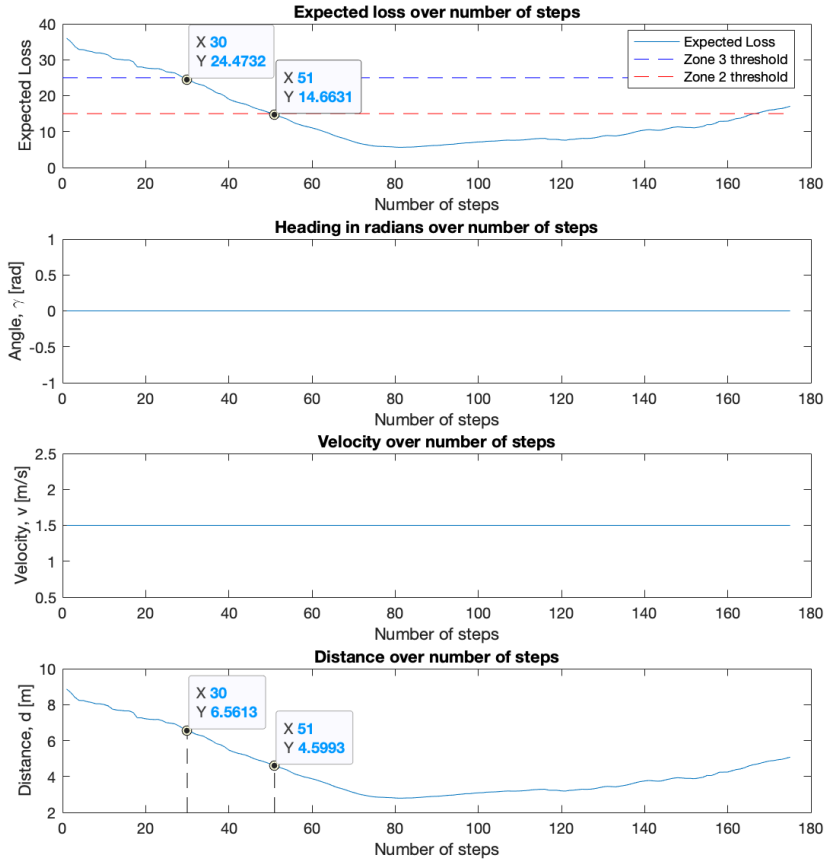


Figure 3.12: Loss over time with the constraints of condition 1, $\hat{\gamma}_k = 0, \hat{v}_k = 1.5$. The distance threshold for triggering zones was shown to be 6.561 m for Zone 3 and 4.599 m for Zone 2.

Table 3.6: Comparison of desired thresholds and actual thresholds from the loss function, made for the five conditions stated in table 3.5.

Condition	Desired thresholds (Z3/Z2)	Actual thresholds (Z3/Z2)
1	6.5 m/4.5 m	6.561 m/4.599 m
2	5.5 m/3.5 m	5.470 m/3.483 m
3	-/-	-/-
4	4.5 m/-	4.440 m/-
5	5.5 m/3.5 m	5.470 m/3.580 m

3.4.4 Dynamic thresholds for zone activation

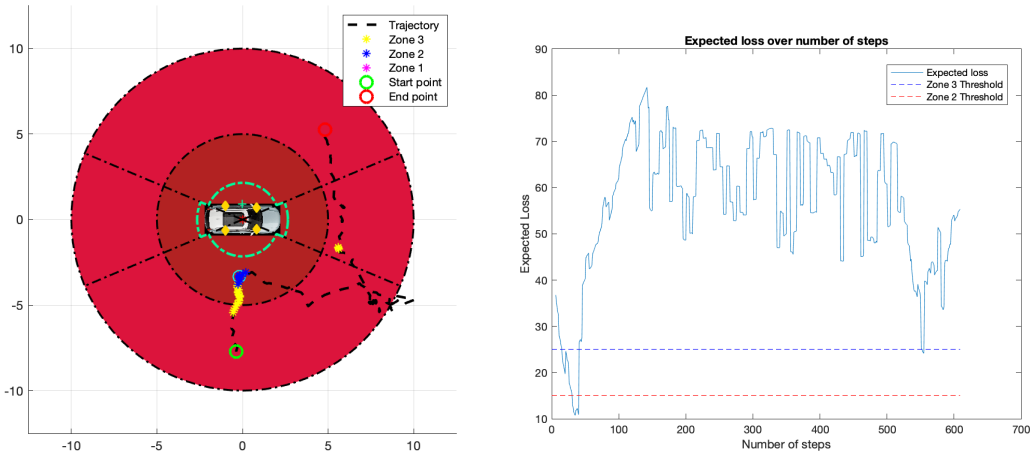
With the loss function introduced, a way to describe the real-time loss, and therefore intention, is now in place. However, the issue with false positives still remains. In this thesis, this has foremost been addressed by introducing dynamic thresholds

for zone activation. The purpose of including dynamic thresholds for activating the zones is to circumvent the scenarios similar to those that would occur with static zones, namely that the zone triggering can be abused to repetitively occurring. Not only does this translate into a frequent occurrence of false positives, but also scenarios that consume battery power and can appear as frustrating or confusing to the user. The chosen solution for this problem was to implement a dynamic threshold for the IP-algorithm that lower the threshold value after a zone has been triggered and then exited without reaching the vehicle. After leaving a zone, a set timeout interval of Ψ keeps the functions for that zone active before turning them off, and then decreases the threshold. Following this, the criteria for triggering that zone the next time the key is observed become stricter, which has been deemed as one way to decrease the number of false positives.

The way that the system decreases the threshold is linked to the maximum number of false positives that is allowed to be observed before the zones becomes deactivated. In this thesis, the threshold of a zone is decreased with a loss value of 5 for each occurrence, leading to a maximum number of allowed false positives for each of the zones to be effectively five times for Zone 3 and three times for Zone 2. To keep track of how long each zone has been active, a counter for each zone is implemented which increments for each measurement made. The number of times the counter need to be incremented, η_{z_i}, before deactivating the respective zone is dependent on the sampling time, T. This is summarised in the following formula:

$$\eta_{z_i} = T \text{ (samples/s)} \times \Psi(s) \tag{3.17}$$

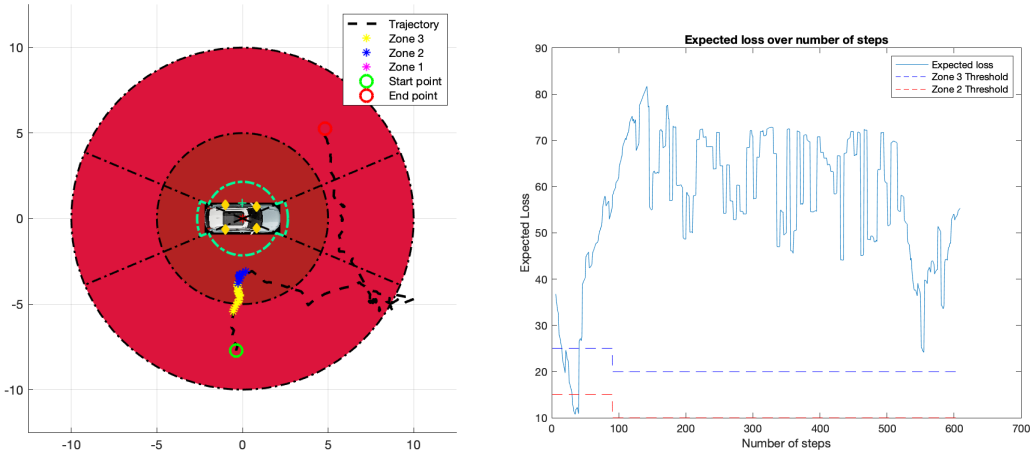
To illustrate this solution, a scenario that is triggering a zone twice without and with dynamic threshold, is shown respectively in figure 3.13 and figure 3.14. In this scenario, the driver approaches the vehicle, turns away, stands still for 30 seconds and then moves again.



(a) Trajectory with zone trigger identifications

(b) Loss value over time, $\Psi = 0s$

Figure 3.13: Trajectory and loss value without the use of dynamic threshold.



(a) Trajectory with zone trigger identifications

(b) Loss value over time, $\Psi = 5s$

Figure 3.14: Trajectory and loss value with the use of dynamic threshold.

When the driver reaches Zone 1 and enters the car, both of the thresholds will be reset to their initial values. Another way implemented to reset the threshold was the use of a timer, where a certain time duration without any observations of the key made resets both thresholds to their initial values. The choice of Ψ is rather subjective and was settled upon together with supervision and insight from Volvo Cars to 50 consecutive samples (5 seconds) for Zone 2 and Zone 3.

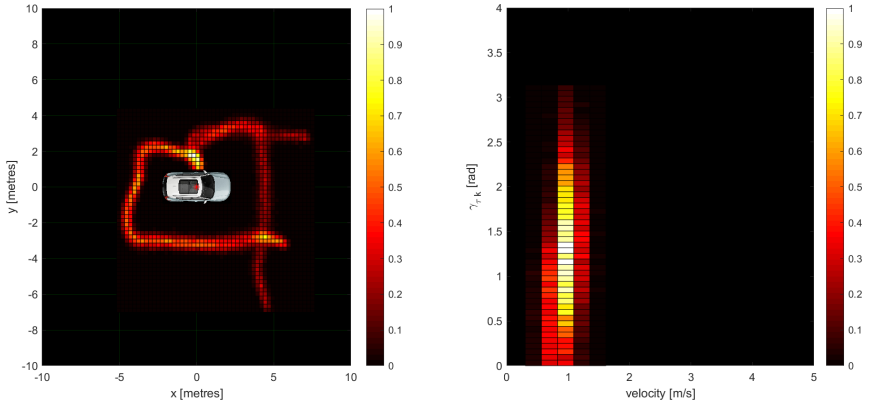
3.4.5 Concept of modelling driver usage patterns

Other than the dynamic threshold controlled by time related variables, there is also the possibility for the IP-algorithm to set the threshold values τ_{z2} and τ_{z3} by modelling the usage patterns of the car in terms of the trajectories, headings and velocities estimated by the UKF. The idea of this functionality is that it eventually "learns" the most common attributes concerning the key and the driver that lead to the driver using the vehicle, and uses this for future driver intention predictions. This is practically done by storing what trajectories $\hat{p}_{1:N}$, key headings $\hat{\gamma}_{1:N}$ and key velocities $\hat{v}_{1:N}$ that were estimated using the sensor observations during a sequence with length N that lead to the driver using the vehicle at the current ε-parameters, $\boldsymbol{\varepsilon}_k = [\varepsilon^{\text{GPS}} \ \varepsilon^{\text{time}}]^\top$, corresponding to the GPS position of the car and the time of the day respectively.

The factor that determines if the driver uses the vehicle would for example be the driver grabbing the door handle. As the IP-algorithm has the movement history of the key near the vehicle stored in its memory, the IP-algorithm would then evaluate if a new observation corresponds to any of the observations in its memory and use that information to determine the intention of the driver by setting adequate threshold values. This feature recognition-like functionality, where the features are the usage patterns of the vehicle by the driver, is believed to be most of use in the event that the real-time loss computed by the loss function is low, and thus the immediate probability of the driver using the vehicle is considered as high, but

the algorithm memory at the current $\boldsymbol{\varepsilon}$-parameters deeming the probability of the driver using the vehicle as low. This could for instance be the driver passing the car at short range, which would yield a low loss, yet during a time and GPS position where it has never been observed that the driver has ended up using the vehicle. This would then lead to the overall probability of the driver using the vehicle as low, and thus the threshold values of τ_{z2} and τ_{z3} would be set to low values ultimately leading to a false positive being avoided.

As a conceptual example, the memory has been divided into two parts, M_1 and M_2, and are illustrated as "heat maps" being matrices where each element ranges between the value 0 (cold, black) and 1 (hot, white). One can also consider them as 2D-histograms where how often an element occurs sets its value, i.e. a certain element is estimated often, then that element will approach the value 1. Memory M_1 is related to the position of the key in Cartesian coordinates at a certain GPS position ε^{GPS} and time of the day $\varepsilon^{\text{time}}$ that lead to the vehicle being used. Memory M_2 models $\hat{\gamma}_k$ as a function of the velocity $\hat{v}_k$ that has been observed leading to the driver using the vehicle, at the same GPS position and time of the day as M_1. Due to this system requiring the memorisation of possibly hundreds or thousands of trajectories, the time constraints of this thesis has lead to this system being only on a conceptual stage with a simulated memory. This has been considered as enough by Volvo Cars since they only requested research within this area. The resolution of each element has for these examples been 0.2×0.2 metres for the first memory and approximately 0.25 m/s $\times$ 0.5 radians for the second memory. The values for ε^{GPS} and $\varepsilon^{\text{time}}$ are not explicitly chosen, but could for example be a GPS-position with a radius of 10 metres and a time-of-the-day-interval between $08:00 - 08:15$. How these "heat maps" for each memory may look like can be seen in figure 3.15.



(a) Heatmap of M_1 with commonly estimated positions in Cartesian coordinates $\hat{p}_k$

(b) Heatmap of M_2 with commonly estimated key headings $\hat{\gamma}_k$ and key velocities $\hat{v}_k$

Figure 3.15: Examples of IP-algorithm memories M_1 and M_2 containing commonly estimated key trajectories and headings w.r.t. vehicle center as a function of key velocity at a specific time $\varepsilon_k^{\text{time}}$ and GPS position $\varepsilon_k^{\text{GPS}}$ that lead to the driver using the vehicle.

It is believed that this functionality can be extended with further parameters within the same category as GPS and time of the day to additionally connect to the idea of the vehicle adapting to the driver. This could for example be calendar data where appointments contain location data, map data containing travel history or saved locations and activity of using Home Assistant Systems such as Google Home.

4

Experimental Setup

The following chapter contains the experimental setups made for the Driver Intention System, including the choice of sensor placement and the formulation of evaluation tests. The evaluation tests were set up in order to answer the research questions of this thesis.

4.1 UWB anchor setups in a test vehicle

By setting up the UWB sensors in a test vehicle, the performance of the Driver Intention System using real life observations could be studied. A total of four sensors were used as anchors and a fifth sensor as a tag. In order to find an optimal placement of the anchors, alternative anchor placements were tested. This was evaluated in terms of measurement quality, by studying how the UWB sensors cope with the line of sight being obstructed by different materials of the vehicle. Thus, identifying areas of poor measurement quality around the vehicle, where the measured signal quality for the UWB sensors drops, and also identifying how well the measurements were estimated when switching between anchors during trilateration. The initial anchor setup was settled together with Volvo Cars, which is similar to an anchor setup they used in a previous test with a Volvo V60 test vehicle. This setup was chosen as a starting point which could later be modified. The anchors were placed inside of the vehicle and as close to the vehicle chassis as possible in order to mimic a future potential placement. In this thesis, all the tested anchor setups include a Volvo XC40, and the first setup, anchor setup 1, which has the anchors mounted inside of the vehicle close to the A- and C-pillars can be seen in figure 4.1 below.

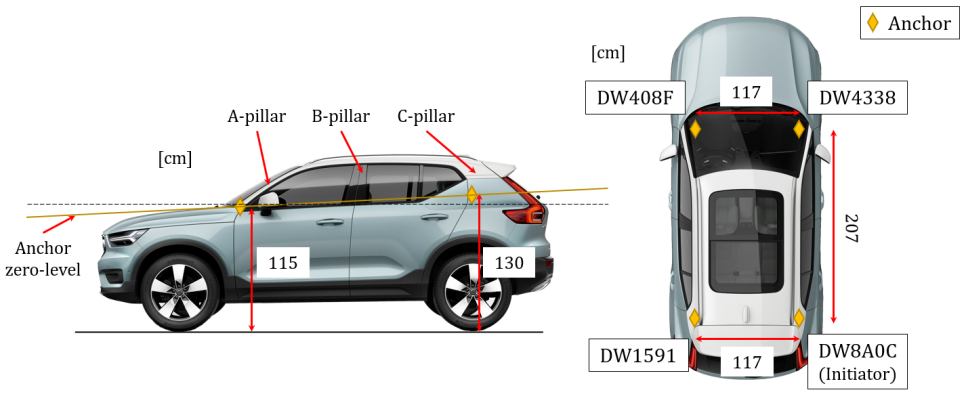


Figure 4.1: Anchor setup 1 in a Volvo XC40

After acquiring measurements using this setup, a second anchor setup was tested. This setup involved moving the two sensors in the rear from the C-pillar to the area above the door handles, seen in figure 4.2.

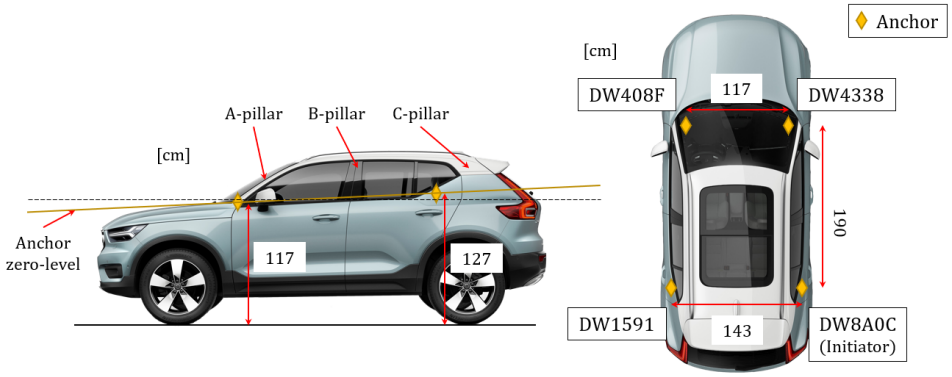


Figure 4.2: Anchor setup 2 in a Volvo XC40

A few other anchor placements were investigated as well but they will not be covered in this section since the two anchor placements above produced the overall best measurements. The method used to investigate the different anchor placements was to move the target along the trajectories 1a-1d that can be seen in figure 4.3.

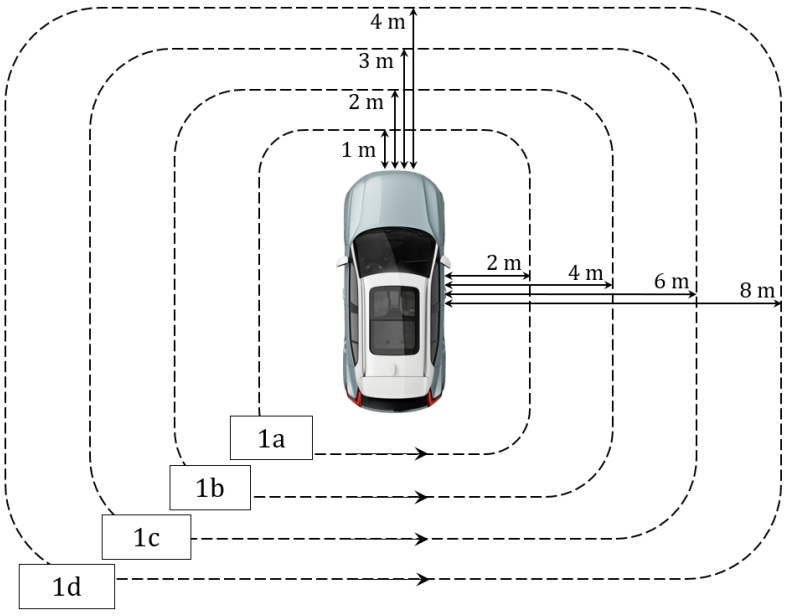
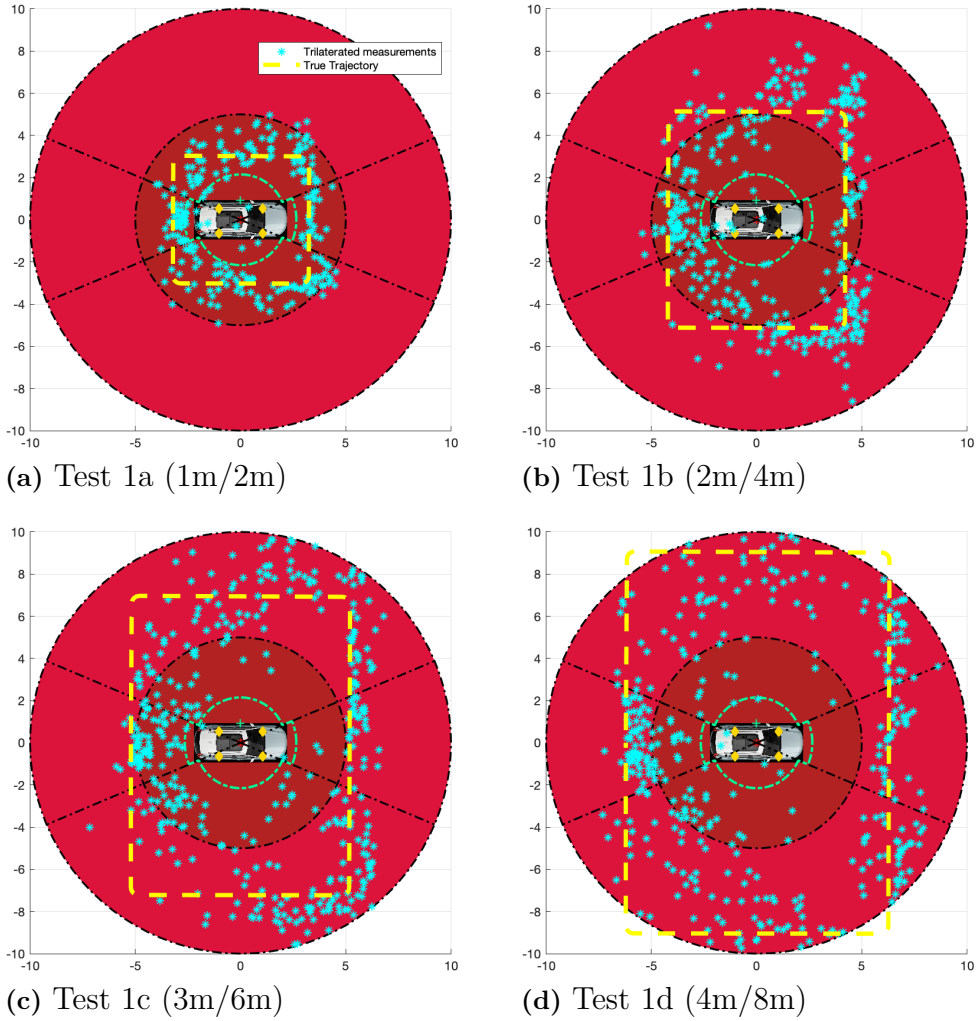


Figure 4.3: Dimensions of the trajectories used to evaluate different anchor placements.

These trajectories were settled as a sufficient strategy to ensure utilisation of all four anchors and how the different combination of used anchors performs at any angle of arrival at multiple distances. Each of the trajectory started at the rear of the vehicle with the direction depicted by the black arrows in figure 4.3.

4.1.1 Comparing anchor setups for optimal placement

This section compares the resulting trilaterated measurements from both anchor setups for the trajectories 1a-1d. The preprocessed trilaterated measurements and the true trajectory for the first anchor setup can be seen in figure 4.4.



(a) Test 1a (1m/2m)

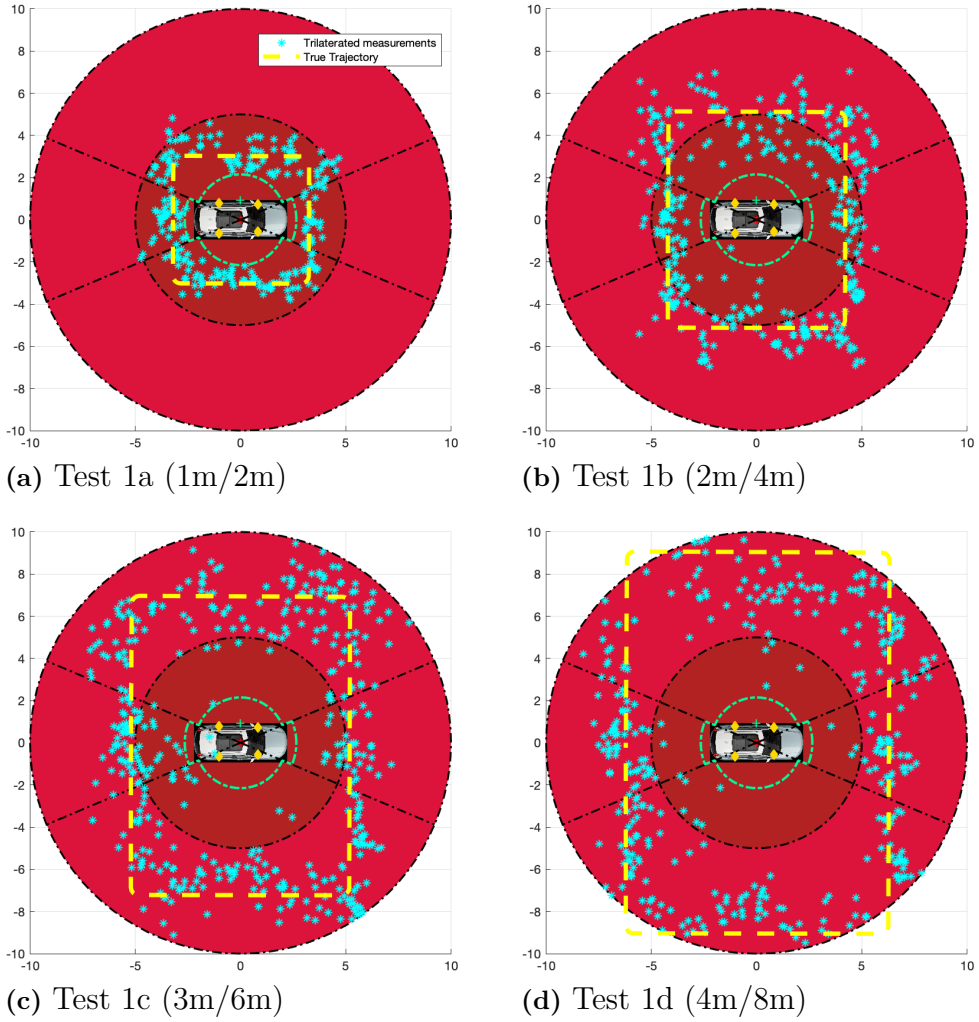
(b) Test 1b (2m/4m)

(c) Test 1c (3m/6m)

(d) Test 1d (4m/8m)

Figure 4.4: Preprocessed trilaterated measurements and true trajectory for anchor setup 1.

It can be seen that the measurements in the front of the vehicle are fairly accurate, as they are closer to the true trajectory, but are worse to the rear of the vehicle. There are even some outliers that are several meters off, as can be seen in figure 4.4b, such as some of the measured positions being inside the vehicle. The measurements to the rear of the vehicle are also shown to be further misplaced from the true trajectory as the distance from the vehicle increases. This is assumed to be partially caused by the C-pillar being a potential poor-quality-zone as it is an area containing several layers of different materials. Moving over to anchor setup 2, the preprocessed trilaterated measurements and the true trajectory for that setup can be seen in figure 4.5.



(a) Test 1a (1m/2m)

(b) Test 1b (2m/4m)

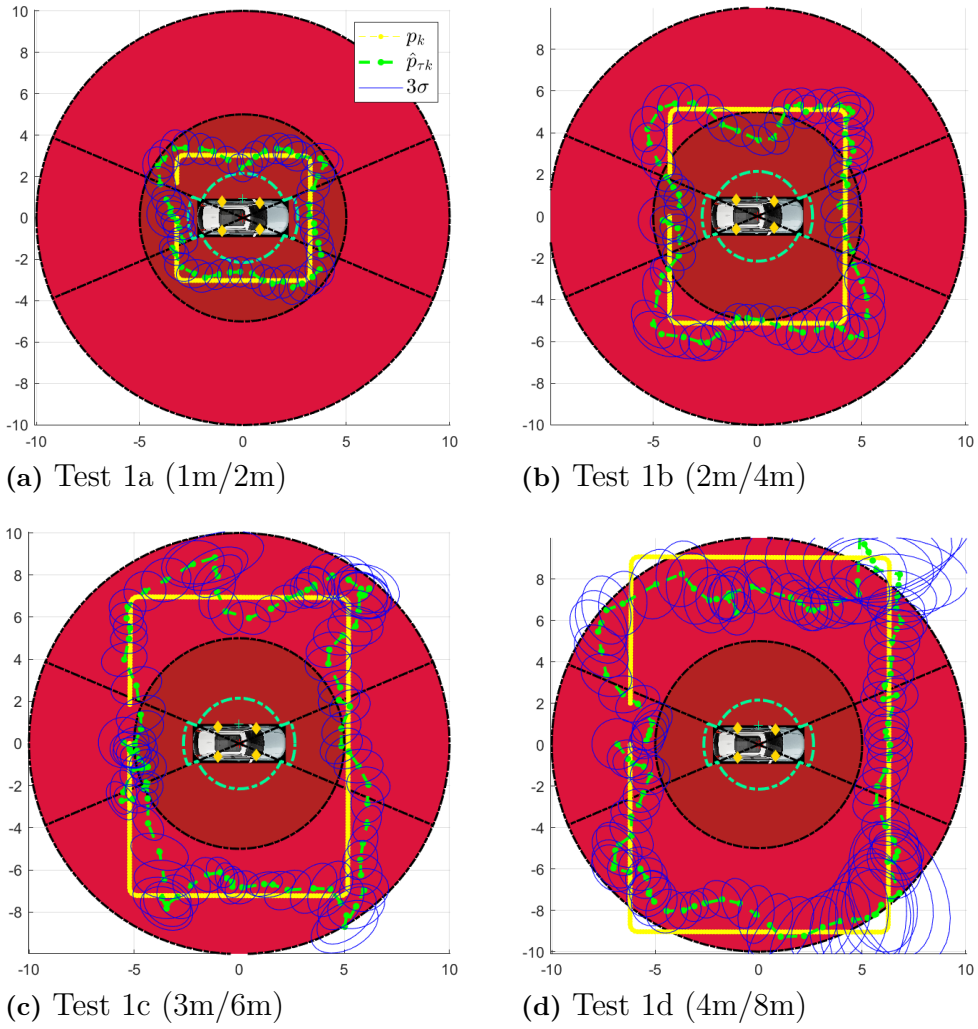
(c) Test 1c (3m/6m)

(d) Test 1d (4m/8m)

Figure 4.5: Preprocessed trilaterated measurements and true trajectory for anchor setup 2.

Here, it is clear that anchor setup 2 performs better as the measurements are closer to the true trajectory throughout the whole measurement sequence. This further strengthens the suggestion that the C-pillar is a potential zone for poor quality of the measurements and that the quality of the measurements deteriorates when the line of sight is obstructed by several layers of material. However, one issue still remains, being that the error between the measured positions and true positions become larger as the distance increases. This was however expected, as well as the occurrence of outliers, yet the magnitude of the errors seemed to be lower for anchor setup 2, especially to the rear of the vehicle, being the most important advantage in this case. The performance for anchor setup 2 was therefore overall considered as better, and will further on in this thesis be referred as the final anchor setup.

Filtering the measurements gathered from test trajectories 1a-d produced the estimated trajectories seen in figure 4.6.



(a) Test 1a (1m/2m)

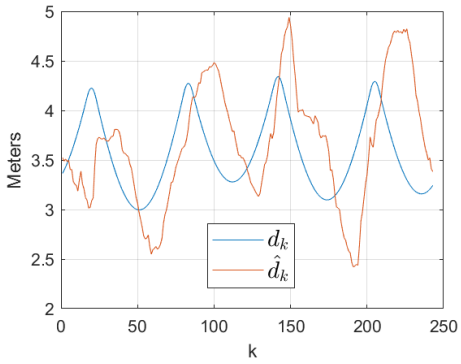
(b) Test 1b (2m/4m)

(c) Test 1c (3m/6m)

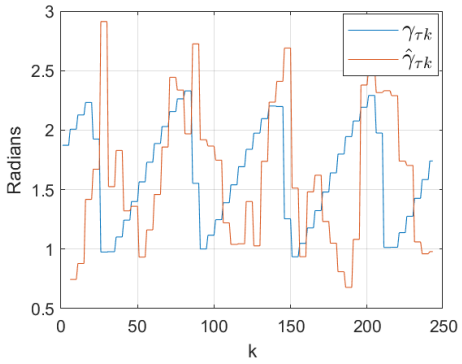
(d) Test 1d (4m/8m)

Figure 4.6: True and estimated trajectories for the second sensor setup together with 3σ-regions

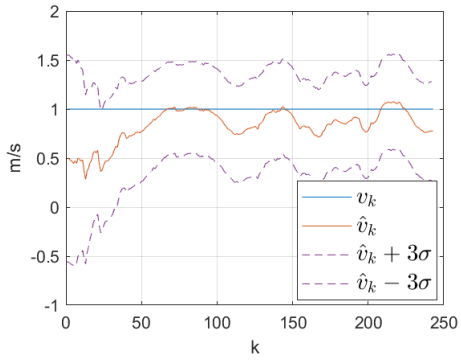
One can see that the estimated trajectory is close to the true trajectory for short distances, but further away for longer distances. There also seems to be a zone where the measurements are of poor quality on the left-hand side of the vehicle. However, large parts of the true trajectory is within three standard deviations, meaning that the UKF is performing well in terms of estimating the position of the key despite the occasionally poor measurement quality. An example of the TE-states produced, in this case for trajectory 1a, are shown in figure 4.7.



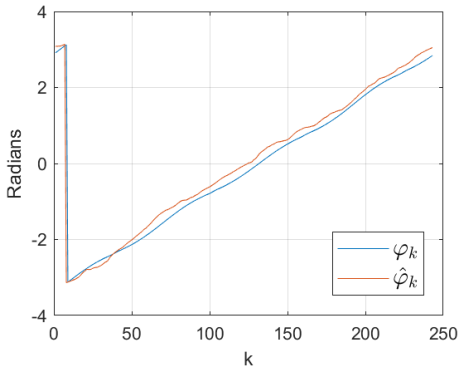
(a) True and estimated distance



(b) True and estimated heading w.r.t. center of car



(c) True, estimated velocity and 3σ-regions



(d) True and estimated angle of arrival

Figure 4.7: True and estimated TE-states of trajectory 4.6a

4.2 Setups of the evaluation tests with final anchor setup

To answer the research questions in this thesis, a few tests were carried out in order to evaluate the complete performance of the Driver Intention System. First, a set of trajectories were defined to test the general performance of the UWB system under the influence of disturbances. These disturbances involved electromagnetic disturbances and obstructed line of sight. Later, another set of trajectories were used to test the overall performance of the TE algorithm, which provided the foundation for answering the first research question. Finally, the performance of the IP algorithm was evaluated for trajectories ending at or passing by the vehicle. The measurements used for these tests are all gathered from the UWB system, and the trajectories are filtered through the TE algorithm. The results from these tests are later presented in chapter 5. To remind the reader, the research questions introduced in section 1.3 are again presented below.

- **RQ1:** How can the key position measured by UWB sensors be used to optimally estimate the position and trajectory of the key, and how robust is this solution?

- **RQ2:** To what extent can the Driver Intention System be used to identify false positives, that is, when the driver seemingly intends to use the vehicle but decides not to and how can they be minimised in order to minimise power consumption?

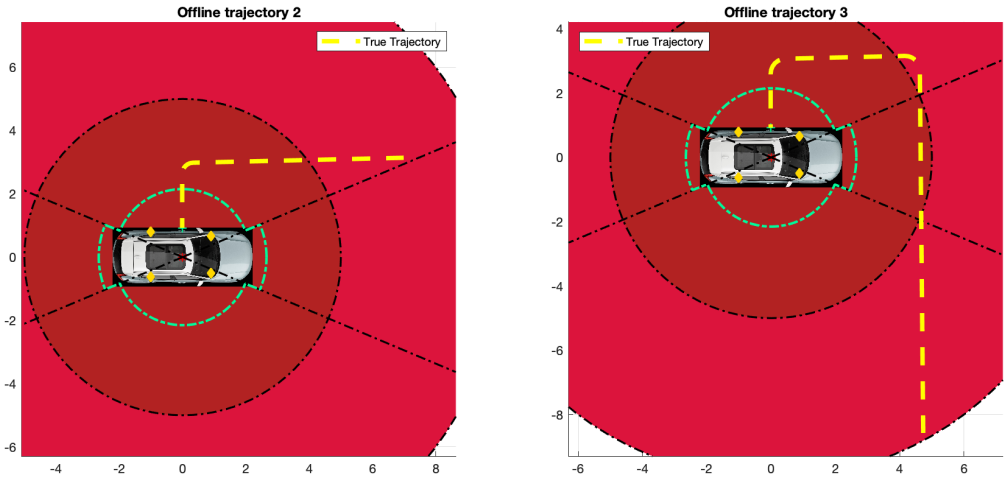
- **RQ3:** How does the Driver Intention System perform for trajectories that enter or leave the vehicle, thus trajectories that can induce false positives and false negatives respectively, compared to other proposed solutions?

- **RQ4:** How can the UWB-data and other data types such as GPS position and time of day be used to further model the intention of the driver?

4.2.1 General performance with influence of disturbances

Even though UWB sensors with their ability to handle disturbances outperforms other types of sensors, such as BLE, the influence of disturbances can heavily affect the overall performance. The disturbances investigated are electromagnetic disturbances, i.e. from a laptop, and obstruction of line of sight, from body mass and other vehicles. These tests are meant to provide answers to the first research question, where the estimated trajectory of the key may be affected from disturbances. Early in the process of gathering measurements, a clear link between the influence of magnetic disturbances and the deteriorating sensor quality was discovered. Since the gathering of measurements is performed with a tag connected to a laptop via USB, this would occur when the computer was within close range to the tag. To evaluate this, a trajectory with the influence of magnetic disturbance that uses the computer within a 20 cm range to the tag, is compared with another trajectory without disturbance that use a range over 50 cm from the computer to the tag. The difference in performance was tested with the setup for trajectory 2, which is shown again in figure 4.8a.

The next type of disturbance investigated was the line of sight being obstructed by body mass or some kind of human interference. For a realistic scenario of the Driver Intention System, the key may be placed in a pocket or bag when approaching the vehicle. Such scenario may lead to the line of sight being blocked by body mass, and is the reason for why it is important to test this type of disturbance. To test this, trajectory 3, shown again in figure 4.8b, was used for two different scenarios; keeping the tag in the front pocket facing the vehicle, and in the back pocket facing away from the vehicle. The resulting measurements are then compared with the same trajectory, to evaluate the difference in accuracy in relation to the true trajectory.



(a) Magnetic disturbance test trajectory

(b) Body mass test trajectory

Figure 4.8: Trajectories used to evaluate the influence of magnetic disturbance and obstruction of line of sight through body mass.

Finally, the disturbance of other vehicles was tested since it is a very common type of disturbance, for example when the vehicle is parked in a parking spot. To evaluate this, four different trajectories was defined where the driver moves towards and away from the vehicle that is parked around other vehicles. The reason for investigating the movement to and from the vehicle, is to further evaluate the UWB system and its ability to provide measurements when initiated with high or low sensor quality. The trajectories used in this tests are named as $2a - 2d$ which can be seen in figure 4.9.

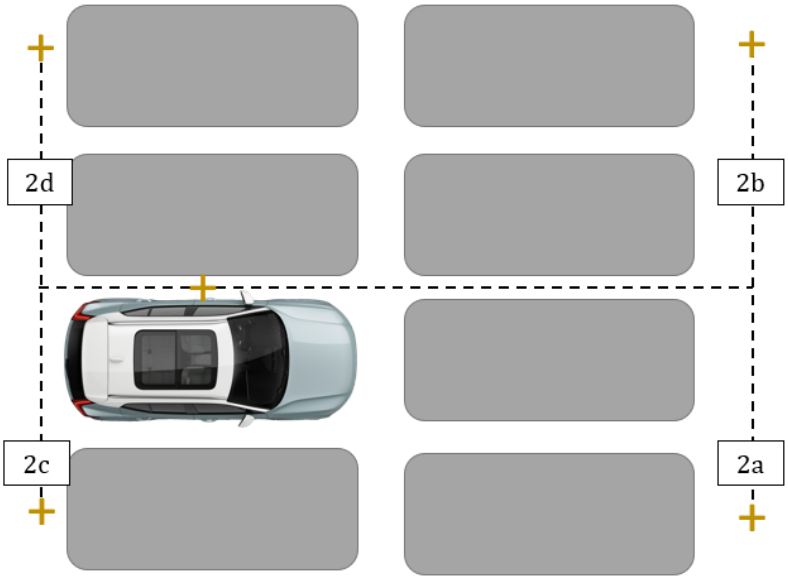
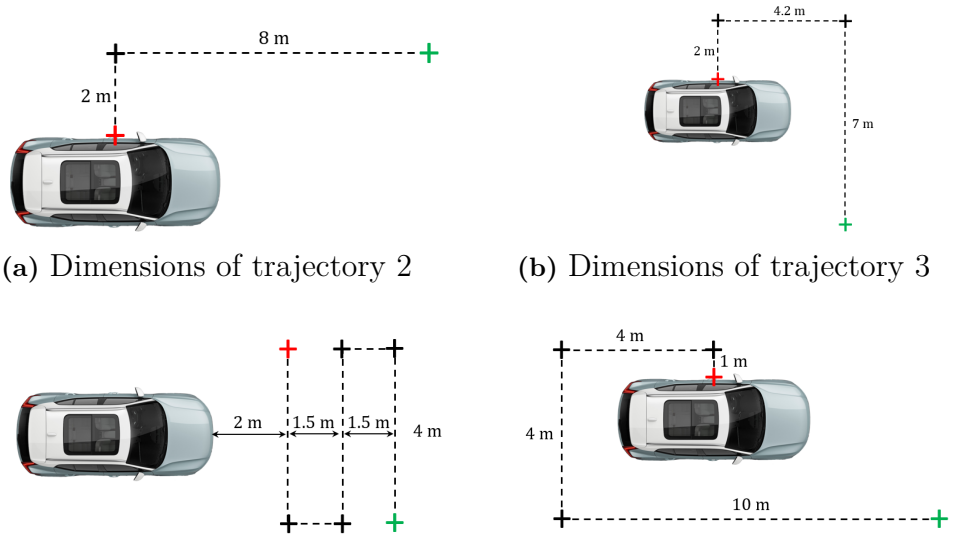


Figure 4.9: Trajectories used for investigating the impact of disturbances caused by other vehicles, depicted as gray boxes. The yellow crosses marks either the starting or ending point of the trajectory, depending on the test.

4.2.2 TE-algorithm performance

With the overall performance evaluated in section 4.1.1 in order to find the final anchor, the following step is to evaluate the performance of the TE in more realistic scenarios. This was done by letting the tag travel some of the trajectories previously used during the tuning process of the Unscented Kalman Filter in section 3.3.1 and gathering measurements of these using the UWB-system. The trajectories chosen for this test were the true trajectories 2, 3, 5 and 6, previously seen in figure 3.8. With the TE-states computed by feeding these measurements through the TE-algorithm, the performance of the TE-algorithm was measured in terms of the performance metrics, introduced in section 2.6, namely accuracy and precision, as well as RMSE-values, in comparison to the true TE-states based on the true key trajectory. To create a fair metric of the performance of the algorithm, each trajectory was conducted three times and the RMSE-values were averaged over these. To ensure obtaining results that focused on the TE algorithm, the tests were done with the vehicle placed in an open space with a clear surrounding line of sight. The reason to why these trajectories were chosen was also to be able to draw conclusions regarding the accuracy of the position estimates during high turning rates, as in figure 4.10c, switching between anchors, as in 4.10d as well as the performance over different distances occurring for all trajectories.



(a) Dimensions of trajectory 2

(b) Dimensions of trajectory 3

(c) Dimensions of trajectory 5

(d) Dimensions of trajectory 6

Figure 4.10: The dimensions of the trajectories chosen for the key to travel which was then measured by the UWB-system. The starting points are marked in green and ending points in red.

4.2.3 Evaluation test for the IP algorithm

This test is made to evaluate the IP algorithm and its ability to avoid false positives, and also the ability to trigger zones early for trajectories that enter the vehicle. Eight scenarios were used in this test, where four of them enter the vehicle, and four do

not. The trajectories used for this test are produced from the TE algorithm, in order to test the IP algorithm through the Driver Intention System. The result from the IP algorithm is compared to results from two separate models with static zones, which was described earlier in chapter 2.1. The distance thresholds used for the static models are presented below in table 4.1, where the distances are from the center of the vehicle to the key. The condition for reaching zone 1 is identical for the static models as the IP-algorithm, thus the only difference between the three models are the conditions for triggering zone 3 and zone 2.

Table 4.1: Distance thresholds of triggering zones for the two static zone models.

	Zone 2 threshold	**Zone 3 threshold**
Static Model 1	3.5 m	5 m
Static Model 2	4 m	6 m

The results from this test are used to answer the second and third research question, where the static models act as the ”other proposed solutions”. For the scenarios that do not enter the vehicle, the evaluation for each model is based on the active up time for the zones, where low up active time corresponds to low power consumption. The evaluation of the scenarios that enter the vehicle is based on the amount of time each zone is active, and also if any zones are triggered several times before reaching the vehicle.

5
Results

This chapter contains the results gathered through the experiments with the experimental setups that were presented in chapter 4.

5.1 Evaluation of the general performance while under the influence of disturbances

This section covers the results of the tests presented earlier in section 4.2.1. The types of disturbances that were tested are the influence of magnetic disturbance, other vehicles and body mass.

5.1.1 Influence of magnetic disturbance

The first disturbance that was evaluated is the influence of magnetic disturbances from a computer. Presented in figure 5.1 are the results for walking the trajectory with and without the influence of the computer, where both configurations were tested three times each.

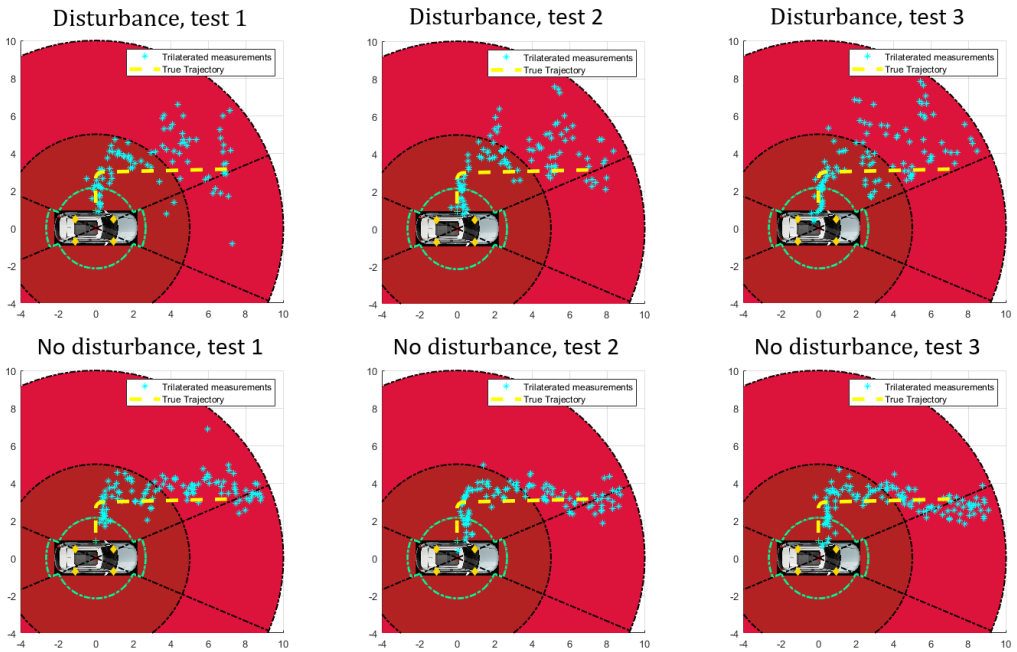


Figure 5.1: Comparison of preprocessed trilaterated measurement with and without the influence of magnetic disturbance.

The measurements influenced with magnetic disturbance are seen to be consistently inaccurate to the true trajectory for longer distances, but better within close range. The measurements without the disturbance are more accurate with an expected increased variance for longer distances.

5.1.2 Influence of body mass

The test for evaluating the disturbance of human interference and body mass was conducted two times each for the front pocket and back pocket. The results of these tests are shown below in figure 5.2.

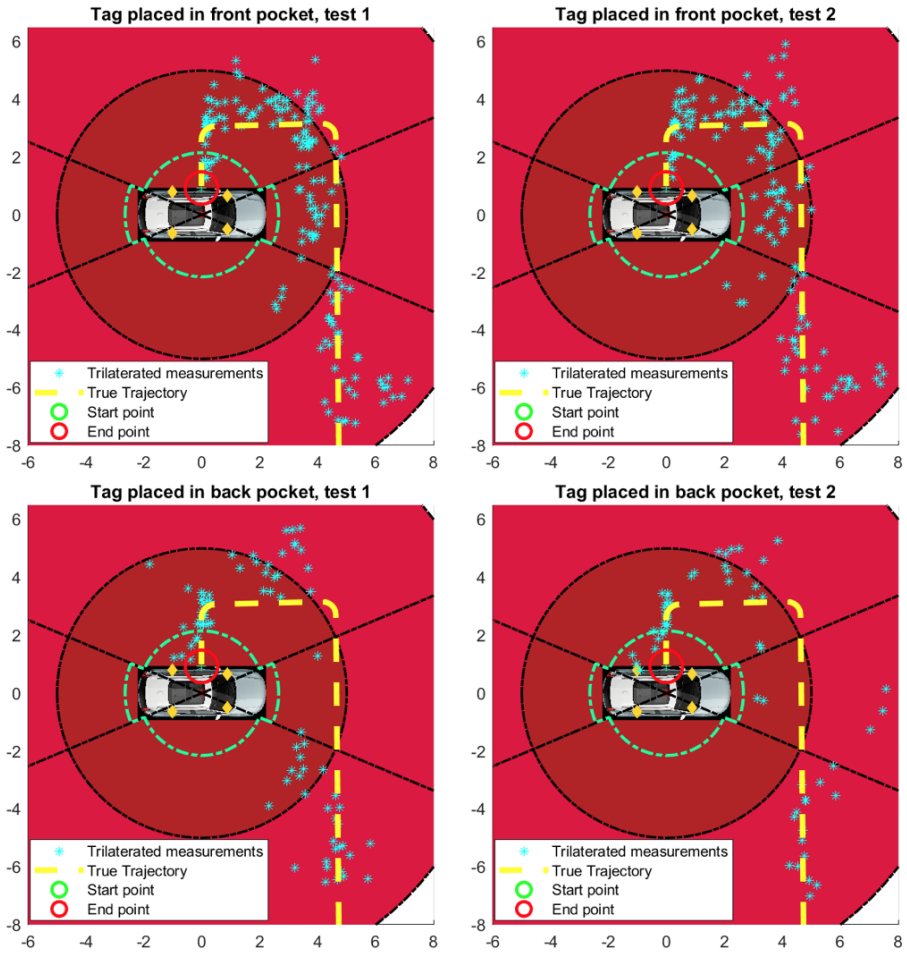
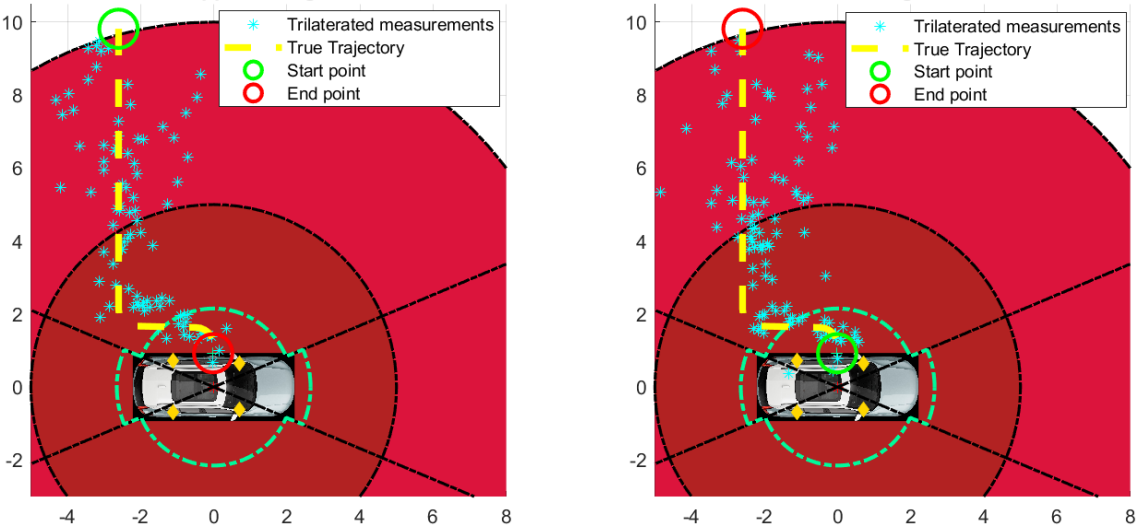


Figure 5.2: Resulting measurements from the test with disturbance of body mass.

It is shown for the tests with the tag placed in the front pocket that the measurements are a bit off in comparison to the true trajectory for distances longer than 3 m, but become more accurate for closer range. However, the tests with the tag placed in the back pocket are shown to lose a lot of measurements due to the low signal quality. The measurements within a range of 3 m seems to be accurately detected, but the rest of the measurements are not well enough to be used for the IP algorithm.

5.1.3 Influence of other vehicles

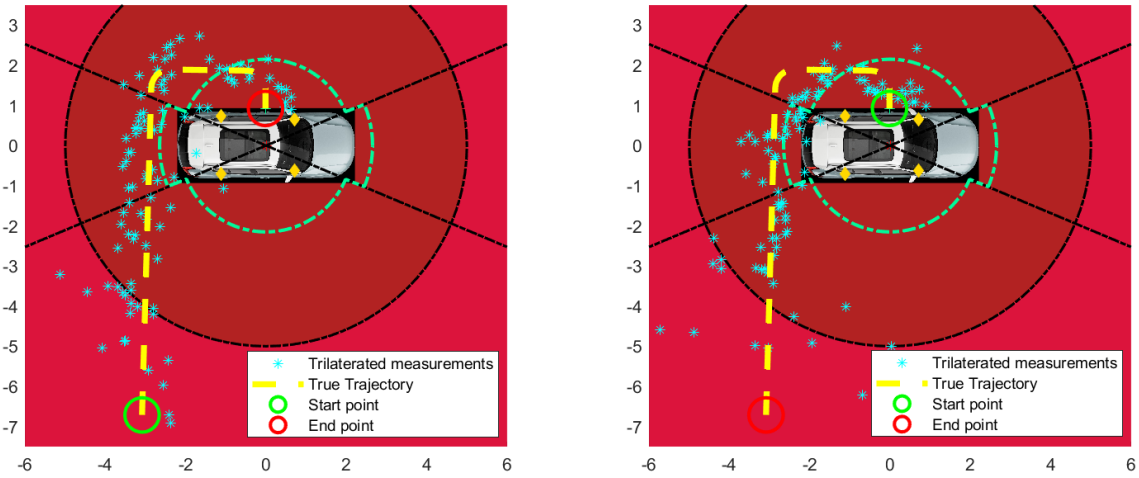
The next disturbance to evaluate is the performance while enclosed around other vehicles in a parking spot. Each of the trajectories 2a-2d, see figure 4.9, were executed both with the tag approaching and leaving the vehicle respectively. The result from the first trajectory, 2a, is seen in figure 5.3.



(a) Approaching the vehicle **(b)** Leaving the vehicle

Figure 5.3: Test 2a, approaching and leaving the vehicle parked in a parking spot.

The resulting measurements for both approach and leave are similar, where both the measurement sequences are well enough to yield decent TE-states. The next plots in figure 5.4 shows the results from test 2b.

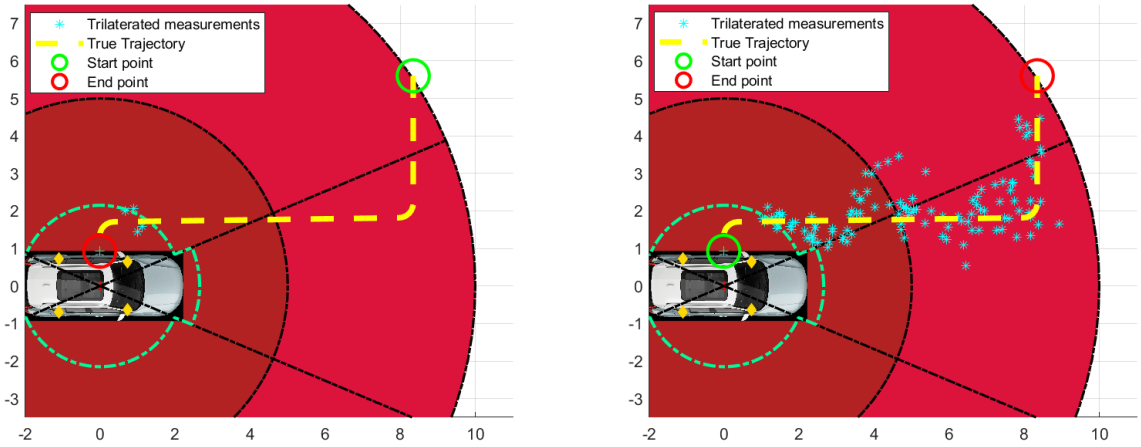


(a) Approaching the vehicle **(b)** Leaving the vehicle

Figure 5.4: Test 2b, approaching and leaving the vehicle parked in a parking spot.

Same can be said here as the previous test, except a worse accuracy of the measure-

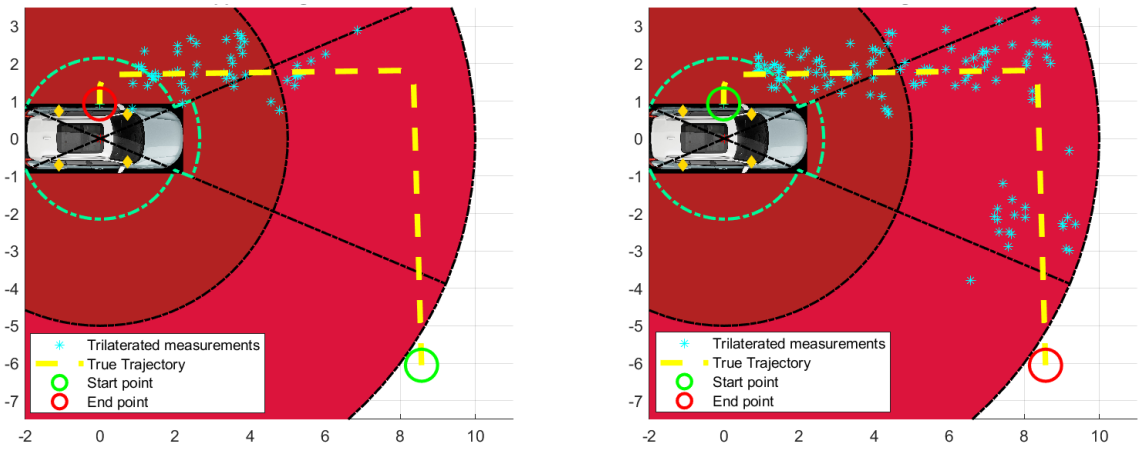
ments when leaving the vehicle. The next test, 2c, was conducted with another line of cars in the front of the vehicle. The results are shown in figure 5.5.



(a) Approaching the vehicle **(b)** Leaving the vehicle

Figure 5.5: Test 2c, approaching and leaving the vehicle parked in a parking spot.

The measurements for the trajectory leaving the vehicle are fairly accurate, but the trajectory for approaching the vehicle does not yield any measurements until the driver's side is reached. The final test, 2d, is presented in figure 5.6.



(a) Approaching the vehicle **(b)** Leaving the vehicle

Figure 5.6: Test 2d, approaching and leaving the vehicle parked in a parking spot.

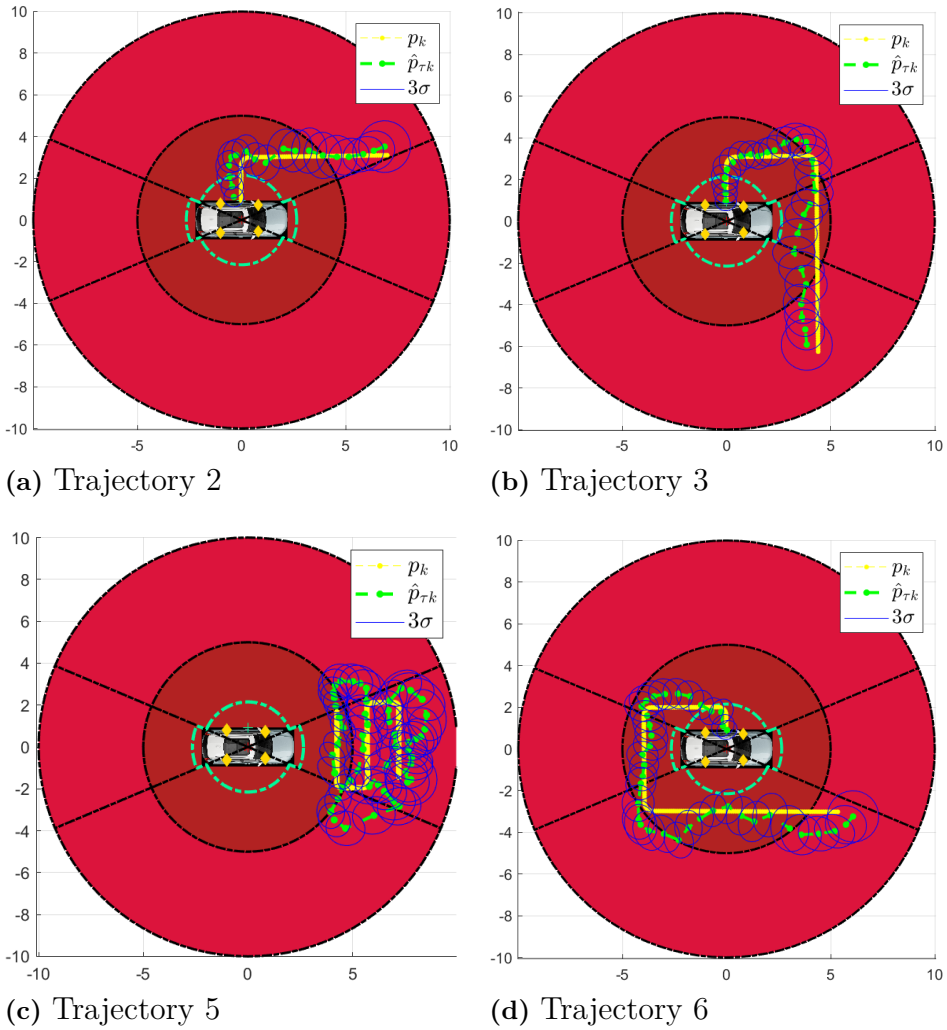
Similar behaviour can be found here as the previous test, where the measurements for approaching the vehicle can not be made until the tag is close enough. It is also possible to see that no measurements are gathered in the front of the vehicle for the trajectory leaving it. This is most likely due to the car that is parked in the front which is obscuring of line of sight.

5.2 Evaluation of the state estimation performance of the TE

The following section presents the results of how the TE performs in terms of how well it estimates the position of the key using measurements from the UWB-system, and also results of how it performs in terms of estimating the TE state vector $\mathbf{x}_k^{(TE)}$.

5.2.1 TE performance with selected key trajectories

In this sub-section the results for accuracy and precision are presented for trajectories 2, 3, 5 and 6, as well as the RMSE values of them compared with those created with the simulated measurements. Figure 5.7 shows the position estimates and the 3σ uncertainty regions of each estimate for one out of three filtered measurement sequences of each trajectory.

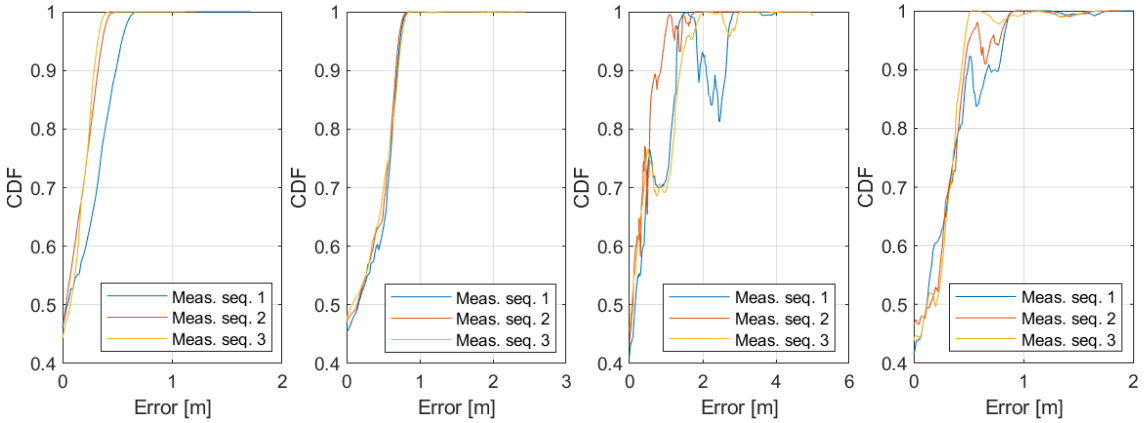


(a) Trajectory 2 **(b)** Trajectory 3

(c) Trajectory 5 **(d)** Trajectory 6

Figure 5.7: True and estimated trajectories together with uncertainty regions for each estimates.

Figure 5.8 displays the Cumulative Distribution Functions of the accuracy of the position estimates for each trajectory.



(a) Trajectory 2 **(b)** Trajectory 3 **(c)** Trajectory 5 **(d)** Trajectory 6

Figure 5.8: CDF of the accuracy of the position estimates of each trajectory over each measurement.

Table 5.1 shows the RMSE values for the TE states for each of the four trajectories.

Table 5.1: RMSE values for each of the TE-states averaged over three measurement sequences with anchor setup 2 and compared to the RMSE of their simulated counterpart. The differences with "+/-" means that the RMSE value is higher, or lower, when using the gathered measurements than the simulated.

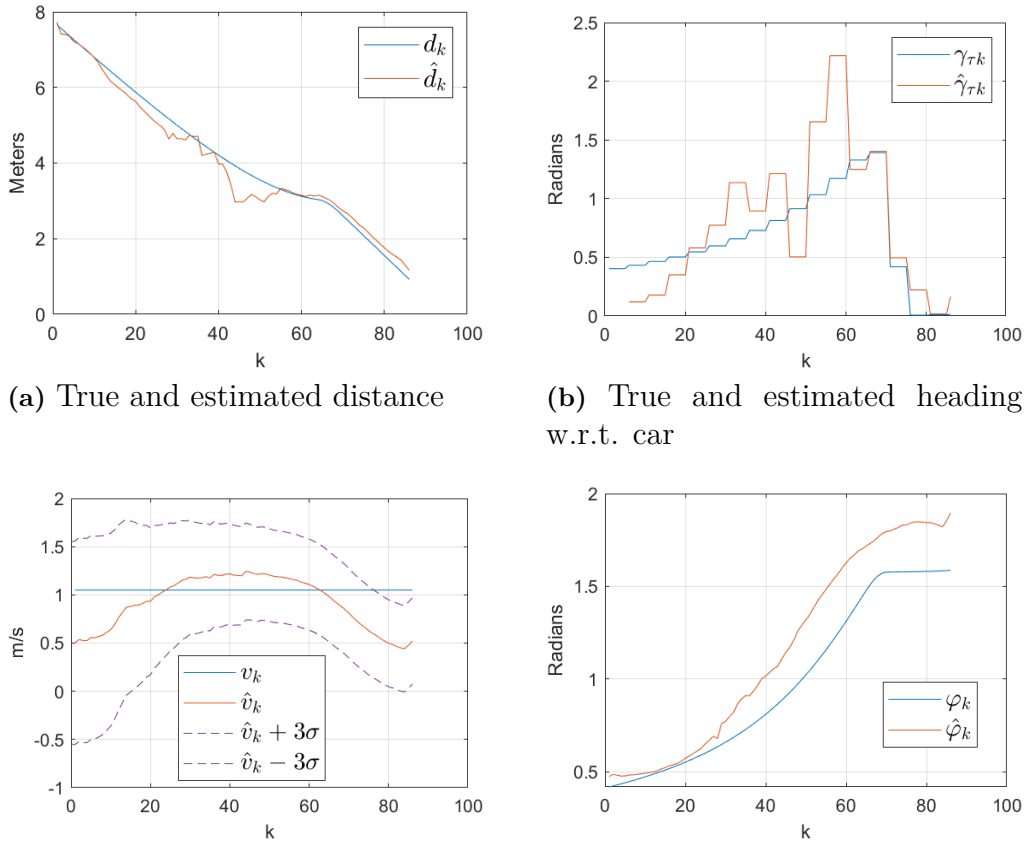
	Traj. 2			**Traj. 3**		
	Sim.	Meas.	Diff.	Sim.	Meas.	Diff.
d_{RMSE}	0.413	0.402	-0.011	0.396	0.723	+0.327
γ_{RMSE}	0.476	0.385	-0.091	0.488	0.539	+0.051
v_{RMSE}	0.226	0.324	+0.098	0.236	0.392	+0.156
$\cos(\varphi_{RMSE})$	0.100	0.188	+0.088	0.067	0.147	+0.080
	Traj. 5			**Traj. 6**		
	Sim.	Meas.	Diff.	Sim.	Meas.	Diff.
d_{RMSE}	0.374	0.813	+0.439	0.362	0.773	+0.411
γ_{RMSE}	0.731	0.880	+0.149	0.510	0.546	+0.036
v_{RMSE}	0.245	0.309	+0.064	0.173	0.287	+0.114
$\cos(\varphi_{RMSE})$	0.019	0.067	+0.048	0.062	0.114	+0.052

Table 5.2 shows the RMSE values averaged over all four trajectories for the simulated measurements and UWB-system measurements respectively.

Table 5.2: RMSE values for each of the TE-states averaged over all four trajectories.

	Sim.	Meas.	Diff.
$\bar{d}_{RMSE}$	0.386	0.678	+0.292
$\bar{\gamma}_{RMSE}$	0.551	0.588	+0.037
$\bar{v}_{RMSE}$	0.220	0.328	+0.118
$\cos(\bar{\varphi}_{RMSE})$	0.062	0.129	+ 0.067

Figure 5.9 shows an example comparison between the true and estimates states for trajectory 2 seen in figure 5.7a.



(a) True and estimated distance

(b) True and estimated heading w.r.t. car

(c) True and estimated velocity with 3σ estimates.

(d) True and estimated angle of arrival

Figure 5.9: Comparison between true and estimated TE-states for example trajectory 2.

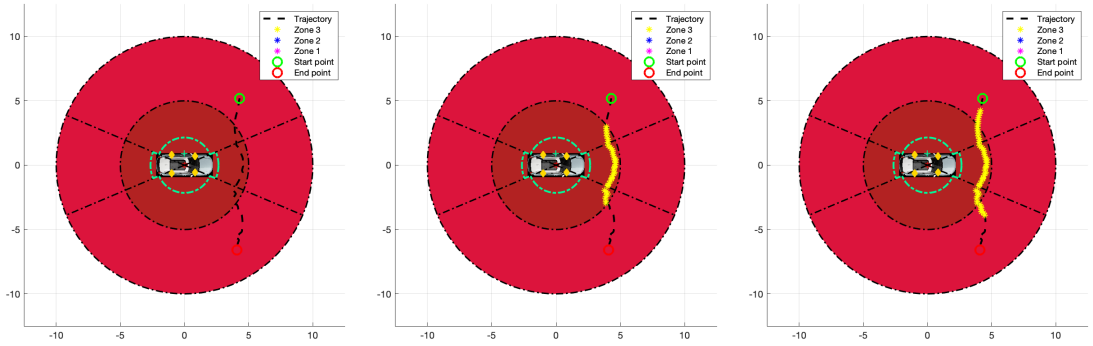
5.3 Evaluation of the IP algorithm's performance against static zone models

This section covers the results for the performance of the IP algorithm compared to the static zone models. The evaluation is made for scenarios that enter the vehicle,

and those that do not. All the trajectories are gathered through the UWB system, and the IP-states are produced from the TE algorithm. The full setup for these tests can be found earlier in section 4.2.3.

5.3.1 Results for scenarios that do not enter the vehicle

The figures presented in this section only illustrates the points in the trajectory which triggers zone 3 and zone 2. Additionally, during the set timeout interval $\Psi = 5$ s after the zone has been left, the functions of each zone are still active which is accounted for in the "Active up time" for the functions of each zone, which is presented in the table after each trajectory. The result from the first IP-trajectory and the up time for each zone is presented below in figure 5.10.

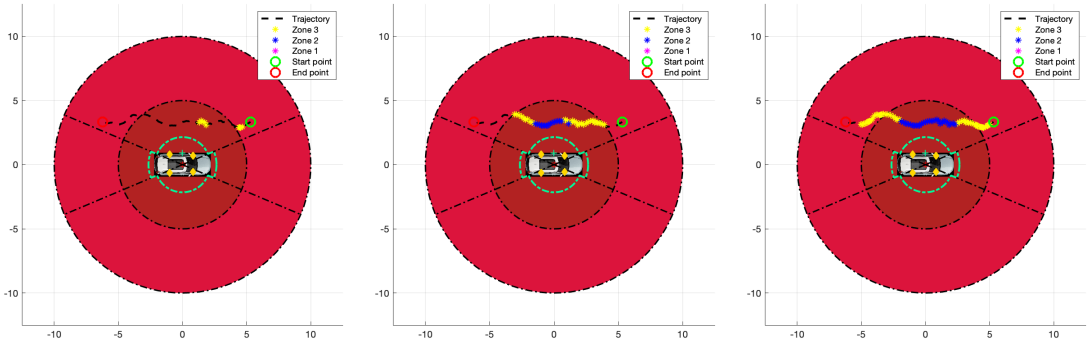


(a) IP-Algorithm **(b)** Static Model 1 **(c)** Static Model 2

	IP-algorithm	Static model 1	Static model 2
Zone 2	-	-	-
Zone 3	-	8.0 s	10.5 s

Figure 5.10: Result for IP-trajectory 1

The result shows that the IP-algorithm manages to avoid activating any zones for this trajectory, while both static models triggers zone 3. The next trajectory and the results are presented in figure 5.11.

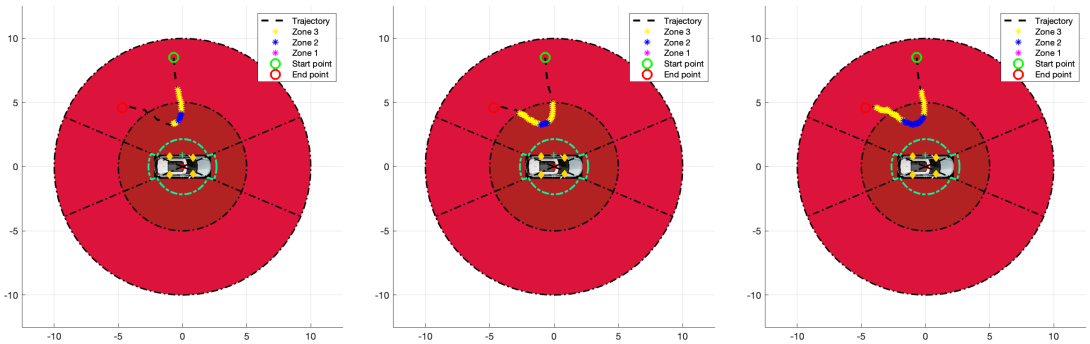


(a) IP-Algorithm (b) Static Model 1 (c) Static Model 2

	IP-algorithm	Static model 1	Static model 2
Zone 2	-	6.6 s	8.0 s
Zone 3	7.9 s	9.8 s	13.1 s

Figure 5.11: Result for IP-trajectory 2

The IP-algorithm reaches the criteria for triggering zone 3 but manages to keep the loss value high enough to avoid zone 2. The time between the two points that triggers zone 3 are lower than $\Psi = 5$ s, which means that the functions for zone 3 are constantly active during the 7.9 s of active up time. Both the static models enters zone 2 and zone 3. The result for the next trajectory is shown in figure 5.12.

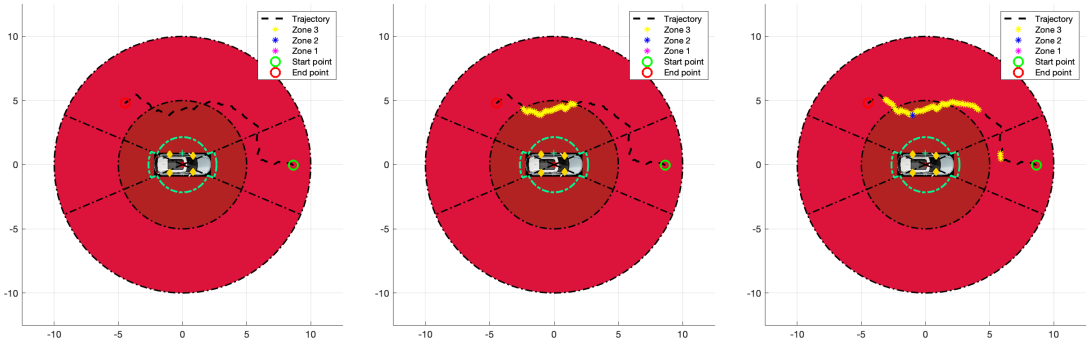


(a) IP-Algorithm (b) Static Model 1 (c) Static Model 2

	IP-algorithm	Static model 1	Static model 2
Zone 2	5.6 s	5.7 s	6.8 s
Zone 3	7.3 s	8.5 s	10.3 s

Figure 5.12: Result for IP-trajectory 3

All of the models are seen to trigger both zones where the IP-algorithm is the first to trigger both of them. Since the IP-algorithm also leaves both zones first, it manages to achieve the lowest active up time for this trajectory. The result for the last trajectory is presented in figure 5.13.



(a) IP-Algorithm (b) Static Model 1 (c) Static Model 2

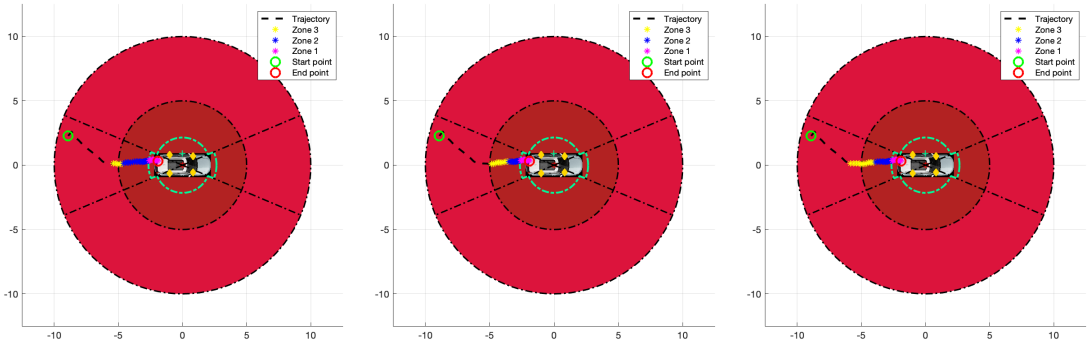
	IP-algorithm	Static model 1	Static model 2
Zone 2	-	-	5.1 s
Zone 3	-	8.5 s	13.0 s

Figure 5.13: Result for IP-trajectory 4

Again, the IP-algorithm avoids triggering both zones while both of the static models enters zone 3. The second static model is shown to briefly entering zone 2 for one measurement, which by design keeps the zone 2 functions active for 5.1 s. In summary, the IP-algorithm achieves the lowest period of active up time for both zones for every test scenario that does not enter the vehicle.

5.3.2 Scenarios that enter the vehicle

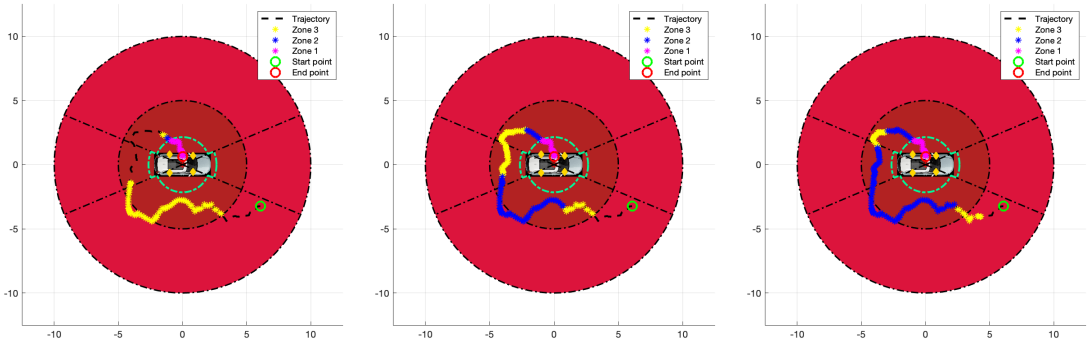
The result presented for these tests includes the complete zone active up time, including the set timeout interval for each scenario. The result for the first trajectory is presented below in figure 5.14.



(a) IP-Algorithm (b) Static Model 1 (c) Static Model 2

Figure 5.14: Result for IP-trajectory 5

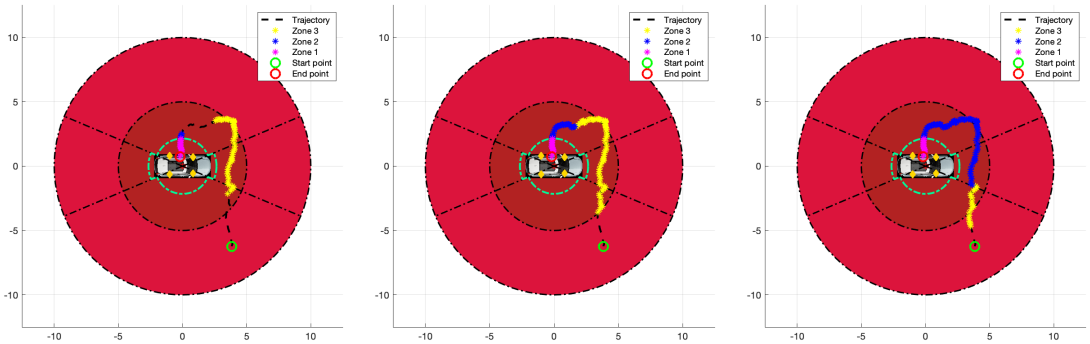
The results show that the IP-algorithm triggers zone 3 roughly at the same distance as the second static model, and is the first to trigger zone 2. The result for the next scenario is shown below in figure 5.15.



(a) IP-Algorithm **(b)** Static Model 1 **(c)** Static Model 2

Figure 5.15: Result for IP-trajectory 6

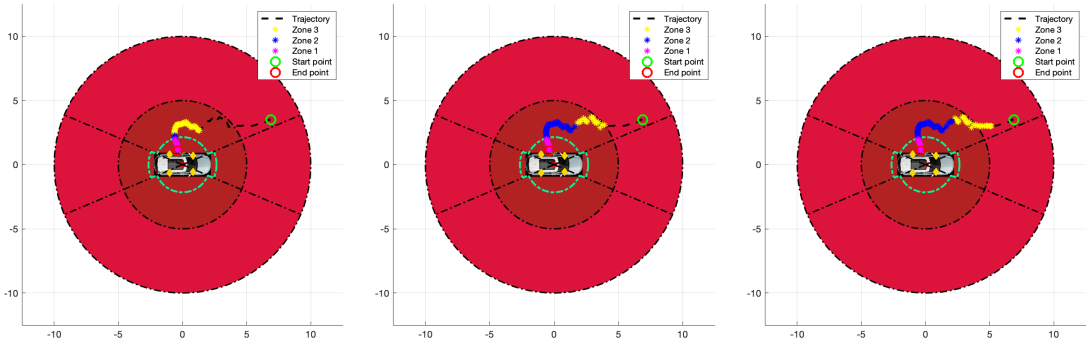
The IP-algorithm triggers and deactivates zone 3 before reaching the vehicle, thus decreasing the threshold for zone 3 before entering again. The two static models are activating and deactivating the functions for both zones before entering the vehicle, which means the lights will turn fully on and off twice before reaching the vehicle. The next scenario is presented below in figure 5.16.



(a) IP-Algorithm **(b)** Static Model 1 **(c)** Static Model 2

Figure 5.16: Result for IP-trajectory 7

In this scenario, the IP-algorithm has the worst performance, where the zone 3 is deactivated before entering zone 2, which does not occur for the static models. The final scenario is presented below in figure 5.17.



(a) IP-Algorithm **(b)** Static Model 1 **(c)** Static Model 2

Figure 5.17: Result for IP-trajectory 8

For this scenario, the IP-algorithm triggers both zones at a closer distance than the two static models. Overall, the IP algorithm is shown to trigger both zones at closer distance than the static zones for trajectories that does not head directly towards the vehicle.

6
Discussion

The following chapter presents the discussion and analysis of the results from the evaluation tests, and the analysis of the project in its entirety. A set of guidelines and ideas for future work are also presented, such as how this prototype could be further developed at a later stage. Lastly, the conclusion of the thesis presents the answers to the four research questions, which marks the end of this thesis.

6.1 Evaluation tests

Starting with the disturbance tests, the results show that the UWB system is indeed influenced by each of the three investigated disturbances. For the magnetic disturbance test, when using the tag within a 20 cm range of the laptop, the measurement quality gathered by the anchors consistently drops for all three test runs, and so does also the accuracy for the measured distance. This type of disturbance was however explicitly tested since the gathering measurements were made within close vicinity of a laptop, which would not be a common scenario for a finalized product. Though, it indicates that a finalised product would have to be optimised to withstand similar scenarios that may be caused by mobile phones, if the phone is to be used as the key.

The next disturbance test, influence of body mass, indicates that obstructing the line of sight with a human body can completely block the UWB signal. This issue is highly concerning for common scenarios, such as when the driver approaches the vehicle when keeping the key in e.g. a backpack or placed in a back pocket. A discussed solution to this would be to include other sensors, such as the accelerometer and gyroscope in the phone to be used as a complement for the UWB system, especially while observing low values of sensor quality of the anchors. Again, this further proves that a UWB-based system for monitoring the key would have to be optimised for how it may be commonly used.

The final disturbance test, influence of other vehicles, indicates that the measurements can be influenced, or fully blocked, while the vehicle is parked close to other vehicles. The measurements gathered during tests 2a and 2b, shown in figures 5.3 and 5.4, have larger variances than those of the trajectories that have a free line of sight, which may be due to reflections caused by nearby vehicles. The measurements gathered from tests 2c and 2d, presented in figures 5.5 and 5.6, show large differences between the trajectories that are approaching and leaving the vehicle respectively. Looking at test 2c when leaving the vehicle, the very few measurements

obtained seem to be accurate, but with the main issue that they are too few. The reason for this is not clear, but an assumption is that the UWB system does not start to provide measurements before a certain threshold of the sensor quality has been reached, which seems to be easier to maintain when leaving the vehicle. It was discussed that these issues could be possible to circumvent if the source code to the Decawave software was modified, as well as increasing the sample rate since the current sample rate combined with the velocities of the key may lead to the UWB-system detecting the key "too late" when reading the measurement quality. Alternatively, more anchors could be used to cover a larger area to detect the key. The first circumvention to increase the sample rate would require access to the source code of the Decawave software, and the latter more hardware resources, which was unfortunately not possible for this thesis.

Moving over to the tests regarding the performance of the TE algorithm to estimate the position of the tag, it is also noticeable that an obstructed line of sight has a negative impact on the UWB system, but also that trajectories with several sharp turns close to each other causes poor position estimates. This was especially the case for trajectory 5 seen in figure 5.7c when the rear anchors would be obstructed by several layers of objects within the vehicle, mainly being the seats, and having to "look through" the vehicle when the tag is travelling in a trajectory with several sharp turns. The effect of this can not only be seen in figure 5.7c when comparing the estimated trajectory to the true trajectory, but also in figure 5.8c where an error larger than two metres is almost always to be expected.

The accuracy can be considered much better for trajectories 2,3 and 6 where the error is almost always less than one metre, which can be seen in figure 5.8. However, these trajectories are more constant and with less sharp turns. Despite the differences between trajectory 5 and 2, 3, 6, most of the true trajectories are within the three standard deviations of the position estimates, as seen in figure 5.7, indicating sufficient filter performance. The RMSE values of the estimated velocity and angle of arrival using the measurements from the UWB system are also low, further indicating that the filter performs adequately in estimating these. However, for trajectories 3 and 6, the RMSE values using the UWB measurements are higher for the estimated distance, even though these have a less obstructed line of sight and sharp turns. This may indicate that the filter may not completely consider the variance in the distance of the trilaterated measurements in an adequate way.

Figures 5.7a, 5.7b and 5.7d also imply that the TE algorithm performs better when the line of sight is less obstructed and also when the trajectories are more constant. These factors are believed to cause the UWB-system being able to gather more measurements and thus more information for the Unscented Kalman Filter to work with at the current sample rate. Due to the poor performance when trying to estimate trajectory 5, the average RMSE values over all the trajectories presented in table 5.2 can most likely be much lower with a more optimised system. As mentioned earlier, the initial steps to achieve such a system could be by including more anchors alternatively that the sample rate is increased.

For the final test regarding the IP algorithm, it is shown that the algorithm has potential to outperform the static zone models, both for avoiding false positives and waking the vehicle in optimal ways during scenarios when the driver enters the car. With the chosen parameters, the IP algorithm manages to avoid false positives for two out of the four chosen IP-trajectories that does not enter the vehicle. Additionally, the dynamic threshold would make sure to avoid the two other trajectories, if the same trajectory was redone a second time.

Regarding the four chosen IP-trajectories leading to the vehicle, the IP algorithm performs well compared to the static zone models — in terms of triggering zones early — for IP-trajectory 5 shown in figure 5.14. As presented in section 3.4.3, with the chosen weight parameters and zone thresholds, the IP algorithm has the potential of triggering both zones earlier than the static zone models, for low $\hat{\gamma}_k$ and high $\hat{v}_k$. The remaining IP-trajectories show the problem of using a too short time-out interval $\Psi = 5s$, where the zones are activated and deactivated before reaching the vehicle. The choice of Ψ is not trivial since it provides a trade-off between high energy consumption for events of false positives, and the observed problem of deactivating functions before arriving to the vehicle. Further research into the aspects of user experience, such as the modelling of driver usage patterns introduced in section 3.4.5, and allowed energy consumption within the vehicle would be required to find an optimal value for Ψ, which unfortunately is beyond the scope of this thesis.

6.2 General considerations and future work

An important aspect to consider regarding the general problem of optimal vehicle wake-up is that the solution should satisfy every user. To develop such a solution is challenging due to every user having their own individual preferences of moving around and approaching their vehicle, as well as the location of the designated parking spot varying from user to user. This thesis proposes a general solution which is not formed to satisfy each and every user, but rather to be used as a baseline to which further functionalities are to be added. This could include features that adapts to the user over time, such as the concept of modelling the driver usage patterns introduced in section 3.4.5.

Another aspect that needs to be addressed is the use of multiple tuning parameters in the Driver Intention System, especially the weight factors for the IP algorithm. The inclusion of these parameters makes it possible to tune it for an individual, such that it can satisfy their personal preferences. However, a problem with this solution is that the weight factors are co-dependent towards the resulting loss value, which means it is fairly difficult to tune them in order to design a detailed desired behaviour of the IP algorithm. The method for tuning the parameters, presented in section 3.12, is acceptable for a proof of concept, but may require a more sophisticated method for future implementations. These methods could also be connected to the concept of modelling the driver usage patterns over time.

While on the subject of the IP algorithm, the decision to not fully base the decision-making method on Bayesian Decision Theory was made because the covariance of the IP-states was considered complex to estimate. However, through late consultation with our supervisor, an idea to add a separate filter between the TE algorithm and IP algorithm was discussed. This filter, such as another UKF with the TE-states as inputs, could then be used to estimate the covariance of these states which would produce the correct type of inputs for Bayesian Decision Theory. Another idea would be to modify the motion model so that it included the IP-states directly so that the UKF would estimate them directly, which would have produced the needed covariances for Bayesian Decision Theory. Due to the time constraint, this was unfortunately never tested and will have to be tried out for future work. Though, an argument for using the chosen decision-making method in this thesis, is that this method is simpler, where the alternative method may over-complicate the prototype.

Moving over to the choice of hardware, the Decawave UWB system was used explicitly due to Volvo Cars recently using it in other tests and requesting that this system should be investigated further. However, this can be considered as a budget choice of UWB system, and is mainly designed for experimental use. A comparison against other types of UWB systems unfortunately lie beyond the scope of the thesis, but could be important for future work. The ideal and recommended use of this UWB system is to place the anchors at a height of 2.5 - 5 m from ground level, and track the tag in an area where it is surrounded by the anchors in use [20]. Even though the chosen anchor setup does not follow these recommendations due to geometric constraints of the vehicle, it still manages to produce fairly accurate measurements. For future work, a permanent anchor placement with more anchors needs to be investigated, where the height of the placed anchors is important in order to avoid the influence of body mass and where the anchor placement is more integrated in the vehicle. A test with placing an anchor directly under the outer part of aluminium-chassi and another directly behind the grille showed substantially better measurement quality. Therefore, Volvo Cars could begin with attaining more anchors and try to find permanent anchor positions in these areas, combined with placing other anchors high above ground, e.g. in the rails on the roof of the vehicle.

As mentioned in chapter 3.2, the trilateration of measurements were designed to be computed with the three closest anchors to the tag. A limitation with the Decawave system is that the information such as sensor quality and accelerometer values are not available through the USB protocols, or through the use of a listener. Also, the maximum number of anchors that can be used simultaneously is limited to four, which removes the option of utilizing many anchors and trilaterating with several combinations of anchors to gather further information of the tag position. To access this information and remaining anchors, one need to use certain development tools by Decawave, or alternatively modify the code of the Decawave software, which was not part of the scope for this thesis. The access of this information should be highly prioritised for future work, since it enables use of other methods for the trilateration, as well as increasing the sample rate, that could significantly improve the accuracy

in the Driver Intention System. This would also make it possible to receive the real-time accelerometer data of the UWB module to further improve the accuracy and precision of the TE algorithm.

The design choice of removing outliers before trilaterating was considered necessary in order to obtain a good enough measurement quality for the filtering within the TE algorithm. Different methods to remove the outliers were investigated, where the most successful method was to remove measurements that increased with a static value of 0.8 m over one sample instance. This method should be further improved for future work, and could instead be substituted with a dynamic value that is directly dependent on the measurement quality of each sensor.

Another important improvement for future work would be to trilaterate the measurements in 3D, rather than in 2D which was decided as a limitation for this project. A minor test was performed to test the difference when tracking the tag for different heights, and the results were quite similar except for a minor offset which could occur for shorter distances to the vehicle. Though, for future work, the inclusion of trilateration in 3D could increase the accuracy of the position estimates, if the tag can consistently communicate with four or more anchors.

Another aspect concerning the trilaterated measurements is that if they are continued to be considered as Gaussian, the measurement noise covariance matrix $\mathbf{R_k}$ should be modified to include the estimated angle of arrival $\hat{\varphi}_k$. This is believed to help in resolving the issue of the UKF poorly estimating the distance, as the implemented $\mathbf{R_k}$ does not fully capture the uncertainties in the distance-offset of the trilaterated measurements originally caused by each individual anchor of the UWB-system occasionally measuring the distance poorly. The poorly estimated distances can be seen in the comparison of RMSE values to using simulated measurements in table 5.1, where the values for the distance are higher with measured data for three out of four trajectories. With the angle of arrival, the UKF would be able to know which Cartesian component of the $\mathbf{R_k}$-matrix should have a larger uncertainty when gathering measurements, and with this information it is believed that the UKF could produce better estimates. As an example, if measurements are made on the left hand side of the vehicle, and thus $\hat{\varphi}_{k-1}$ being approximately $\frac{pi}{2}$, then there should be a larger variance on the y-component than on the x-component, as the trilaterated measurements are believed to have a larger variance in the y-direction in this case.

Concerning the performance evaluation with true trajectories, it is worth pointing out that the true trajectories are not exactly true, as they do not fully incorporate the actual heading, velocity or turning rate of the key, but mainly the Cartesian coordinates. This is because the tag was transported by a person holding it, and walking the path of the trajectory. Thus, the tag is affected by irregularities caused by the movement of the person, which is not captured for the true trajectory — especially for the velocity which was not constant during the gathering of measurements. Instead, a tracking system with a much higher performance could be incorporated

in the test setup that adds the additional information to the true trajectories for better results.

The functionality of modelling the driver usage patterns, introduced in chapter 3.4.5, was only considered on a conceptual stage in this thesis due to the time constraints not allowing the process of gathering hundreds or possibly thousands of trajectories from different users. This is also why the link between the functionality and the setting of threshold values in the IP algorithm was left out, as the primary focus is on the overall concept of including ε-parameters. What Volvo Cars therefore could do for future work is to start logging position and movement data of the key around the vehicle in each vehicle of the current test vehicle fleet that has a similar technology to the UWB system. With this information, they could find ways of integrating ε-parameters and the IP-states in the way proposed in this system, and focus more on the practical implementation which has been outside the scope of this thesis.

As a final remark, this thesis focus mainly on a technical approach for the subject of vehicle wake-up, where the main question to be answered is: "*Can* this function be developed?". Another aspect which should be considered is the perspective of user experience, where the main question would be: "*Should* this function be developed?", and what is the value in such a solution for the user? As stated in the introduction, a solution that completely avoids false positives is close to impossible, which would lead to situations where the user is not fully in control. This thesis presents a proposed solution which has the potential to be dynamically tuned for individual users, though the question of *what the individual user would prefer* remains unanswered and is an aspect that is recommended to be investigated for future work.

6.3 Conclusion

To conclude this thesis, the main objectives have been reached, where a prototype was developed that can quantify the predicted intention of the driver using UWB radar technology, which in return enables the possibility to activate wake-up functions in a vehicle. The remainder of this subsection contains the answers to the four research questions, introduces in section 1.3, in their presented order.

The estimation of the position and trajectory of the key, measured only by UWB sensors, was attainable through the implemented Driver Intention System. In terms of robustness, the estimations are fairly accurate for optimal conditions of free line-of-sight, but are sensitive to external disturbances, such as magnetic fields, body mass and other vehicles.

The inclusion of the relative heading γ and velocity v of the key in the loss function contributed to a better performance for the Driver Intention System in identifying false positives when compared to the static zone models. The possibility of tuning the weight parameters of the loss function makes the system configurable such that an optimal setup can be set for the preference of an individual user. With the set tuning for the weight parameters in this thesis, the power consumption for the tra-

jectories that do not enter the vehicle was lowest for the IP algorithm, in comparison to the two static models.

When testing the Driver Intention System with the same tuning parameters for trajectories that enter the vehicle, the IP algorithm performs well for trajectories that head directly towards the vehicle – but worse for trajectories that do not. The chosen timeout interval of $\Psi = 5$ s before zone deactivation led to a behaviour which deactivated the functions in the vehicle before reaching it. By using a small timeout interval, the power consumption decreases for events of false positives, but increase the risk of deactivating functions in the vehicle for trajectories that do not directly enter the vehicle. Further investigation in finding an optimal timeout interval, Ψ, will be needed for future work.

By using additional types of data, such as GPS position and time of day together with the UWB-data, a conceptual way of modelling the usage patterns of the driver has been formulated, which is hypothesised to be beneficial when predicting the intention of the driver. The concept suggests that the algorithm uses memorised data consisting of the ways the driver usually moves around the vehicle before using it combined with vehicle GPS position and time stamps for when the vehicle was used. The possibility to also include accelerometer and gyroscope data from the key to better estimate its trajectory has also been identified, and with this information further predict the intention of the driver.

Bibliography

[1] *Volvo Global News Rooms Image Gallery*. Accessed: 2020-02-07. 2014. URL: https://www.media.volvocars.com/image/low/175507/1_1/5.

[2] Joakim Andersson. "Predicting Vehicle Motion and Driver Intent using Deep Learning". Accessed: 2020-04-16. MA thesis. 412 96 Gothenburg: Chalmers University of Technology, 2018. URL: https://odr.chalmers.se/bitstream/20.500.12380/255338/1/255338.pdf.

[3] M. Roth, G. Hendeby, and F. Gustafsson. "A Driving Intention Prediction Method Based on Hidden Markov Model for Autonomous Driving". In: (2014). URL: https://arxiv.org/pdf/1902.09068.pdf.

[4] Jedidi Kamouaa. "Turn-Key Passive Entry/Passive Start Solution". In: *Automotive Compilation* vol. 10 (2013). Accessed: 2020-02-07.

[5] *Model 3 Key Fob*. Accessed: 2020-02-07. 2020. URL: https://www.tesla.com/support/model-3-key-fob.

[6] Xian Ling Liang. "Ultra-Wideband Antenna and Design". In: (2012). DOI: 10.3390/s16050707.

[7] Mukta Athavale and Andrea Giorgetti. "Performance analysis and effect of multi-user interference for ultra wideband communications". In: (2008).

[8] Jaanus Sepp. *TDoA vs ToF in Ultra Wideband RTLS - Why is it important to know*. Oct. 2017. URL: https://www.eliko.ee/tof-vs-tdoa-ultra-wideband-rtls/.

[9] *Cell Phone Trilateration Algorithm*. July 2019. URL: https://www.101computing.net/cell-phone-trilateration-algorithm/.

[10] *DWM1001 Product Brief*. 2018. URL: https://www.decawave.com/dwm1001/productbrief/.

[11] *MDEK1001 Quick Start Guide*. 2017. URL: https://www.decawave.com/mdek1001/quickstart/.

[12] *DWM1001-DEV Product Brief*. 2017. URL: https://www.decawave.com/dwm1001dev/productbrief/.

[13] Simo Särkää. *Bayesian Filtering and Smoothing*. Cambridge, United Kingdom: Cambridge University Press, 2013.

[14] M. Roth, G. Hendeby, and F. Gustafsson. "EKF/UKF Maneuvering Target Tracking using Coordinated Turn Models with Polar/Cartesian Velocity". In: (2014). URL: https://liu.diva-portal.org/smash/get/diva2:734112/FULLTEXT01.pdf.

[15] S.E. Li et al. "Kalman filter-based tracking of moving objects using linear ultrasonic sensor array for road vehicles". In: *Mechanical Systems and Signal*

Processing 98.1 (2017), pp. 173–189. DOI: https://doi.org/10.1016/j.ymssp.2017.04.041.

[16] Max Nilsson. "Performance Comparison of Localization Algorithms for UWB Measurements with Closely Spaced Anchors". Accessed: 2020-03-11. MA thesis. 971 87 Luleå: Luleå University of Technology, 2018. URL: http://www.diva-portal.org/smash/get/diva2:1251202/FULLTEXT01.pdf.

[17] *Bayesian Decision Theory Made Ridiculously Simple*. Accessed: 2020-02-26. 2017. URL: http://www.statsathome.com/2017/10/12/bayesian-decision-theory-made-ridiculously-simple/.

[18] A. Ferreira et al. "Performance Analysis of ToA-Based Positioning Algorithms for Static and Dynamic Targets with Low Ranging Measurements". In: (2017). DOI: 10.3390/s17081915.

[19] *MDEK1001 System User Manual*. 2017. URL: https://www.decawave.com/sites/default/files/mdek1001_system_user_manual.pdf.

[20] *Precise Indoor Positioning is finally here thanks to Ultra-wideband RTLS*. Oct. 2018. URL: https://locatify.com/blog/in-practice-precise-indoor-location-detection-with-uwb-ultra-wideband/.